

Generating Documentation Versions for Release Cycles

August 25, 2026

Preparing a Release Cycle for Version Generation

Before you click **Generate Version**, open the release cycle you plan to use and review the details already attached to it. In Atloria, this step matters because the version you generate will follow the release cycle's selected documentation scope and release identity. If the release cycle is missing key details, the version you create may be incomplete or tied to the wrong release.

Start in the project's documentation versions area and open the release cycle record you prepared earlier. If you need help organizing the release before this point, use [Coordinating Version Work Before Release](#). On the release cycle screen, confirm the product, project, or documentation set shown for that cycle matches the material you want to generate. Then check the release identifier or version label already associated with the cycle so you do not create output for the wrong release.

Review the inputs that affect generation. Depending on what your team uses, make sure the release cycle points to the correct source content, branch, or saved content state. Also confirm the version label, default language, and any comparison baseline or review baseline linked to the cycle. If these details are visible on the release cycle or version setup area, verify them before you proceed rather than correcting them after a failed run.

It also helps to scan the existing versions list for that release cycle. Look for duplicate names, older test runs, or a version entry that already uses the same release label. This avoids creating two nearly identical versions and sending reviewers to the wrong one.

Check that you can see the version actions and results area. If you cannot access the release cycle details, generation action, or results view, you may need a Documentation Manager or Project Administrator to grant the right access.

[SCREENSHOT: Release cycle page showing release identifier, source content details, language setting, and versions list]

Creating a New Documentation Version

1. Open the release cycle you want to use, then go to the version management area on that screen. In Atloria, this is where you start a new output for the selected release. Click **Generate Version** to open the version creation form.
2. Fill in the version details carefully. Use the release-ready version name your team expects reviewers to see. Check the source content reference shown in the form and make sure it matches the content snapshot, branch, or selected source for this release cycle. If Atloria shows an output scope, choose the scope that matches what you want to generate for this release. If a publication target appears, review it before continuing so the generated version is tied to the correct destination.
3. Review the generation options before submitting. If Atloria gives you a choice between rebuilding everything and generating changed content only, select the option that fits your release. A full rebuild is usually the safer choice when the release includes broad structural changes, navigation updates, or language changes. A changed-content option is more suitable when you only need to refresh recent edits and want a faster run.
4. Submit the form by clicking the action button that confirms generation. After you submit, Atloria creates a new version entry in the versions list for that release cycle. Watch the initial status shown beside the new entry. It may appear as waiting, queued, or another early processing state before the actual build begins.

Use the new version entry as your main tracking point from this stage onward. The version name, release label, and status shown there help you confirm that the correct release cycle is now being processed.

Field or option	What to check	Why it matters
Version name	Matches the release your team expects	Prevents confusion during review
Source content reference	Points to the correct content state	Ensures the right material is generated
Output scope	Includes the intended pages or sections	Avoids missing documentation
Publication target	Matches the intended destination if shown	Keeps release output aligned
Generation option	Full rebuild or changed content only	Affects speed and completeness

[SCREENSHOT: Generate Version form with version name, source content reference, scope, and generation options]

Monitoring Generation Progress

1. After submitting the version, open its details page or stay in the release cycle activity area where Atloria shows processing updates. The status will usually move through early states before the final result appears. Keep an eye on the status text first, because that is the fastest way to tell whether the run is waiting, actively processing, or finished.
2. Check the timestamps shown for the run. If Atloria displays **Started**, **Completed**, or similar time markers, compare them with the current status. A version that has a recent start time and changing activity is still moving. A version with no new event for a long period may need closer attention.
3. Review any progress markers shown during build and validation. Some version screens show separate stages, such as generation, validation, or output preparation. These stage updates help you understand whether Atloria is still working through the run or whether it stopped before the final checks.
4. Refresh the page or reopen the results view when you expect new information. This is especially useful if you are waiting for build logs, warning messages, or downloadable output links to appear. If the latest event time changes and new messages appear, the run is still updating.

When you are deciding whether a run is active or stalled, compare three things together:

- The current status text
- The elapsed time since the last visible event
- Whether new logs or stage markers have appeared

If the status still says it is processing but nothing has changed for an unusually long time, note the last event shown before deciding to retry. Do not assume a long run has failed if Atloria is still updating timestamps or adding new messages.

[SCREENSHOT: Version details page showing status, started time, completed time, and generation activity]

Reviewing Results Before Publication

1. Open the completed version entry and read the final status first. In Atloria, the result may show a successful completion, a completed run with warnings, or a failed run. Treat these outcomes differently. A successful result usually means you can move into review. A warning result means the version exists, but you still need to inspect what was excluded or flagged. A failed result means you should stop and fix the issue before handoff.
2. Review the output attached to the version. Look for preview content, generation logs, validation messages, and version details saved with the release cycle. These items tell you whether the generated version matches the release you intended to produce.
3. Check the structure of the generated documentation. Confirm that the navigation looks complete, the expected page count is present, language variants appear where needed, and linked assets such as related content or supporting files were included. If your team expects a specific set of documents for the release, compare the generated output against that expectation before sending it to reviewers.
4. Use the warnings and validation messages as a decision point. Missing pages, broken links, excluded assets, or incomplete language coverage may not block generation, but they can still make the version unsuitable for review or publication.

