paligo

Paligo vs MadCap Flare

Flare

vs
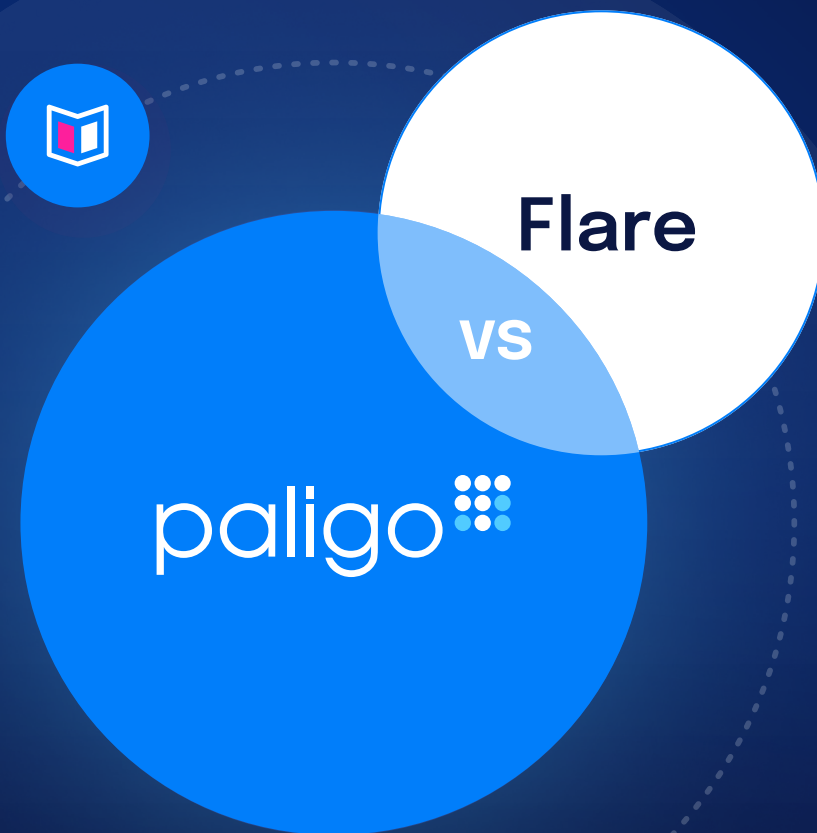
paligo

Table of Contents

Paligo vs Madcap Flare Product Comparison

Madcap Flare and Paligo are both tools that are designed for creating single-source, reusable technical documentation. But although they have a similar purpose, there are some very significant differences between the products.

Here, we are going to look at how Flare, an HTML-based Help Authoring Tool (HAT) compares to Paligo, an XML-based full component management system (CCMS). The comparison is based entirely on our understanding of Flare features and you should of course evaluate each assessment based on your own experience. This evaluation is intended to help you identify the main differences we believe you should be aware of, and why Paligo is a more robust long-term solution and a better choice for your technical documentation.

Structured and Topic-Based Authoring

Feature	Flare	Paligo
Single-sourcing and topic-based authoring	Flare supports single-sourcing and topic-based authoring.	Paligo supports single-sourcing and topic-based authoring.
XML-based	Flare is not XML-based. While they often call the editor an XML editor, the source format is actually XHTML. This is an unstructured format that fosters inconsistency, and which can be very difficult to maintain, process, and convert.	Paligo is fully XML-based, and uses an adapted topic-based version of the robust content model DocBook as the source format. This is an international open XML standard. Paligo has taken this XML standard and added a framework of additional features around it to offer a fully topic-based model very similar to DITA while at the same time providing the stability and maturity of DocBook. This XML source ensures proper structure, consistency, and an unparalleled flexibility for processing the content in multiple ways.
Structured Authoring	No. Flare does not ensure structured authoring. Being XHTML-based, Flare does not impose true structure on the content. There are very few rules about where different types of content can be placed. Also, most of the markup depends on authors adding classes. In the long term, this can can result in a chaos of inconsistent tagging.	Paligo fully supports structured authoring with a very robust XML schema based on an open standard. Rules govern the content model and all authors use the same elements. This creates a consistent, future-proof structure that can easily adapt to any output format. it's implemented in such a way that you do not need to learn Docbook as Paligo guides you through the process.

Feature	Flare	Paligo
Based on Open Standard XML (no vendor lock-in)	No. Flare is using a proprietary modified version of HTML. The Flare source is XHTML and this allows inconsistent structure. Potentially, this can result in poor quality exports for migration or content exchange purposes.	Yes. Paligo is based on the open standard DocBook 5. The fact that Paligo uses DocBook for topic-based authoring is all handled in the system, and the content can all be exported to DocBook 5 format at any time. Paligo offers two types of XML exports to ensure no vendor lock-in. Your content will be future-proof.

