

VASONT FEATURE SERIES:

Release Management in Vasont

This white paper focuses on the critical business issues surrounding release management and explains how the Vasont Content Management System's branch capabilities can solve these problems for technical communicators.



Phone: 717-764-9720
Email: info@vasont.com

www.vasont.com

Release Management in Vasont

In the world of technical communications, one of the biggest issues is how to manage the editorial cycles of documentation for multiple product releases at the same time. No matter what product an organization sells—motors, smart phones, refrigerators, software or hardware—they often have multiple versions of these products overlapping through the manufacturing or development process. For the writers who must create and edit the user manuals and other technical documentation needed to accompany these products upon their release, this can be a coordination nightmare.

This paper focuses on the critical business issues surrounding release management and explains how the Vasont Content Management System's branch capabilities can solve these problems for technical communicators.

SITUATION ANALYSIS

Organizations are unable to provide customers with accurate documentation for their products when multiple releases of those products occur in a short period of time. This is because writers struggle to track and coordinate editorial changes to the documentation across multiple releases simultaneously. Often, this tracking is done manually using spreadsheets. With a team of writers working on the different pieces of the documentation, this manual method of tracking and incorporating changes leaves much room for errors.

The consequences of these errors can be very costly for the organization. These errors can result in more expense to the organization for providing additional support for confused customers. Inaccurate documentation could ultimately cause customers to lose trust in the organization's products; therefore, customers buy their competitors' products, resulting in lost revenue.

To illustrate the complexity of release management, the next section will step through a hypothetical scenario of an editorial cycle.