A version is usually ready for reviewer handoff when:

- The final status is successful, or warnings are minor and understood
- The preview reflects the intended release content
- Navigation and page coverage look complete
- Language and asset coverage match the release plan
- The version metadata matches the release cycle

If anything important is missing, run another generation pass after corrections instead of asking reviewers to work from an incomplete build.

[SCREENSHOT: Completed version results showing final status, preview link, validation messages, and metadata summary]

Managing Regeneration and Version Updates

1. If you fix content, adjust release details, or correct version information after a run, return to the existing version entry or use the release cycle action menu to start generation again. In Atloria, regeneration is the quickest way to refresh output

when the release itself has not changed, but the generated result needs to be updated.

2. Decide whether to reuse the current version entry or create a separate version record. Reuse the existing entry when you are correcting the same release candidate and want one clear record for the latest output. Create a separate version record when you need to preserve an earlier run for comparison, testing, or approval history, especially if the release label or source has meaningfully changed.
3. Before rerunning generation, update any version fields that no longer match the intended release. This may include the version label, source content reference, selected scope, or other release-linked information visible on the version or release cycle screen. Make these updates first so the next run reflects the correct release candidate.
4. After regeneration finishes, confirm that the newest result is the one attached to the review and publication workflow. Check the latest timestamp, final status, and visible output links. If Atloria shows multiple runs or multiple version entries, open the one with the most recent successful result before sending it forward.

Use extra care when several similar version names appear in the same release cycle. Reviewers can easily open the wrong entry if an older run remains visible. A clear label and a quick check of the latest generation time help prevent that mistake.

For comparing different outputs before deciding which one to keep, pair this step with [Working with Version Comparison Views](#).

[SCREENSHOT: Release cycle versions list with multiple runs, updated labels, and latest generation result highlighted]

Fixing Common Version Generation Problems

If a version does not generate as expected, start with the visible information on the release cycle, version form, and results screen. Atloria usually gives enough on-screen detail to narrow the problem before you ask someone else to investigate.

- **Generation does not start**
 - Reopen the release cycle and confirm it is the correct one for this release.
 - Check the version form for missing required fields such as the version name, source content reference, or other visible release details.

- Make sure you can see the generation action and results area. If those options are missing or unavailable, your access level may not allow version generation.
- **Generation stays in progress too long**
 - Open the version details and look at the latest event or most recent activity line.
 - Compare the last update time with the current time.
 - Refresh the page and check whether the status, logs, or stage markers have changed.
 - If nothing updates and the run appears stuck, retry the generation from the version entry or release cycle actions.
- **Build completes with warnings**
 - Read the validation output carefully.
 - Look for missing pages, unresolved links, excluded assets, or incomplete language output.
 - Decide whether the warnings are acceptable for internal review or whether you should correct the content and regenerate first.
- **Build fails entirely**
 - Open the error details and generation log attached to the version.
 - Check whether the source content reference is valid and whether the release cycle details still match the intended output.
 - Review the version label and release data for mismatches that could cause the wrong content to be pulled into the run.

When repeated failures affect the same release cycle, compare the current setup with a previously successful version in the same project. That often reveals a changed field or release detail that needs correction.

[SCREENSHOT: Failed generation result with error details, warning messages, and retry action]

Overview

Generating documentation versions in Atloria is the step where a prepared release cycle becomes a reviewable documentation output. You begin from a release cycle, create a version entry with the correct release details, submit the generation request,

and then follow the status until Atloria finishes building the result. From there, you review the output, decide whether warnings are acceptable, and either move the version forward or regenerate it after fixes.

This workflow fits after your release planning and pre-release coordination work. If you have already organized the release content, confirmed ownership, and aligned the version timeline, generation becomes much more predictable. That earlier preparation is covered in [Coordinating Version Work Before Release](#).

The screens that matter most in this process are:

- The release cycle record
- The versions list for that release cycle
- The **Generate Version** form
- The version details page
- The generation results area with status, logs, and validation output

As you work through generation, focus on three questions:

- Is this version tied to the correct release cycle and release label?
- Did Atloria generate the expected pages, languages, and linked assets?
- Is the final result clean enough to send into review and publication steps?

You do not need to solve every release decision during generation. The goal here is to produce the right version output and confirm it is accurate enough for the next stage. Once the version has been generated and checked, you can move on to status handling and release readiness decisions in [Coordinating Version Statuses and Release Readiness](#).

Prerequisites

Before generating a documentation version in Atloria, make sure the release cycle and project workspace are already prepared. This guide assumes you are not starting from a blank project and that the release has already been planned.

You should have the following in place:

- Access to the project's documentation versions area
- Permission to open release cycles and create or rerun versions
- A release cycle that already identifies the intended release
- The correct source content reference or saved content state for that release

- A version label your team plans to use for review and release
- The expected default language, if your project uses language-specific output
- Enough familiarity with the project workspace to recognize the correct version entry and release record

It is also helpful to confirm these items before you begin:

- Existing version entries do not already use the same release label
- The release cycle points to the correct documentation set or project content
- Any review baseline or comparison baseline tied to the release is already selected
- The team has finished the content changes intended for this release candidate

If you still need to organize the project workspace or locate the right release area, see [Managing Project Version Workspaces](#). If you need a broader refresher on how versions move through the release cycle, use [Managing Documentation Versions Across the Release Cycle](#).

Once these prerequisites are in place, you can generate a version with much less risk of duplicate labels, incomplete output, or review confusion.