ATLORIA

Managing Documentation Versions Across the Release Cycle

August 25, 2026

Opening a documentation version and understanding available actions

In Atloria, documentation version work usually starts from the project area where version records are listed. Open your project, go to the documentation versions area, and select the version you want to work on. When you open a version page, that page becomes the main workspace for release-cycle tasks. Instead of moving between unrelated screens, you use the version page to launch the actions tied to that specific release.

[SCREENSHOT: Documentation versions list with one version selected]

On the version page, look for actions related to:

- **Generate** or refresh version content
- **Compare** this version with another version
- **Review** the generated documentation
- **Release readiness** or readiness checks
- **Screenshots** for version-specific image updates
- **Export** options for release deliverables

These actions matter because each one depends on the current version you have open. If you start from the wrong version page, you may compare, review, or export the wrong release.

Different team members usually focus on different actions:

- **Documentation Managers** use **Compare**, **Review**, and **Release readiness** to decide whether a version can move toward publication.
- **Project Administrators** usually handle actions that depend on project setup, connected sources, or release-sensitive settings, especially when generation or export results do not match expectations.

- **Technical Writers** spend most of their time in **Review** and **Screenshots**, checking that pages, wording, and visuals match the intended release.

Treat the version page as the control center for the release cycle. A common flow is to open the version, generate or refresh content, compare it with an earlier version, review the results, update screenshots if needed, and only then move to export. If you need help understanding the broader project workspace around versions, see Managing Project Version Workspaces.

Generating and refreshing version content

Use the **Generate** action on the version page whenever you need Atloria to create or refresh the documentation output for that version. This is usually the first major step after opening a version, especially if the release includes new features, updated screens, changed wording, or newly connected source material.

1. Open the version you want to update.
2. Click **Generate**.
3. Confirm the action if Atloria asks you to proceed.
4. Stay on the version page and watch for progress updates, status changes, or a recent activity timestamp.

[SCREENSHOT: Version page showing the Generate action and generation status]

When you regenerate a version, Atloria refreshes the documentation output tied to that version. From a user perspective, that means the version's generated pages and related release content are brought up to date based on the latest available project information. If your team has made changes since the last run, regeneration helps make sure those changes appear before review.

Rerun generation during the release cycle when:

- source content has changed
- documentation pages were updated
- screenshots or release details need to align with newer content
- comparison results look out of date
- reviewers report missing or stale information

After generation starts, look for visible signals on the version page such as an in-progress state, a refreshed status, or an updated "last generated" style timestamp.

These indicators help you confirm that the version has been rebuilt recently enough for review and export work.

Generation affects everything that comes after it. If you compare versions before refreshing content, your comparison may miss recent changes. If you review too early, you may approve outdated material. If you export before regeneration finishes, your deliverables may not reflect the latest version state. For a deeper walkthrough focused only on creating new version output, see Generating New Documentation Versions.

Comparing versions to review release changes

The **Compare** action helps you check what changed between one documentation version and another. This is one of the most useful tools on the version page because it turns a general “something changed” feeling into a clear release review task. Instead of manually opening pages side by side, you can compare the selected version against an earlier or alternate version and inspect the differences.

1. Open the version page for the release you are preparing.
2. Click **Compare**.
3. Choose the target version you want to compare against.
4. Review the comparison results for added, removed, or changed content.

[SCREENSHOT: Compare view with current version and target version selectors]

Use the comparison view to confirm whether release updates are actually reflected in the documentation. Pay close attention to:

- newly added sections for new features
- removed content that should no longer appear
- changed wording that reflects updated behavior
- UI-related content that should match recent screen changes
- documentation edits that may affect release notes, training, or support material

This step is especially helpful when multiple people have worked on the same release. A Documentation Manager can use the comparison results to decide whether the version is complete enough for formal review. A Technical Writer can use the same results to spot missing explanations, inconsistent terminology, or pages that still reflect the previous release.

Comparison is most valuable at three points in the cycle:

- **After regeneration**, to confirm the latest output is present

- **Before review is completed**, so reviewers know what changed
- **During release readiness checks**, to verify that the release scope matches the documentation scope

If the comparison shows missing updates, return to the version page and run **Generate** again before moving forward. If you want a more focused guide to using comparison views, see Working with Version Comparison Views.

Reviewing content and checking release readiness

Once a version has been generated and compared, move to the review-related actions on the version page. This is where your team decides whether the content is complete, accurate, and suitable for release. In Atloria, review is not just proofreading. It is the point where content quality and release timing come together.

1. Open the version page.
2. Start the available **Review** action or open the review area tied to that version.
3. Check the generated content for accuracy, completeness, and release alignment.
4. Look for readiness signals on the version page before deciding whether the version can move forward.

[SCREENSHOT: Version page showing review status and release readiness indicators]

During review, Technical Writers usually focus on whether the documentation reads clearly and matches the intended user experience. They check that pages are present, instructions make sense, and release-specific updates are included. Project Administrators often help validate details that depend on project configuration or connected content, especially when a page appears incomplete or does not match the current setup.

Documentation Managers typically use the review outcome to make the release decision. Signals that a version is likely ready include:

- the latest generation has completed
- comparison results reflect the expected release changes
- review work has been completed without unresolved issues
- screenshots and supporting materials are current
- no visible blockers remain on the version page

Signals that a version is still blocked include:

- outdated generated content
- missing release changes in comparison results
- incomplete review work
- screenshots that no longer match the interface
- export preparation started before review was finished

If review reveals problems, update the content, regenerate the version, and repeat the comparison if needed. That loop is normal during active release work. For a deeper look at approval and decision-making, see Reviewing and Approving Documentation Versions and Understanding Version Lifecycle and Release Readiness.

