ATLORIA

Understanding Version Lifecycle and Release Readiness

August 25, 2026

Following a version from draft to release candidate

In Atloria, the **Versions** area is where your team tracks each documentation release from early work to publish-ready status. A version usually begins as a new entry in the version list, where its **version name**, **release label**, and **project** help everyone understand what release it belongs to. If your team is working on more than one release at the same time, these details are what separate an in-progress update from a version that is nearly ready to go live.

While a version is still being built, teams typically treat it as a **working draft**. On the version list and version detail page, people look for clear signals that show whether the release is still in progress or ready for a decision. Those signals usually include:

- whether the version is still in a **draft** or working state
- whether it has been moved into a **review-ready** state
- whether it is already the **published** version
- whether required content appears complete
- whether screenshots have been updated for the release
- whether access and visibility settings match the intended audience

The version detail page is where Documentation Managers and Project Administrators usually confirm these readiness signals before moving forward. They use the page to review the version's identifying information, check whether the release is meant for internal work or public readers, and verify that the version belongs to the correct project.

[SCREENSHOT: Versions list showing multiple versions with status indicators, version names, and project association]

If you already worked through version statuses and list behavior in Managing Version Lists Statuses and Comparisons, use that as your reference for reading the list itself. Here, the main point is understanding the lifecycle: a version starts as a draft, moves

through review and readiness checks, and becomes a release candidate only after content, screenshots, and visibility settings all support publication.

Creating and organizing versions for an upcoming release

To prepare for an upcoming release, start in the **Versions** screen and use the **new version** action to create a fresh version record. When you create it, Atloria uses the identifying details you enter—especially the **version name** and the related **project**—to place that release in the correct workspace. These details matter most when several releases are active at once, such as a maintenance update and a larger upcoming release.

A **fresh version** is usually the right choice when you are documenting a new release cycle and want a separate place to prepare content without disturbing what readers already see. Teams often create a new version when they need to preserve the currently published release while building the next one in parallel. If your team is continuing from an earlier release, you may also work from an existing version so prior published content remains available for reference while the next release is updated.

Once the version appears in the list, organize it so the intended release candidate is easy to spot. Teams usually scan the list by:

- **release sequence**, to see which version is older or newer
- **draft state**, to find versions still being edited
- **publish state**, to identify which version is currently live
- **project association**, to avoid mixing releases across projects

On the version detail page, review any editable settings available right after creation. Depending on what your team uses, this may include the version title, internal notes, and visibility-related settings that affect who can access the release later. It is best to confirm these details early so reviewers are not comparing or approving the wrong version.

[SCREENSHOT: New version action from the Versions screen and the version detail page with identifying fields]

For broader release planning across multiple versions, refer back to Managing Documentation Versions Across the Release Cycle.

Reviewing changes and comparing versions before approval

Before approving a release, teams need to see exactly what changed. In Atloria, that usually means opening the version's comparison controls and checking the current draft against the previously published version. This comparison view helps reviewers focus on release-impacting differences instead of manually opening every page one by one.

When reviewers compare versions, they usually look for a few specific signals:

- **added pages** that must be included in the release
- **updated pages** where product behavior or instructions changed
- **updated screenshots** that reflect the current interface
- **changed access restrictions** that may affect who can read the content
- **missing release-critical content** that should appear in the new version but does not

The most effective review flow usually moves between three places: the **version detail page**, the **page list**, and the **comparison view**. Reviewers start on the version detail page to confirm they are checking the correct release. From there, they open the page list to inspect individual pages that were updated, then return to the comparison results to make sure no important differences were overlooked.

This comparison step is especially useful when the release candidate looks complete at first glance but may still contain hidden gaps. A page may have been updated without its screenshot being refreshed, or a section may have been removed by mistake. By surfacing these differences clearly, comparison gives Documentation Managers and Project Administrators stronger confidence that the release matches the actual product state.

[SCREENSHOT: Version comparison view showing differences between the current draft and the previously published version]

If you need a deeper walkthrough of reading comparison results, the next document, Working with Version Comparison Views, focuses on that screen in more detail.

Checking access control and screenshots for release readiness

A version is not truly release-ready until the right people can see it and the visuals match the product. In Atloria, that means checking both **version access settings** and the screenshots used on pages before you publish.

Start with the version's visibility and access settings on the version detail page. Reviewers use these settings to confirm whether the version is meant for all readers, a limited audience, or a restricted group. If the version is intended for a specific audience, make sure the selected visibility matches that plan before release. A version with complete content can still fail a release check if the wrong audience will see it—or if the intended audience will not see it at all.

Then review screenshots connected to the pages in that version. This matters most after interface changes, renamed buttons, updated navigation, or redesigned screens. Teams often open the affected pages and confirm that each image still matches the current product experience. Outdated screenshots can create confusion even when the written instructions are correct.

Common release blockers in this stage include:

- screenshots that still show the previous interface
- pages that were updated but their images were not
- visibility settings that expose the version too broadly
- restricted settings that prevent the intended readers from opening the release

Documentation Managers usually confirm page-level readiness by checking the version's content and screenshots together, while Project Administrators often verify that access settings align with the release plan. Both checks happen before any publish decision is made.

