



Create multi-version documentation

Last Modified on 10/02/2025 10:36 am EDT

Multi-version documentation in KnowledgeOwl



While KnowledgeOwl has a **built-in versions feature**, this feature is intended for content management and only one version can be live at a time. If your product or software requires multiple versions to be live, you will need to use a different set of features.

If your product or software has multiple versions, you might need to have multiple versions of the documentation live at the same time. While we don't have a built-in solution, there are multiple ways to solve this with KnowledgeOwl.

1. Multiple knowledge bases, with one knowledge base per software version.
2. One knowledge bases with different articles for different versions, controlled by reader groups
3. One knowledge base using tabs in article for versions.

Pros and cons of multi-version documentation approaches

Approach	Pros	Cons	Recommended situations
----------	------	------	------------------------

Approach	Pros	Cons	Recommended situations
Multiple knowledge bases	<ul style="list-style-type: none"> • Ease of use - you can copy a knowledge base and only update the documentation that changed • Works with public knowledge base - readers can choose which version and toggle between kbs • Synced categories and share content articles - keep content updated between kbs 	<ul style="list-style-type: none"> • Cost - each knowledge base (kb) is \$40/kb/month) • Setup - each kb has its own URL and security, so this will need to be set up in each kb • Maintenance - global changes will need to be made in multiple kbs 	<ul style="list-style-type: none"> • One-time major software rebrands/updates • Enterprise customers with a custom version of the software • Similar products with different names/brand but shared content • Very small number of versions
Reader groups	<ul style="list-style-type: none"> • Only show a single version to the reader • Can show specific content to only certain versions using snippets or topic articles 	<ul style="list-style-type: none"> • Requires authentication • Maintenance - reader groups must be set up and managed for readers and content • No ability to toggle between versions 	<ul style="list-style-type: none"> • Bespoke software versions where you want to restrict who can see the documentation • Small number of versions

Approach	Pros	Cons	Recommended situations
<p>Tabs in articles</p>	<ul style="list-style-type: none"> ● Reader ease of use - allow readers to toggle between versions in a single article ● Author ease of use - allow authors to maintain all versions in a single article ● Promote upgrading to recent software versions with callouts ● Optionally pass in version information through the URL or reader groups/custom fields to set the version 	<ul style="list-style-type: none"> ● Set up - tabs require custom scripting ● Maintenance - scripts must be kept updated and articles must follow proper formatting for versions 	<ul style="list-style-type: none"> ● Products and software with ongoing version upgrades ● Must support most recent x versions ● Want to show all documentation to everyone