

Comparing Documentation Versions for Release Decisions

August 26, 2026

Confirming you can compare the right documentation versions

Before you start a release comparison in Atloria, make sure the **newly generated version** and the **baseline version** are both available in the version list for the project you are reviewing. Open the project workspace, go to the area where documentation versions are listed, and check that the release candidate appears as a saved version rather than an in-progress item. The version you compare against should usually be the **currently approved**, **previous release**, or **published** version, depending on your team's release process.

If the version you expect to review does not appear yet, return to the version job or results area and confirm that the generation run has finished. A comparison is only useful when the new version reflects the latest completed output. If you need help checking job results first, use [Managing Version Generation Jobs and Results](#).

It also helps to decide your comparison roles before opening the diff view:

- **Release candidate:** the new version you may publish
- **Baseline:** the version you trust as the current reference
- **Scope:** the set of pages or documentation areas included in the comparison

If Atloria does not show the comparison option, check whether you can open version history and version details for that project. Teams often limit who can review release-level changes. You should be able to access the project workspace, open saved versions, and use the comparison screen.

[SCREENSHOT: Version list showing a newly generated version and a previously approved version ready for comparison]

A quick check at this stage prevents the most common review mistake: comparing the right release candidate against the wrong baseline.

Opening the comparison view for a newly generated version

In Atloria, start from the project's documentation version list or version history area. Select the **newly generated version** you want to review, then choose the **Compare** action. Atloria should prompt you to choose another saved version as the comparison source. In most release reviews, that source is the **previous approved version** or the **currently published version**.

Once the comparison opens, look first at the header area. This is where Atloria shows which version is being compared against which reference version. Before reading any changes, confirm that the header matches your intended review:

- the **new version** is the release candidate
- the **comparison source** is the correct baseline
- the **document scope** matches the pages or content set you meant to review

If Atloria offers more than one comparison layout, switch to the one that best fits the type of review you are doing. A **side-by-side** layout is useful when you want to read old and new content together, especially for rewritten sections or moved text. An **inline diff** is better when you want to scan exact wording changes in a single reading flow.

Use the layout controls to move between these views if a section feels hard to interpret. For example:

- use **side-by-side** for heading changes, larger rewrites, and page structure review
- use **inline diff** for sentence edits, terminology changes, and short additions
- return to the comparison header whenever you need to double-check the selected versions

[SCREENSHOT: Comparison screen header showing source version, target version, and diff layout options]

Do not begin release sign-off until the comparison header clearly shows the correct pair of versions. A clean review starts with the right comparison setup.

Reviewing added, removed, and modified content before release

The comparison view in Atloria is most useful when you read it as a change map rather than a full document. Focus on how the screen marks **added**, **removed**, and **changed** content. New text should stand out as inserted content, deleted text should appear as removed content, and rewritten passages should show as edited rather

than entirely new sections. This helps you separate genuine release updates from simple wording adjustments.

As you review, pay attention to changes that affect how readers move through the documentation. These often matter more than isolated sentence edits. Look closely for:

- **renamed headings** that may change meaning or searchability
- **reordered pages or sections** that affect navigation flow
- **removed topics** that may leave gaps in the release
- **newly generated sections** that need a quality check before publication

If Atloria shows a change summary or lets you narrow the view, use that to focus on high-impact updates first. Start with the sections most likely to affect release readiness, such as policy content, setup instructions, technical reference wording, or other material your team flagged during generation.

You should also review linked content and embedded materials. If links, references, screenshots, or other assets changed between versions, confirm that those changes were intentional. A release candidate can look complete at the text level while still pointing to outdated pages or missing visuals.

Helpful review habits include:

- scanning the summary first for large changes
- opening major edits before minor wording changes
- checking whether deleted content was replaced elsewhere
- confirming that moved content still appears in the right place

[SCREENSHOT: Diff view highlighting added text, removed text, and edited headings]

When you treat the comparison screen as a release review tool instead of a proofreading screen, it becomes much easier to spot changes that affect publish decisions.

Validating that version changes are ready for publication

After you identify what changed, the next step is deciding whether those changes match the release you intended to produce. In Atloria, compare the diff against your release scope, update plan, or requested changes. The goal is not only to confirm that new content appears, but also to catch missing updates that should have been included in the release candidate.

Start by checking whether the expected pages, sections, and edits are present. If a section was supposed to be updated but appears unchanged in the comparison view, that is a signal to investigate further. Open that page directly if needed and verify whether the change was missed, saved elsewhere, or excluded from the version you generated.

If the comparison screen includes document details beyond body text, review those as well. Fields such as **title**, **description**, **labels**, **status**, or **version notes** can affect release clarity just as much as the page content itself. A version may be textually correct but still not ready if these details are outdated or inconsistent.

Use Atloria's review tools wherever they appear in the version workspace. Record your findings in:

- **comments** for section-specific concerns
- **approval notes** for release-level decisions
- **status or tracking fields** when your team uses them to document readiness

This creates a clear trail showing why a version was approved, held, or sent back for revision. That record is especially useful when several reviewers are involved or when publication timing depends on stakeholder sign-off.