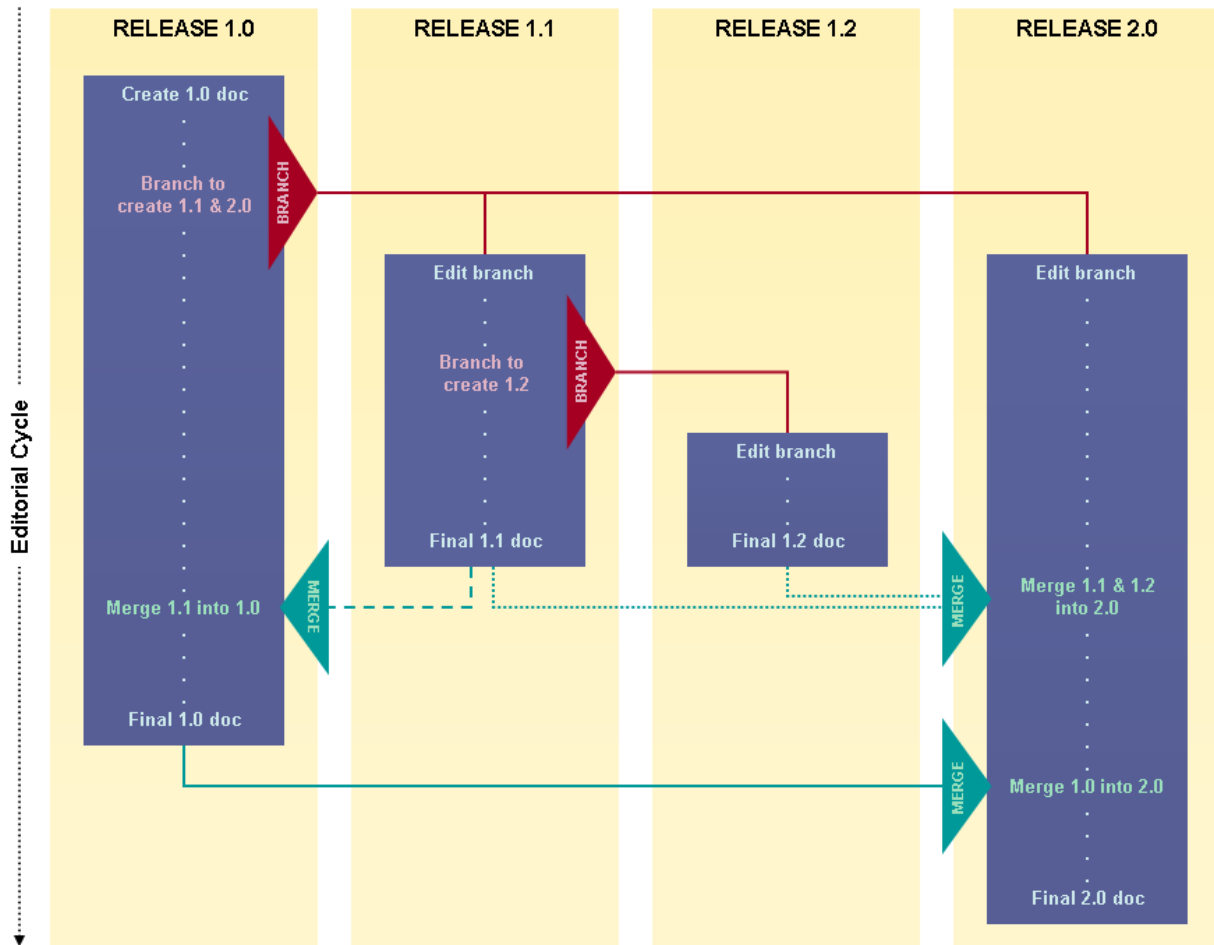


Figure 1. Hypothetical example of an editorial cycle for the documentation of four product releases. The light blue dotted lines in the editorial cycle represent ongoing editing to the documentation.

Scenario

Suppose an organization develops a product, and the first release of this product is labeled 1.0. The technical communicators create the documentation — let's say, a user guide — that is published and delivered with release 1.0 of this product (see the first column in Figure 1).

At some point, the product is enhanced, and specifications begin to come through to the technical communicators for release 1.1. In addition, a new, similar product is in development that is called release 2.0. Both products have many similarities to release 1.0, so the technical communicators make two copies of the documentation for release 1.0. They edit one copy to create the documentation for release 1.1, while the other copy is edited to create the documentation for release 2.0 (see the second and fourth columns in Figure

1). Meanwhile, the documentation for release 1.0 is still being edited by its content owners.

Later, the technical communicators receive specifications for yet another enhancement to the first product. This new enhancement is labeled release 1.2. Once again, the technical communicators make a copy of the documentation, this time for release 1.1, and begin to edit it to create the release 1.2 documentation (see column 3 in Figure 1). Now, four sets of documentation are being edited all at the same time for releases 1.0, 1.1, 1.2, and 2.0.

After a period of time, the documentation for releases 1.1 and 1.2 are final, and no further changes will be made to these documents. However, release 1.0 documentation is still being edited for republication, and the changes made to release 1.1's documentation must be incorporated back into the 1.0 manual (see the dashed teal line in Figure 1). Now, a writer must determine what those changes are and merge them together.

At the same time, the documentation for releases 1.1 and 1.2 must be merged into the 2.0 document since both of these manuals were created in their entirety after the 2.0 document was started (see the dotted teal lines in Figure 1). In this case, three manuals must be coordinated together, making sure the information is accurate and no important information is missed.

Finally, release 1.0's documentation is completed. Since many updates have been made to this manual since it was duplicated to create the documentation for release 2.0, it must now be compared and merged with release 2.0's manual (see solid teal line in Figure 1). After this merge, final updates can be made to release 2.0's manual to complete its editorial cycle.

Issues

In this example, the complexity of release management is very clear. As multiple product releases move through development, the documentation must run on parallel editorial cycles, creating a number of issues.

- *Duplicated content:* Technical communicators must duplicate content at various points in the editorial cycles to create new documents in order to leverage existing content. When each copy is updated, the updates do not populate to the other copies of the same content. As the documents are completed, all copies are different. However, it may be important for some of those changes to appear in the documentation for other releases, too.

