Multi-Version Support

Scroll Versions enables you to manage concurrent versions in a single space. Multiple versions of software, different product variants, and even multiple languages of documentation can be managed. Previously, managing concurrent versions of documentation involved a lot of manual work. See Background: Managing multiple versions of Documentation in Confluence.

With version management, all changes to pages in a space are scheduled for one of the existing versions. The Author selects the working version, and subsequent changes are scheduled for that version. After all changes for a version are complete, the version is published in one step and available to Wiki-Users. For more information, see the following screencast about the concurrent version management:

![Info]

The screencast was made with Scroll Versions 1.0. If you are using Scroll Versions 2.0, there might be some differences in some steps.

Background: Managing multiple versions of Documentation in Confluence

The version management in Scroll Versions is based on the documentation release process used at Atlassian for their own software products. The Atlassian process is the best practice for managing documentation with Confluence.

To manage their documentation, Atlassian keeps all documentation for one product in one space and creates a different space for each different documentation version:

- The master space always contains the most recent version of the product documentation. For example, "JIRA" is the space key of the most recent JIRA documentation.
- Older versions of the documentation are archived to other spaces. For example "JIRA044" is the space key of the documentation for JIRA 4.4.

The following diagram illustrates the spaces involved in this process:

![Diagram]

Additions or updates to the documentation are added to the master space and hidden from users by using page restrictions. Prior to the product launch, the technical writing team uses the copy space plugin to archive the existing version of the documentation (step 1), remove the page restrictions and make the new version of the documentation visible to the users (step 2). The process is described in detail in this excellent blog post by Rosie Jameson: Technical Writing in a Wiki – Documentation Release Management.

The Scroll Versions version management automates the version management process and provides additional functionality to support more complex scenarios such as product versions, more flexible publishing and archiving and continuous deployment.