Authoring

Feature	Flare	Paligo
Consistent document structure (future-proof)	MadCap Flare lets authors invent their own tags/styles at will. This is *not* a good thing. Apart from the inconsistent structure, it can also cause a maintenance nightmare, especially where there are teams of authors. Other issues than can arise include major migration issues involving inconsistent content and continuity when someone leaves.	Paligo ensures consistency and future-proof documentation by using a rule-based content model. Authors do not invent their own tags, but instead have a rich full set of established and proven elements to select from.
User-friendly interface	User-friendliness is a subjective topic, but the busy interface of Flare is well-known to have a steep learning curve for new users. Companies often need to bring in consultants to get their projects going.	Paligo is in the cloud and so has no installation. It's clean, simple interface makes it easy to find the content and features you need. Following the simple Getting Started tutorial is often enough to start producing reusable topic content and publications on the first day.
Global reuse of content	By default, Flare only reuses content within one project. To reuse content globally you have to set up complex structures with special projects to import reusable content.	Full global reuse is a core feature of Paligo. All of your content is in the database, ready to be reused.

Feature	Flare	Paligo
Dynamic text reuse	No. Flare's text snippets are reusable, but are regular files to keep track of, files that can cause broken links where reused, etc.	Paligo's Dynamic Text Reuse and Informal Topics are more powerful than snippets in several ways, as everything is maintained in a database. This improves flexibility and also eradicates the problem of broken links, leaving you free to move objects around as you please. This also enhances ease of reuse, and provides lots of reuse information, great findability and more.
Full automatic link management	Flare has tools for fixing broken links to a certain degree, but only within a project, and/or with complex imported global projects (that need to be updated). There is no real dynamic link management. This makes link management in Flare a rather error prone process.	With Paligo you don't have an issue with broken links because linking is not managed by relying on files and file paths. Links are managed by relations in the database, and cannot break by moving topics or images around. Paligo makes sure they are always intact.
Single-source multi-dimensional conditions	In Flare there is only one condition attribute. This can create very complicated lists of values that are hard to maintain. It can become a logical mess when you have a lot of condition scenarios in your environment. And if you need to rename a value, you'd need to go back and reapply it everywhere.	Paligo has a large number of available condition attributes that can be combined to create a logical profiling. You can have different conditions say for audience, product, version and more. You can test and publish the results easily. If you need to rename attribute values, that is no problem, it is automatically updated everywhere.
Single-sourced multi-dimensional variables	Variables in Flare are maintained in a regular file, and there is no feature for updating them automatically and finding out where they are used is rather complex. By default, Flare does not have multi-dimensional variables, you have to work around it.	Paligo has fully multi-dimensional variables, where by default you have columns to specify variants for different products, audiences, or whatever it might be. There are also several different types of variables available, text variables, translatable variables, XML variables, and even image variables. You can also easily see where variables are used, just like with everything in Paligo.

Feature	Flare	Paligo
Taxonomies	Flare has no support for taxonomies.	Paligo has great support for taxonomies and with several usage areas: Use taxonomies to organize complex projects, find and filter content, filter/ condition content, create related topic links lists, and more. If you have been using Flare or a similar HAT tool before, taxonomies will be new to you. They will become a very useful tool.
Offline editing	Flare does not need an internet connection for editing topics.	Paligo is cloud-based and the normal approach is to work fully in the cloud-based editor, requiring an internet connection. But for those occasions where you need offline access, such as on a plane journey, Paligo offers the optional Oxygen plugin for offline editing. But more realistically, these days this is rarely an issue. If you don't have internet access 99% of the time, you'll probably have a hard time creating useful documentation anyway.

Publishing

Feature	Flare	Paligo
Multi-channel publishing	Flare has a long list of formats, but many of them seem outdated and not likely to interest many users today. Apart from the main delivery options such as HTML and PDF, you find formats like CHM, Air, etc, formats which are not even supported anymore, and which have next to no demand.	Multi-channel publishing is one of the core features of Paligo, with a wide range of output formats. Additionally, Paligo focuses more on the output formats that matter most today. Paligo publishes to clean HTML, an HTML5 help center design, professional print PDF, Word, SCORM for eLearning, presentation slides, support knowledge bases such as Zendesk, Freshdesk, and Salesforce. Some of the lesser used formats are available in Paligo as well, such as Eclipse help, JavaHelp, etc. They are just not enabled by default as we focus on the more popular outputs.