- ▾ *Manual change tracking:* In order to merge the changes in these documents, the changes must be tracked in some way. Often, changes are tracked manually using spreadsheets or lists in separate documents. Using this manual method, there is no guarantee that every change was accounted for in the list.
- ▾ *Manual comparison for merging:* Technical communicators spend many hours of time comparing the change lists to the manuals, trying to figure out where to merge content. This is a very time-consuming process.
- ▾ *Copy-and-paste merge:* Once the merge points are identified, technical communicators must copy-and-paste content back and forth into the documents — a process that is far from being error-free.

To complicate the process, an organization typically has a team of technical communicators working on various topics within the documentation. This can be three writers or three hundred writers, depending on the size of the organization. The coordination of many content changes, along with the coordination of large groups of technical communicators, compounds the issues.

SOLUTION: VASONT CMS'S BRANCH CAPABILITY

To solve the issues of release management, the Vasont Content Management System provides a branch feature that helps users manage complex editorial cycles for simultaneous releases.

Important Considerations

First and foremost, it is important to understand that the Vasont CMS requires a user to have the appropriate permissions to access and branch content. These security permissions restrict unauthorized users from altering content. User permissions can be set and maintained in Vasont's Administrator tool by an organization's CMS administrator.

Second, note that Vasont's branch capability can be performed at various levels. Entire maps or manuals can be branched, as illustrated in Figure 1. However, Vasont also provides the ability for content to be branched at a topic or module level, allowing maps or manuals to maintain their reuse while pulling in any new branches of topics or modules that might have critical changes that pertain to the document. The flexibility to perform branching at different levels gives users more options for handling their branching issues.

Third, Vasont's architecture is based on maximizing content reuse at all levels, and this principle is applied throughout the system wherever possible — in words, components, modules, and translations. Content reuse is also in play in Vasont's branch feature. When a content module is branched in Vasont, any content components that remain unchanged within the module are reused, not copied. For example, if a user branches a topic, and a paragraph within the topic remains unchanged, this paragraph will reflect a content reuse relationship with any other occurrence of that paragraph in the system. This component-level reuse ensures that content is not duplicated in the CMS when content is branched.

How Vasont's Branch Feature Works

When content is branched to develop new releases, users give each branch a unique label. Vasont maintains connections between all of the branches created from a source. For example, using the scenario in Figure 1, Vasont would maintain connections between the branches for releases 1.0, 1.1, 1.2, and 2.0 through their entire editorial cycles. Because of Vasont's ability to maintain relationships between branches, users do not have to maintain manual logs of this information. A list of branches with labels and other details for a selected map or module is accessible in the Details pane (see Figure 2).

