

Monitoring Version Generation Jobs and Results

August 25, 2026

Understanding the version generation workflow

In Atloria, version generation starts from the project area where you manage documentation versions. Open your project, go to the version management screen, and use the **Versions** list to find the version you want to work with. Each version appears as its own row, and that row is where you can usually tell whether the version is only created as a record or whether generated output already exists. If a **Generate**, **Build**, or similar action is available for the version, that means the version can be processed into usable output.

It helps to separate two ideas: creating a version and generating a version. A version record is the entry that identifies the release you want to manage. It may appear in the **Versions** list with its name, status, and timestamps even before any output has been produced. Generation is the step that creates the actual result you can review, compare, publish, or move forward in the release process. If a version exists but has no generated output, you may see that it has no current build result, no output link, or no recent completion status.

Once you start generation, Atloria tracks that work as a job. The job usually moves through a clear lifecycle:

- **Queued**: the job has been submitted and is waiting to start
- **Running** or **In Progress**: Atloria is actively generating the version
- **Completed**: the output finished successfully
- **Failed**: the job stopped because of a problem

You can usually see this state in the version row itself, and in some cases in a job details panel or history view. After the build finishes, watch for clear result indicators such as:

- a success badge or completed label
- a failed badge or error message
- a **Completed at** timestamp

- a preview, output, or result link
- a logs or details link for reviewing what happened

If you need help deciding whether a finished result is good enough for release review, use [Comparing Documentation Versions for Release Decisions](#) alongside the job status details.

Starting a new version build

1. Open the project that contains the documentation version you want to generate. From the project workspace, go to the version management area and open the **Versions** list.
2. Find the version you want to build. Check the row for its current status so you do not accidentally rebuild the wrong entry. This is especially important when several versions have similar names or were created close together.
3. Select the version and click the available build action. Depending on what your Atloria workspace shows, this may appear as **Generate Version**, **Start Build**, **Generate**, or a similar action attached to that version row or details view.
4. Review any confirmation window that appears. Pay attention to the selected version identifier and any output target shown in the dialog. If Atloria displays a summary before submission, confirm that you are building the correct version and not an older draft.
5. Click the confirmation button to submit the job. Right after submission, the screen should change to show that the build has started. In most cases, you will see one of these signs:
 - the version status changes to **Queued**
 - the version row changes to **In Progress** or **Running**
 - a new job entry appears in the job list or activity area
 - the build action becomes temporarily unavailable while the job is active
6. Stay on the version screen for a moment and make sure the new state appears. If the row still looks unchanged, refresh the page once and check again. You want to confirm that Atloria accepted the request before moving on.

[SCREENSHOT: Versions list showing one version row changing from ready to queued after clicking Generate Version]

If you are already managing multiple version runs at once, it is useful to compare the new job with earlier attempts in the same version history. That makes it easier to tell whether you started a fresh run or are still looking at an older result.

Watching build progress in real time

1. After starting the build, stay on the version screen and watch the status area for changes. Atloria may show the current state directly in the version row, in a side panel, or in a job details view. The most common states to watch are **Queued**, **Running**, and **Completed**.
2. If a progress area or activity feed is available, open it to follow the build as it moves forward. This is where you can tell whether the job is still waiting in line or actively processing the version output.
3. Check the timestamps shown with the job. These are often the quickest way to tell whether the build is moving normally:
 - **Started at** tells you when processing began
 - **Updated at** helps you see whether activity is still happening
 - **Completed at** confirms the run has finished
4. Open the job details or logs panel when you need more than the status badge. If Atloria exposes build stages, you may see the version move through steps such as content collection, rendering, packaging, and publication. Even if the screen does not show every stage, the details view can still help you tell whether the job is advancing or has stopped at one point.
5. Watch how the page updates. Some Atloria screens refresh status automatically, while others may need a manual reload before the latest state appears in the list. If the **Updated at** time is not changing and the badge still looks old, refresh the page or reopen the version details to load the newest information.

[SCREENSHOT: Job details panel with status badge, timestamps, and log entries]

A job that is healthy usually shows movement: the status changes, the **Updated at** value advances, or new detail lines appear in the logs. If none of those change for an extended period, treat the run as possibly stalled and review the job details before starting another build.

Interpreting completed, failed, and partial results

When a version build succeeds in Atloria, the result is usually easy to spot. The version row or job details view shows a **Completed** status, a success-style badge, or a finished timestamp such as **Completed at**. You should also expect to see some sign that output is available, such as a preview link, generated result link, published indicator, or another action that lets you open what was produced. A successful job is not just one that stopped running; it is one that finished and left behind usable output for review.

A failed result looks different. Instead of a completion badge, the job list or details panel shows a failure state such as **Failed**, an error label, or a visible failure message. In many cases, the output link is missing because Atloria did not finish creating the version result. If you open the job details, look for the point where progress stopped. That is often more useful than the top-level failure badge because it tells you whether the problem happened early or near the end of the run.

You may also run into results that are neither fully successful nor fully usable. For example, a job may finish but still leave you with outdated output, warnings, or only part of the expected result. In the interface, this often shows up as a completed job without the links or labels you expected for the latest run. If the version appears finished but the generated files, preview action, or release-ready indicators are missing, treat it as incomplete until you confirm the output.

