

Working with Version Comparison Views

August 25, 2026

Opening a version comparison page

To work with a comparison view in Atloria, start in the **Versions** area for the project or document set you are reviewing. You should be able to open both in-progress and approved versions from the version list. If you cannot see the version you need, stop there and confirm that your account can view versions that are still under review as well as versions that have already been approved.

1. Open the project in Atloria and go to the **Versions** list.
2. Find the version you want to review. Use the visible version label and any status badge shown in the list to make sure you are opening the correct release candidate.
3. Open that version, then choose the **Compare** action from the version page.
4. On the comparison page, use the version selectors to choose:
 - the **baseline** version, which is the version you are comparing from
 - the **target** version, which is the version you are reviewing
5. Confirm that the selected versions match the review you intend to perform before you start reading changes.

[SCREENSHOT: Version list with a selected version and the Compare action highlighted]

The comparison page is only useful if the selected versions reflect the real review scenario. In most release reviews, the baseline should be the last approved version, and the target should be the new version under review. If you compare two drafts, you may still see useful differences, but that view may not be appropriate for a final approval decision.

Pay attention to the status badges shown near each selected version. A version marked as **Draft** may still be changing. A version marked as **Approved** is usually the better baseline for release review. If the page shows that you are comparing a draft against another draft, treat the results as a working review rather than final approval

evidence. For background on how those statuses affect release readiness, refer to [Understanding Version Lifecycle and Release Readiness](#).

Reading the differences between two versions

Once the comparison page opens, focus first on how Atloria marks changes. The page is designed to help you spot what was added, removed, or edited without rereading every page from top to bottom. Look for highlighted text, change markers beside sections, and any visual labels that show whether content is new, deleted, or modified.

In many comparisons, Atloria presents content in two panes so you can read the earlier version and the newer version side by side. This layout is best when you need to understand wording changes in context, such as a revised instruction, a renamed heading, or an updated note. In other cases, the page may present a more compact change view that groups edits together. Use whichever layout is available on the page to answer a simple question: what changed, and does that change affect meaning?

The comparison header is also important. Before reviewing the body of the page, check the visible version details shown at the top. These commonly include the version name or number, the current approval status, and the most recent update details. If Atloria shows who last edited the version or when it was modified, use that information to understand whether a change is recent and whether it may have happened after an earlier review.

When reading a changed section, separate content changes from presentation-only changes. A changed heading, paragraph, link, or embedded asset usually points to a meaningful documentation update. By contrast, a small formatting adjustment may not affect release readiness. If the comparison highlights a whole block, read the actual text inside that block before deciding how important the change is.

[SCREENSHOT: Comparison page showing highlighted additions, removals, and edited sections]

Checking changed pages and sections before approval

Before approving a version, narrow your review to the pages that actually changed. In Atloria, this usually means using the comparison navigation or any **changed only** option available on the comparison page. That keeps your attention on pages that differ from the selected baseline instead of forcing you to scan the full document set.

1. Turn on the view that shows **changed pages only** if that option is available.

2. Use the page list or comparison navigation panel to open each changed page one by one.
3. On each page, review the section-level markers to find changed headings, body text, tables, links, and media.
4. Compare those sections against the baseline version and confirm that the update is complete.
5. Repeat until every changed page in the navigation has been checked.

[SCREENSHOT: Comparison navigation panel listing only changed pages]

This review is especially important for release-specific content. Check pages that usually change during a release, such as release notes, setup instructions, policy or process pages, and reference content. If your team maintains technical or reference-heavy documentation in Atloria, make sure those pages reflect the intended release scope and do not contain partial updates.

As you move through changed pages, watch for unintended removals. A deleted navigation entry, a missing cross-link, or a removed content block can be easy to overlook in a broad review. Give extra attention to sections that require approval, pages that are frequently reused across versions, and content that readers depend on to move through the documentation correctly.

If a page appears in the changed list but the update looks incomplete, do not rely on the version summary alone. Open the page and inspect the marked sections directly. The goal is not just to confirm that something changed, but to confirm that the right material changed and that nothing essential disappeared.

Using comparison results to support release decisions

A comparison page is most useful when you treat it as review evidence, not just a reading tool. Start with the change summary at the top of the comparison page, if one is shown. A small, focused set of updates may support a quick approval decision. A large number of changed pages, especially across core documentation areas, may signal that the version needs a deeper review before anyone signs off.

When you evaluate readiness, combine three things on the screen: the visible differences, the version status, and any reviewer comments attached to the version workflow. A version may look complete in the comparison view, but if the status still shows **Draft** or if review feedback is unresolved, it may not be ready for approval. In the same way, a version with only a few changes still needs those changes to match the release plan.

Use the comparison to confirm that expected updates are present. For example, if the release should include updated release notes, revised setup steps, and changes to linked pages, verify each of those items in the comparison before making a decision. If one of those expected updates is missing, the version may need revision even if the rest of the content looks correct.

During review meetings or approval discussions, make sure everyone is looking at the same comparison. Share the comparison page directly inside Atloria or reference it during the discussion so Documentation Managers, Technical Writers, and Project Administrators are reviewing the same baseline and target versions. That avoids confusion caused by people comparing different drafts.