Feature	Flare	Paligo
Creating layouts: customization	Flare is known to have good built-in tools for creating your layouts. However, Flare has taken the Do-It-Yourself approach all the way. Partly because it is based on HTML, rather than XML. This means the possibilities of more advanced customization than the built-in tools allow are much more difficult or even impossible, depending on the output format.	Paligo offers a built-in Layout Editor too, for building the most common layouts for PDF-based technical documentation. Paligo also has all the options you need to create any HTML5 layout. Over and above that, Paligo is not limited to what is built in. Everything is fully customizable as a professional service, or by WebDAV to edit the source configuration files directly if you have the in-house skills. This is possible because Paligo is based on XML, a format inherently made to be fully customizable by open standard techniques.
New and updated output features	Flare has several advanced output features, but because of longer release cycles, updates are rather rare and often doesn't keep up with new possibilities.	The fact that Paligo is a fully cloud-based tool, and based on a robust standard, makes it easier to add more new features to keep up with the latest possibilities available. Recent examples just from the past few months include dynamic code snippets, Swagger integration, feedback features for Google Tag Manager, integration with 3D content platforms, etc.
Integrations with popular support help desk platforms	Flare has an integration with Salesforce as an add-on module.	Zendesk, Salesforce, Freshdesk, Desk.com, Zoomin. Besides just publishing to help desk formats, Paligo also adds additional intelligence, like being able to auto-update and correct linking and while publishing to separate categories in Zendesk, etc.

Pricing and Support

Feature	Flare	Paligo
Excellent support	Flare has a good reputation for a high level of support.	Paligo includes top-notch support, and our customers continuously attest to never having seen any support as responsive and knowledgeable about technical documentation and single-sourcing issues. Here are just a sample out of many many responses Paligo gets on a regular basis: "I just can't say enough about how great Paligo's Support is. You guys are so fast, so helpful - you really care about my issues. Thank you so much! " Rochelle Fisher, Sentinel One See more on our web site and in the Paligo User Community, or contact your account manager for references, but primarily, try it out and see for yourself.
Pricing	The pricing of Flare alone cannot really be compared to Paligo, as Flare itself is an editor, not a content management system. Flare is now selling a number of modules bundled in a larger package and a subscription pricing. While still not offering real CCMS functionality, this is more expensive than the Paligo Business plan.	Paligo is an Enterprise grade CCMS, but priced much lower than traditional CCMS systems due to the cloud-based approach. Pricing really cannot be compared with a HAT editor, as the ROI long term for true structured authoring and controlled content management are unparalleled. Paligo offers the full Component Content Management System all-in-one package with full structured authoring and different plans fitting Small Businesses to Enterprises alike.

Translation

Feature	Flare	Paligo
Single-sourced Translation Management	In Flare, translations are not single-sourced, they are separate batches of files that you need to keep a track of.

This makes translation management as complex as file management is in general, but with the exponential complexity the more languages you have. | In Paligo, by contrast, each topic is directly related to its translation. There are no separate batches of files to keep track of and update.

If you make changes, you don't have to keep track of what to update. |
| **Auto-translated label strings** | Because most of the output is controlled by CSS in Flare, there is very limited support for auto-translated strings. | Notes, warnings, link text, page references, etc, and hundreds of similar strings, are automatically translated to 60+ languages. |

Security and platform

Feature	Flare	Paligo
Security	Traditionalists often have a "feeling" that their content is more secure in installed software, because then it can be managed on the company's own computers, servers, etc.

But most companies now have come to the realization that cloud security is usually even higher than they can provide themselves. | Recent surveys have found that the ones that are most afraid of cloud security are those that are not that familiar with IT. On the flip side, IT specialists rank cloud-based products as highly secure.

Paligo has the well-established Amazon security, backs up the entire database every hour, and saves every backup for 90 days, making 2,160 backups available at any point in time.

Not many companies would do that for users working on installed software with files on their machine or even on a company file server. |
| **Instant deployment** | No. Flare is a Windows-installed client. Deployment needs to be done on each individual machine, and may entail connecting it with other parts of the separate modules or 3rd party software.

This process may have to be repeated at every new release of patches and updates. | Paligo is instantly deployed. There are no downloads, installations or patch installations.

Updates and bug fixes alike are automatically applied at frequent intervals, with no work required by the users or their IT departments.