To make sense of repeated runs, compare the job history carefully. Focus on:

- the most recent timestamp
- the newest status badge
- whether the latest run has output links
- whether older runs show different results

The latest job result matters most. A version may have an older successful run and a newer failed run, or the other way around. Always match the status to the newest timestamp before deciding what to do next.

Responding to generation outcomes

1. When a build finishes successfully, open the result from the available preview, output, or generated version link. Confirm that the version you built is the one now available and that you are not viewing an older run by mistake. Check the version label and the latest completion time before you continue.

2. Review the generated output in the project workspace. At this stage, you are confirming that Atloria produced something usable, not just that the status says **Completed**. If the version is part of a release decision, compare what you see here with your expected changes and with any earlier comparison work you already completed.
3. If the build fails, open the job details or logs panel right away. Look for the stage where the run stopped and note whether the same point appears in earlier failed attempts. This helps you decide whether the issue is temporary or tied to the version content or setup.
4. Use **Retry**, **Regenerate**, or **Run Again** when that action is available and you want Atloria to attempt the build again. A retry should create a new job entry instead of replacing the previous one, so you can compare attempts side by side. That history is useful when you need to see whether the same failure keeps happening.
5. If the result looks outdated, return to the **Versions** list and verify that you selected the correct version before you started the build. It is easy to launch a run for an older version with a similar name. Once you confirm the correct version, start a fresh build so the latest documentation changes are included.
6. After you have a stable result, move toward release preparation rather than repeating the same monitoring steps. The next stage is covered in [Creating Release Ready Documentation Versions](#).

[SCREENSHOT: Completed version row with preview link and a separate earlier failed job in history]

Fixing common version generation problems

A few build problems show up often in Atloria, and you can usually narrow them down from the version list and job details without leaving the project workspace.

- **The build stays in queued status for too long**

First, check whether another version generation job is already running. If Atloria is processing one job at a time, your version may simply be waiting its turn. Look at the job list, activity area, or other version rows to see whether a different run is active. If nothing appears to be moving, refresh the page and confirm that the queue is still updating.

- **The build shows completed but no output is available**

A completed badge by itself is not enough. Open the latest job result and confirm

that the output link, generated files section, or preview action exists for that same run. If the version row says **Completed** but there is no result to open, compare the latest timestamp with older jobs to make sure you are not looking at a leftover success from a previous attempt.

- **The build fails every time you retry**

Compare the messages in the logs across multiple attempts. If the same stage fails each time, the issue is likely tied to that version rather than a one-time interruption. Repeated failures with the same message usually point to a version-specific problem, while different messages across attempts may suggest a broader configuration or timing issue.

- **The version list status does not match the latest job result**

Refresh the page or reopen the job details view. Sometimes the list view lags behind the latest run, especially after a retry. Always trust the newest timestamp in the detailed job view over an older badge that may still be showing in the list.

[SCREENSHOT: Version history showing multiple retries with timestamps and different statuses]

If you are managing several versions at once, keep your troubleshooting focused on one version row and its newest job history. That prevents confusion between an older successful run and a newer failed one.

Overview

This guide focuses on the day-to-day monitoring work that happens after you start a version generation job in Atloria. The main screen you will use is the project's **Versions** area, where each version row shows whether output exists, whether a build is waiting or running, and whether the latest attempt finished successfully. From there, you can open job details, review timestamps, and decide whether the version is ready for review or needs another run.

The most important idea is that a version entry and a generated result are not the same thing. You can have a version listed in Atloria with no usable output yet. That is why this guide keeps returning to visible result markers such as **Queued**, **Running**, **Completed**, **Failed**, completion timestamps, and output or preview links. Those are the signals that tell you whether the version is only recorded in the list or has actually been generated.

This guide does not repeat the comparison work covered in [Comparing Documentation Versions for Release Decisions](#). Instead, it picks up after that stage

and shows you how to monitor active jobs, read the latest result correctly, and respond when a run succeeds, stalls, or fails. If you are handling multiple attempts for the same version, the job history and timestamps become especially important because Atloria keeps each run as a separate result rather than replacing the older one.

Use this page when you need to answer practical questions such as: Did the build actually start? Is it still moving? Which run is the newest one? Is there output to open? Should I retry this version or move it forward?

Prerequisites

Before you follow the steps in this guide, make sure you have the basics in place inside Atloria:

- You can sign in and open the project workspace that contains the documentation version you want to monitor.
- The project already has at least one entry in the **Versions** list.
- You have access to the version management area where build actions such as **Generate Version**, **Start Build**, **Retry**, or **Run Again** are available.
- You can open the version row, job details view, or logs panel for recent generation attempts.
- You already understand how to compare version output and review release differences. If you need that earlier step, use [Comparing Documentation Versions for Release Decisions](#).

It also helps to know what you are trying to confirm before you start watching a build. For example:

- whether you are building a brand-new version
- whether you are rerunning a failed attempt
- whether you are checking that recent content changes are included
- whether you need a finished result for release review

If several people work in the same project, take a moment to confirm which version should be built and whether another generation job is already running. That small check can save time when the **Versions** list contains multiple similar entries.

You do not need to know any behind-the-scenes processing details to use this guide. Everything here is based on what you can see directly in Atloria: version rows, status badges, timestamps, result links, retry actions, and job details. Once those are

familiar, you can move on to preparing the finished version for release in [Creating Release Ready Documentation Versions](#).