For related guidance on broader comparison and release evaluation, see [Comparing Documentation Versions for Release Decisions](#) and [Managing Version Review Requests and Decisions](#).

Comparing versions efficiently across review cycles

As review cycles repeat, efficiency matters. In Atloria, the fastest way to get a reliable final review is usually to compare the current version against the last **Approved** version rather than against another draft. That gives you a clean view of what will actually change for readers if the new version is released.

1. Set the baseline to the most recent **Approved** version.
2. Set the target to the version currently under review.
3. Turn on page filters or changed-only navigation so unchanged pages drop out of view.
4. Review section markers on each changed page instead of rereading the full document set.
5. Check version details such as modified date and editor information before final sign-off.

This approach helps you catch late edits. If a version was reviewed earlier and then updated again, the modified date and visible editor details can show that the content changed after the previous review round. When that happens, revisit the affected pages instead of assuming the earlier review still covers the current version.

For large document sets, move page by page through the changed navigation rather than scrolling through the entire comparison. This is especially useful when the release touches many areas but only a few sections on each page. Section markers

let you jump directly to the changed content and decide whether the update is acceptable.

If a change has a high impact, such as a rewritten workflow, a removed navigation path, or a major release note update, capture a screenshot for discussion or direct teammates to the exact comparison page in Atloria.

[SCREENSHOT: High-impact difference highlighted in a comparison view for review discussion]

The next step in this workflow is [Managing Project Version Workspaces](#), where you will work with the broader space around each version rather than the comparison view alone.

Fixing common problems in version comparison views

If a comparison page is not showing what you expect, start with the version selectors at the top of the page. Many comparison problems come from reviewing the wrong pair of versions rather than from missing content.

1. If no differences appear, check whether the baseline and target selectors are set to the same version.
2. If expected pages are missing, confirm that those pages were included in the target version you selected.
3. If you cannot complete an approval-related action, check the version status badge and confirm that the version is not still in **Draft**.
4. If the page is difficult to read, switch to the available comparison layout or use the changed-pages navigation to review one page at a time.

If a page you expected to review does not appear in the comparison, return to the version details and confirm that the page is part of that version's release scope. A page that was not included in the target version will not appear as a changed item, even if you expected it to be updated.

If approval decisions are unavailable from the workflow around the comparison page, look at the visible status first. Atloria may allow you to compare versions that are still being prepared, but those comparisons do not always support final review actions. You may also need the right review or approval access before decision controls appear.

When the difference view feels too dense, avoid scanning the whole page at once. Open the changed-items list, select one page, and review the marked sections in

order. This is often the easiest way to separate meaningful content edits from small formatting changes.

For help with status-related blockers, return to [Understanding Version Lifecycle and Release Readiness](#). For approval-specific follow-up, see [Understanding Version Review Feedback and Follow Up](#).

Overview

Version comparison views in Atloria help you answer a specific review question: what changed between one documentation version and another, and are those changes ready to move forward? You use this page when a version is being reviewed for approval, when a release candidate needs validation against the last approved version, or when teammates need a shared view of edits across pages.

The comparison page is centered on two selected versions: a baseline and a target. Atloria then highlights the differences between them so you can review changed pages, inspect section-level edits, and confirm whether the new version includes the updates expected for the release. This is especially useful when a version contains many pages and only some of them changed.

A comparison view is not the same as a full version workspace. It is a focused review screen for examining differences. If you need to understand how statuses move through the release cycle, use [Understanding Version Lifecycle and Release Readiness](#). If you need to work with the broader version area after comparison, the next document in this sequence is [Managing Project Version Workspaces](#).

Use comparison views when you need to:

- review changed pages against an approved baseline
- confirm that release notes and related updates are present
- spot unintended removals before approval
- support approval discussions with visible evidence
- check whether late edits were introduced after an earlier review

[SCREENSHOT: Version comparison page showing baseline selector, target selector, and changed page navigation]

Prerequisites

Before you start comparing versions in Atloria, make sure the basic review conditions are already in place. Comparison views are most useful when the version list is

organized and the release stage is already understood.

You should have:

- access to the project and its **Versions** area
- permission to open the versions you need to compare, including versions that are still under review and versions that are already approved
- at least two available versions to compare
- a clear understanding of which version should be the baseline and which should be the target
- enough familiarity with the release stage to know whether the comparison is for working review or final approval

It also helps if you already know the version statuses used by your team. If you are unsure when to compare a draft to an approved version, review [Managing Version Lists Statuses and Comparisons](#) before continuing.

Use this quick check before opening the comparison page:

What to confirm	Why it matters
The correct project is open	Version lists are project-specific
The baseline version is identified	This gives you a reliable reference point
The target version is identified	This is the version you are evaluating
The version status is visible	Status affects whether the comparison supports approval
The release scope is known	You can verify whether expected pages changed

If you are still preparing the version itself rather than reviewing its differences, start with [Managing Documentation Versions Across the Release Cycle](#) or [Managing Version Lists Statuses and Comparisons](#).