

DocumentationGuidelines

The Challenge

Creating a good website and set of project documentation for an open source project can be challenging:

- The documentation must serve the needs of novice users while also providing comprehensive information for those wishing to push the limits and extend the software.
- Users often need information on what configuration or architecture to use. But typical open source documentation gives three different ways to solve every problem, and rarely describes the pros/cons of each approach. Even worse is when the documentation describes in detail one approach that is informally deprecated or recommended against.
- A significant amount of information is shared ad-hoc by a wide range of users/developers. This means users who have a problem often need to search the project web-site, readme files, FAQs, mailing lists and a variety of blogs. This is especially overwhelming for new users.
- Versioning is important. Users tend to download released versions (as opposed to nightly snapshots) but read the docs from the web site. If the documentation refers to unreleased features, this can be confusing.

Luckily, Velocity has a great web site and an excellent set of documentation. This note provides suggestions to keep the quality high while also taking advantage of the knowledge across the community.

Resources

Current Velocity documentation includes:

- [Web Site](#): Official information about the project, community, and software documentation. Stored in source control as xdoc files.
- [Software Download](#): Each released version of the software has a snapshot in time of the web site, project documentation, and Javadocs.
- [Wiki](#): A place for frequently changing content and user contributions, writable by anyone in the community.

Other non-official documentation includes

- [Mail Lists](#): Archived at numerous places, including (for example) at [Eyebrowse](#), [velocity-user](#) and [velocity-dev](#).
- [JIRA Issue Tracker](#): Contains info on known bugs and feature requests.
- [Google](#): Numerous articles and blog entries on Velocity. Some of them are linked to from the Articles page on the Wiki.

Guidelines

Having said all that, here's a few simple guidelines for contributing and storing documentation:

1. The software documentation (primarily the User's Guide and Developer's Guide) should contain a comprehensive overview of Velocity features. As new features are added to source control, these guides should be updated at the same time and stored in source control.
2. The software documentation available on the web site should correspond to the latest released version of the software. This is as opposed to the latest docs in source control (which are available within the nightly snapshot along with the source itself).
3. When possible, typos or errors in documentation for released versions should be fixed, stored in a source control branch, and updated on the web site.
4. The documentation should be comprehensive, but should also offer recommendations and best practices when possible. As community opinion changes as to recommended approach (e.g. the move from VelocityServlet to VelocityViewServlet) the documentation should be updated to reflect this change in thinking.
5. Each issue from the Issue Tracker that is resolved (as well as any other feature additions) should be logged in a Change Log (project documentation).
6. The Wiki can be used for community contributions or frequently changing content. Examples include a FAQ, links to community articles and published articles/books. (Community articles can be on the author's web sites, or with their permission on the Wiki itself).
7. The Wiki is a good place to keep developer-related info (which can benefit from group editing and review). This may include Road Map, issue priorities, and archiving of debates over future architecture and direction.

Get Involved

Want to help? Here's what you can do:

1. Write an article about your experience with Velocity and put a link to it on the Wiki.
2. Update the Wiki with a link to your Velocity-based web site or software.
3. Contribute a FAQ entry in the Wiki based on your experience or an exchange on the mailing list.
4. Fix a typo or mistake in the project documentation and submit a patch via Bugzilla. Or fix a typo in the Wiki directly.
5. Create a new software feature and submit it with documentation in Bugzilla.
6. Offer constructive feedback on these guidelines (or the documentation in general) on the Velocity mailing lists.