[SCREENSHOT: Version comparison with review notes or approval comments visible alongside changes]

A comparison becomes much more valuable when it ends with documented review outcomes, not just a visual scan of edits.

Using comparison results to decide whether to release

A final comparison in Atloria should help you make a practical release decision, not just confirm that changes exist. As you review the diff, decide which findings are **release blockers** and which are minor issues that can wait. This keeps your team focused on what must be fixed before publishing.

Typical blockers in a version comparison include:

- **missing required updates** that were part of the release scope
- **unexpected deletions** that remove important guidance
- **unreviewed generated content** that has not been checked for accuracy
- **wrong version scope** where the release candidate includes unrelated changes
- **broken references or assets** that affect the reader experience

By contrast, some differences are cosmetic. These may include light wording cleanup, formatting shifts, or small layout changes that do not alter meaning. Atloria's comparison view helps you separate these lower-risk edits from changes that affect correctness, completeness, or release timing.

Use the diff during approval discussions with project leads, documentation managers, or other stakeholders who need evidence before publication. Instead of describing changes from memory, you can point to the exact sections that were added, removed, or revised. This is especially helpful when deciding whether the version is ready now or needs another generation or editing pass.

At the end of the review, your decision usually falls into one of three paths:

- **Publish** when the compared changes are complete and accurate
- **Regenerate** when expected generated updates are missing or incomplete
- **Return for revision** when the content needs manual fixes before release

If your team also uses formal review and sign-off steps, pair this comparison with the approval process described in related version review guides rather than treating the diff alone as the final decision.

Fixing common problems when version differences do not look right

If the comparison result in Atloria seems confusing, start by checking the setup rather than assuming the content is wrong. Many comparison issues come from selecting the wrong version pair or opening the diff before the latest version is fully available.

When the **Compare** option is missing, first confirm that the project has at least **two saved versions** available. Atloria cannot compare a version against itself or against an item that has not been saved as a version yet. If versions exist but you still cannot compare them, your access may not include version review or history screens.

If expected edits are missing from the diff, verify that you selected the correct **generated version**. It is easy to open an earlier run with a similar name or date. Also check that the generation work finished successfully before you trust the comparison result. If needed, return to the job results area and confirm the latest output, as covered in [Managing Version Generation Jobs and Results](#).

If the comparison shows too many unrelated changes, the most likely cause is the wrong **baseline version** or the wrong **document scope**. Review the comparison

header and make sure you are comparing the intended release candidate against the correct approved or published reference.

For formatting problems or broken assets in the diff view, open each version on its own and inspect the same page outside the comparison screen. This helps you tell the difference between:

- a real content problem in the version itself
- a display issue limited to the comparison view
- an asset that changed between versions and now needs review

[SCREENSHOT: Comparison header with version selectors highlighted for troubleshooting]

Overview

In Atloria, comparing documentation versions is the step where you decide whether a generated version is actually ready to move toward release. The comparison view helps you review the release candidate against a trusted baseline so you can see exactly what changed before anyone publishes it.

This review is most useful after generation is complete and before final approval. You are looking for more than simple text edits. A strong comparison review checks whether the new version includes the right updates, avoids accidental removals, and stays aligned with the release scope your team intended.

The comparison workflow usually centers on a few key actions:

- opening the **version list** or **version history**
- selecting the **newly generated version**
- choosing **Compare** against a previous approved or published version
- reviewing **added**, **removed**, and **modified** content
- recording review outcomes in **comments**, **approval notes**, or status fields

Atloria's comparison layouts help with different review styles. Use a broader layout when you want to understand page structure and section movement, and a tighter diff view when you need to inspect wording changes line by line. Together, these views help you judge both release completeness and content quality.

This guide focuses on using comparison results to support release decisions. It does not repeat how to manage generation runs or job outcomes. If you still need to confirm

whether the latest version finished processing correctly, refer back to [Managing Version Generation Jobs and Results](#).

By the end of this workflow, you should be able to tell whether the version is ready to publish, should be regenerated, or needs another editing and review pass before release.

Prerequisites

Before you compare versions in Atloria, make sure the project and version workspace are in a reviewable state. You do not need to prepare every release detail in advance, but a few checks will save time and prevent misleading comparison results.

You should already have:

- access to the **project workspace** that contains the documentation set
- permission to open **saved versions**, **version history**, and the **Compare** view
- a **newly generated version** available as the release candidate
- at least one earlier **approved**, **published**, or otherwise trusted version to use as the baseline

It also helps if the generation work for the new version is already complete. If the version is still updating or if the latest results have not appeared in the version list, the comparison may not reflect the final content. Confirming generation status first is especially important when several runs were created close together.

For the smoothest review, have the release scope or expected change list nearby. That gives you something concrete to compare against when you inspect the diff. Without that reference, it is much harder to tell whether unchanged sections are acceptable or whether important updates were missed.

You will get the most value from this guide if you already know how Atloria handles version jobs and saved results. If you need that foundation first, review [Managing Version Generation Jobs and Results](#).

After you finish this comparison workflow and record your release decision, continue with [Monitoring Version Generation Jobs and Results](#) to keep track of follow-up activity and confirm the version stays on course for release.