You can focus on producing quality content instead of making the software work. |

Feature	Flare	Paligo
Full cross-platform support, Windows, Mac, Linux	Flare only works on Windows. The only way to use it on other platforms is to use a virtual machine, but not many users of Mac or Linux are comfortable using such a heavy workaround in a virtual machine. A virtual machine also creates its own set of maintenance problems.	Paligo is platform independent. Your users work in whatever environment they choose. Your implementation can contain a mix of Windows, Mac and Linux Paligo users all working together, while everyone shares the exact same environment for technical documentation.
"Where used" (reuse) information	No. Flare does not have proper reuse information features. There are tools for reporting, but nothing that is a real equivalent of a proper CCMS "Where used" functionality.	Paligo has excellent Reuse Information features, available throughout the system. You can see exactly where topics are reused, blocks of elements, even single text fragments and variables. Not only that, for each single piece of content you have full details of their reuse, history, and can preview in context from anywhere.
Agile release schedule: Continuous updates and bug fixes	Flare has regular updates and bug fixes. But they are approximately 6 months apart, and require re-installation and re-deployment. And that is assuming you have purchased the upgrade, as it's not always included.	The fact that Paligo is fully cloud-based allows us to be truly agile. You won't have to wait for the next release 6 months down the line to get a bug fixed, then deploy to all users, and then hope the new version itself doesn't contain bugs (that would mean waiting for another hotfix and then re-deploy). With Paligo new features and improvements are released continuously, free of charge, with major releases roughly every quarter. Minor improvements and bug fixes are released every 1-2 weeks.
Statistics and reporting	There are some statistics and report tools, but not related to content management and control of your content.	See who did what and when, down to the smallest text fragment. Also see the metrics for reuse, find unused content, view an audit trail of content history, and more.

Feature	Flare	Paligo
Scalability	The amount of content may not be an issue in itself. The problem is not storing a large amount of content on a computer or server. It's the management part that becomes problematic in a file and folder environment, and this is severely limiting. You just cannot keep track of 100s (never mind 1,000s) of topics, and their relationships and dependencies, in such an environment.	Paligo is made to scale. The database structure expands as needed to accommodate more content and users. But the most important thing is that it scales to keep track of the content. Start small and upgrade within the same environment.
REST API integrations	While Flare does have possibilities to integrate with other software, it is a locally installed client, which makes it more difficult to integrate with modern web applications.	Paligo is made from the ground up to work with REST APIs. All our integrations use this API, and the integration list keeps growing fast. It is also possible to integrate directly the other way around with Paligo's API through a customization, and in 2019 there will be a public open API.

Collaboration

Feature	Flare	Paligo
Teamwork: Global collaboration on content	Flare is locally installed, and MadCap central is essentially a file repository for uploading and downloading files, content is not immediately shared and worked on in the same platform.	In Paligo, all users are always working on the same content repository in the same database. The full history of each piece of content is stored as metadata, who did what, when, and what has changed. Everyone is always on the same page.
Seamless all-in-one platform	Flare is an editor. And to achieve additional functionality a mix of different applications need to be combined, like MadCap Central for file sharing, Git or Subversion for a sort of versioning, etc.	Everything is built into the CCMS as an all-in-one powerhouse solution: Editing, publishing, translation management, workflow and review, version management, and much more.
Workflow built in	No. Flare does not have workflow capabilities built in.	Yes, Paligo has a full workflow with editing/contribution, review, translation, and release workflow stages and assignments, together with a powerful Planner tool for project management.

Version management

Feature	Flare	Paligo
Version management made for technical writers	No built-in versioning for technical writers. Only 3rd party connections to systems such as GitHub and Subversion with functionality geared towards development, unsuitable for technical writing needs.	Yes, the version control is purpose-built for technical writing, taking into account the specific nature of reusable content and single-sourcing. Besides branching and release management, there is also version history for all topics, and accountability and rollback down to the smallest text fragment and variable. As a note, branching will revolutionize how you manage multiple versions simultaneously to mirror how your developers create their product.
Check-in and check-out of content	No. Flare can only connect to 3rd party versioning systems, that use the "conflict resolution" model, which is notoriously difficult for non-developers.	Yes. Paligo uses a check-in/check-out model for collaborating on content. This means there are never any conflicts of content to resolve, which means a much more user-friendly approach for technical writing.
Version Compare: see what has changed	No, since Flare does not have workflow for release management, there is no full version compare of publications in Flare.	Yes, single topics as well as full publications have version compare to see what has changed from one review cycle or release to another. Tied to the workflow, making it easy to see just what happened since the last review, release, etc.
Image management for multiple versions	In Flare you need to keep track of multiple files in multiple projects for e.g screenshots in localized software and the like.	Image objects keep track of multiple versions of images, for different output formats. You can manage your images in source formats like Photoshop PSD or Illustrator AI, and Paligo will automatically use the correct outputformat and resolution for different outputs. Paligo's image objects even keeps track of different language variants, no separate files or projects to track to manage multi-lingual images.