[SCREENSHOT: Version detail page showing visibility settings and linked page content with screenshots]

For more on visibility decisions, see Managing Version Visibility and Reader Access. For screenshot workflows, use Checking Screenshot Readiness Before Version Release.

Deciding when to publish or hold a version

The publish decision happens on the version detail page, after the team has finished its readiness checks. Before selecting the publish action, reviewers usually confirm that the version content is complete, comparison results have been reviewed, screenshots are current, and visibility settings are correct for the intended audience.

In practice, teams usually work with three release states:

- **Draft**: the version is still being edited and should not be treated as final

- **Ready for review**: the version is stable enough for formal checking and release discussion
- **Published**: the version becomes the active release readers will use

Keeping a version in **Draft** is the right choice when pages are still changing, screenshots are incomplete, or comparison results still show unresolved gaps. Moving it into a review-ready state signals that the release candidate is ready for decision-making, even if final approval has not happened yet. Publishing should happen only when both documentation quality and access settings are confirmed.

When a newer version is published, the previously published version does not lose its value. Teams often keep earlier releases available as historical records for comparison, reference, or rollback planning. That preserved release history is especially helpful when readers still need access to older documentation or when the team must verify what changed between releases.

The final decision is usually coordinated between Documentation Managers and Project Administrators. Documentation Managers confirm that the pages, screenshots, and release notes are accurate. Project Administrators confirm timing, visibility, and whether the release should become the active version for readers.

[SCREENSHOT: Version detail page with publish action and status indicators for draft, review-ready, and published states]

If your team also uses formal review decisions, pair this process with Managing Version Review Decisions and Approvals.

Resolving common release-readiness problems

When a version will not move forward, the fastest way to solve it is to check the same readiness signals your team uses before publishing. In Atloria, most release problems fall into a few repeatable patterns.

If a version cannot be published because review items are incomplete, open the version detail page and then check the comparison results. Look for missing pages, unfinished updates, or content that changed in draft but was not fully prepared for release. After that, review screenshots on the affected pages and confirm there are no visual gaps. A release may appear complete in the version list but still be blocked by unfinished page updates or outdated images.

If readers cannot access the new version after release, start with the version's **visibility** and access settings. Confirm that the correct audience or restricted access

option is selected and that the intended release was actually marked as the active published version. This is especially important when several versions have similar names.

If the release candidate looks inconsistent compared with the previous version, open the comparison view again and review the differences carefully. Teams often discover:

- pages that were expected but never added
- screenshots that still belong to an older release
- changed access settings that hide important content
- accidental regressions where useful content was removed

When the team is unsure which version should ship, return to the version list and compare the **version name**, **release label**, **publish state**, and any review indicators shown there. These details usually reveal which version is the intended release candidate and which one is still a work-in-progress draft.

[SCREENSHOT: Versions list with multiple similar releases, showing status and publish indicators]

If the issue is mainly about deciding between two releases, use Comparing Documentation Versions for Release Decisions alongside your version review workflow.

Overview

In Atloria, **release readiness** is the point where a documentation version has moved beyond editing and can be trusted as the reader-facing release. The work does not happen in one screen only. Teams usually move between the **Versions** list, the **version detail page**, the **comparison view**, and the related page content to confirm that the release is complete.

A typical release-readiness review brings together several checks:

- the version is clearly identified by **version name**, **release label**, and **project**
- the version is no longer just a working draft
- content changes have been reviewed against the previous published version
- screenshots reflect the current product experience
- visibility and access settings match the intended audience
- the correct version is ready to become the active published release

This document focuses on how those checks fit together as one lifecycle. It does not repeat the detailed list-management guidance from Managing Version Lists Statuses and Comparisons. Instead, it helps you understand how teams decide whether a version is still in progress, ready for review, or ready to publish.

That distinction matters most when multiple releases are active at once. Without clear version metadata and status signals, teams can easily review the wrong release or publish a version that still has incomplete screenshots or incorrect visibility settings. Using the version list and detail page together helps avoid that confusion and gives everyone a shared view of release progress.

[SCREENSHOT: Atloria Versions area showing list view on the left and a selected version detail page]

Prerequisites

Before working through release readiness in Atloria, make sure you can already open the correct **project** and access its **Versions** area. You should also be comfortable reading the version list and recognizing the basic status and comparison indicators shown there. If you need a refresher on that screen, use Managing Version Lists Statuses and Comparisons.

You will get the most value from this workflow if the following are already in place:

- you have access to the project where the version is being prepared
- at least one documentation version already exists in the **Versions** list
- there is a current draft or release candidate to review
- your team has updated the relevant pages for the release
- any screenshots tied to changed pages are available for checking
- you can open the version detail page and any available comparison controls

It also helps if your team has already agreed on who is making the release decision. In many teams, Documentation Managers review page completeness and screenshots, while Project Administrators confirm timing and visibility. Even if one person handles both tasks, it is important to know who is responsible for the final publish decision.

If your team is still setting up version workspaces or preparing the release structure itself, read Managing Project Version Workspaces before continuing. If you are ready to focus on the actual difference-checking process between versions, continue next with Working with Version Comparison Views.