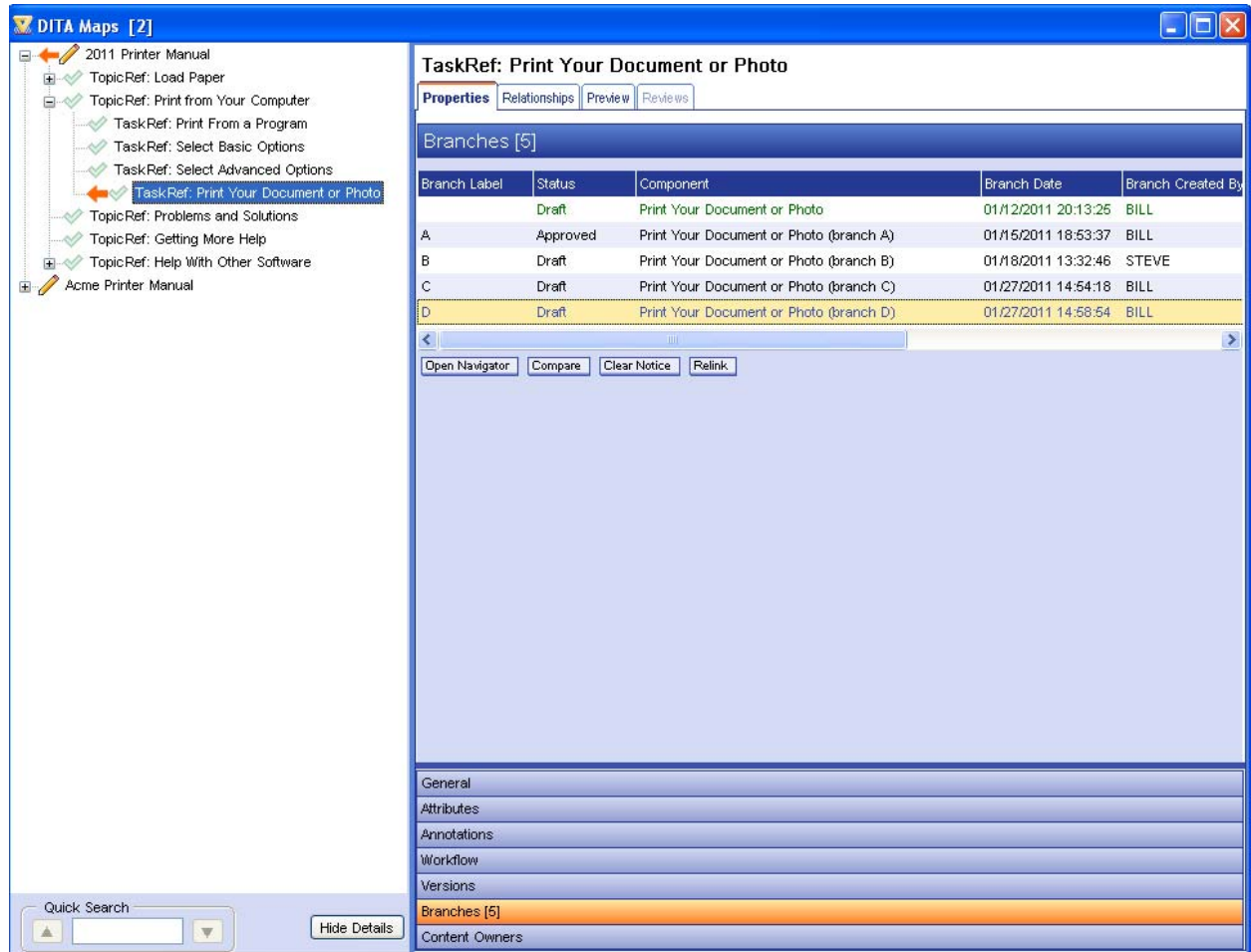


Figure 2. Vasont's Details pane shows a list of the branches created for any content that is selected in the tree view.

As branches are edited, Vasont automatically remembers all changes made to the content. This eliminates the issue of users having to manually track changes in other documents, saving hours of time. Because the CMS tracks changes automatically, every change is captured from all users.

Using the connections between the branches, Vasont alerts the owners of the documentation when new branches are created. For example, the owners of the documentation for release 1.0 would be alerted with a Branch Notice icon on their content when the branches for 1.1 and 2.0 are created.

These users can then evaluate the new branches with the branch they currently use. Any two branches can be compared quickly. Vasont's Branch Compare window highlights the differences between two selected branches (see Figure 3). Using the Branch Compare, users can make intelligent decisions on which branches are most appropriate for their documentation.