Managing screenshots for a version

Screenshots often become the last thing teams notice and the first thing readers question. On the version page, use the screenshot-related action to review or refresh the images tied to that release. This is especially important when the release includes visible interface changes, renamed buttons, updated navigation, or new page layouts.

1. Open the version you are preparing.
2. Select the screenshot-related action from the version page.
3. Review the images associated with that version.
4. Refresh or replace screenshots that no longer match the current content.

[SCREENSHOT: Version page with screenshot action and image review area]

Update screenshots when:

- the interface changed since the previous version
- generated content now describes a different screen flow
- comparison results show UI-related changes
- reviewers flag mismatched images during content review

Screenshot maintenance is not separate from release readiness. If the text says one thing and the image shows another, the version is not truly ready. Technical Writers usually catch these issues first during review, but Documentation Managers often use screenshot completeness as part of the final release decision.

There is also a practical reason to handle screenshots before export. Exported deliverables should reflect the same version state that was reviewed and approved. If screenshots are outdated when you export, your release package may include correct text with incorrect visuals. That creates confusion for reviewers, customers, and internal teams using the exported material.

A good pattern is to refresh screenshots after generation and before final review sign-off. If you update screenshots because of major UI changes, it is worth revisiting the version review once more to make sure the written instructions still match the new images. For more detailed screenshot workflows, see Managing Screenshots for Documentation and Checking Screenshot Readiness Before Version Release.

Exporting version outputs and resolving common release-cycle issues

Use the export-related action on the version page when the version has already been generated, reviewed, and checked for release readiness. Export is the final packaging step for the selected version, so it should happen only after the version page shows the release in its latest approved state.

1. Open the version page for the release you want to deliver.
2. Confirm that the latest generation has completed.
3. Verify that review and readiness work are complete.
4. Click the export-related action for that version.
5. Produce the release deliverable from the current version state.

[SCREENSHOT: Version page showing export action after review completion]

Before exporting, confirm that the version you are looking at is the same one your team reviewed. Then check for signs that the content is current:

- the most recent generation finished successfully
- comparison results match the intended release changes
- screenshots reflect the latest interface
- no review blockers remain visible

A common issue is a mismatch between what the comparison view showed and what appears in the exported output. When that happens, the safest fix is to return to the version page, run **Generate** again, and then repeat the comparison before exporting. This helps ensure the export is based on the latest version content rather than an earlier build.

Another common blocker is outdated screenshots. If the export includes old visuals, refresh the screenshots first and make sure the version still passes review. Incomplete review work is another reason to pause. If a version has not been fully reviewed, exporting too early can spread unfinished content to stakeholders.

Documentation Managers usually own the final export decision, while Project Administrators may step in when export results do not reflect the expected project state. If you need more detail on export controls and validation, see Controlling Version Visibility and Export Options and Validating Export Readiness for Documentation Versions.

Overview

Managing a documentation version in Atloria is a release-cycle workflow, not a single action. The version page brings together the tasks your team uses to move a release from draft output to a shareable deliverable. From that page, you can generate fresh content, compare it with another version, review the results, check release readiness, update screenshots, and export the final output.

The most important idea is sequence. A version usually moves through these stages:

- open the correct version page
- generate or refresh the version content
- compare it against another version to confirm release changes
- review the content for accuracy and completeness
- update screenshots if the interface changed
- export only after the version is clearly ready

Each action supports the next one. Generation creates the latest output. Comparison shows what changed. Review confirms whether those changes are correct. Screenshot updates make sure visuals match the written guidance. Export turns that reviewed version into a release deliverable.

This workflow also helps different roles stay aligned:

- **Technical Writers** focus on content quality and screenshot accuracy
- **Documentation Managers** use comparison, review, and readiness signals to decide whether the version can move forward
- **Project Administrators** help resolve issues tied to setup, connected content, or release-sensitive behavior

If your team skips steps or performs them out of order, problems usually appear later as mismatched exports, missing release updates, or screenshots that no longer fit the text. Keeping work centered on the version page reduces that risk and gives everyone a shared release record for the selected documentation version.

The next document in this series is Managing Version Lists Statuses and Comparisons.

Prerequisites

Before you manage a documentation version across the release cycle in Atloria, make sure you have the right access and enough project context to work from the version page confidently.

You should have:

- an Atloria account and an active sign-in session
- access to the project that contains the documentation version
- permission to open project workspaces and version pages
- a version already available in the documentation versions list
- enough familiarity with the project's release scope to recognize expected changes

It also helps if the following are already in place:

- the project setup is complete
- the documentation content for the release has been updated
- any connected project sources or release inputs your team relies on are current
- reviewers know which version is being prepared for release

If you are still getting started with account access, use Accessing and Registering Your Atloria Account or Signing In to Atloria and Solving Access Problems. If you need to understand how projects are organized before working with versions, see Working with Project Lists and Dashboards and Managing Project Administration from the Project Home.

For the smoothest release-cycle work, gather these inputs before you begin:

- the version you are preparing
- the earlier version you may want to compare against
- any updated screenshots that reflect interface changes
- reviewer feedback or release notes your team is using to validate the version

Without these basics, you can still open a version page, but you may not be able to complete generation, comparison, review, or export in a meaningful order. The next step is Managing Version Lists Statuses and Comparisons.