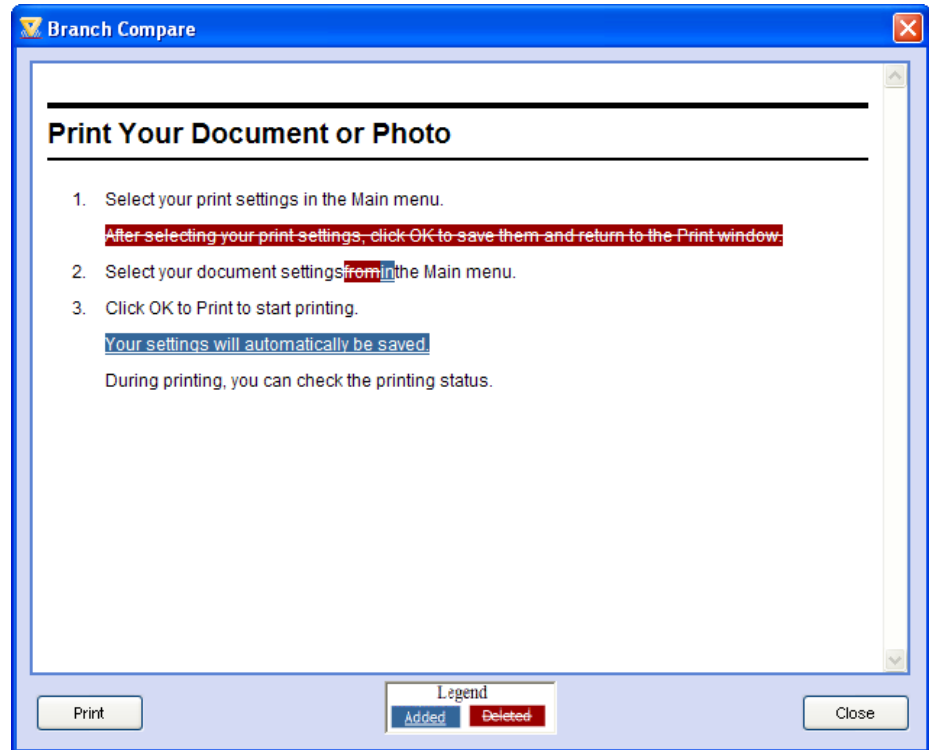


Figure 3. Vasont's Branch Compare window highlights the differences between two branches.

Users can either merge new branches in their documentation or clear the Branch Notices and continue to use their existing branches. Once users perform a comparison of two branches and determine that a new branch is most appropriate for their documentation, they can simply relink the new branch to merge it into their documentation with one click. The Branch Notices automatically clear when branches are merged.

Vasont's branch relinking is bi-directional, meaning that branches can be relinked to either newer or older releases. For example, in Figure 1, we mentioned that when release 1.1 is completed, it must be merged up with release 2.0, but also merged down to release 1.0 (dashed teal line in Figure 1). This is because release 1.0 is being republished, and the editors want the documentation to include changes made in release 1.1. Using the unique branch labels given by the users, it is easy to quickly determine where merges need to occur.

SUMMARY

The Vasont CMS provides a flexible branching and bi-directional relinking solution for organizations with complex release management challenges. Vasont's branch capability automates much of the release management process, saving technical communicators hours of time and making an organization's content more accurate within and across documentation. The Vasont CMS:

- ▾ Eliminates manual tracking of branch relationships
- ▾ Eliminates manual tracking of content changes within branches
- ▾ Prevents content from being duplicated within the CMS
- ▾ Provides a quick and automated way to compare differences between branches
- ▾ Eliminates copy-and-paste merging of branches by automatically relinking branched content with one click

Organizations can build a comprehensive release management strategy that fits their needs using Vasont's adaptable branch feature.

Headquartered in Pennsylvania, Vasont Systems is a provider of content management software and data services, backed by more than 60 years of experience in the information management and publishing industry. Its Vasont® content management system enables organizations to manage and store their multilingual content once for multi-channel delivery. Fortune 1000 companies and organizations from industries including manufacturing, technology, publishing, and healthcare use Vasont to manage and produce multilingual technical documentation, training programs, and reference materials. Vasont® is a registered trademark of Vasont Systems. www.vasont.com