ATLORIA

Using Analytics Audit and Exports in Release Operations

August 25, 2026

Preparing the release workspace and confirming access to analytics, audit history, and exports

Before you start validation, make sure you are working from the correct project and release context in Atloria. If you have already moved content into a versioned release, use the same release candidate you prepared in Moving Document Updates into Versioned Releases. Open the project workspace, then confirm the release header shows the version or release label you intend to review. This matters because analytics, activity records, and exported files are only useful if they match the exact release you plan to share.

1. Open the target project in Atloria and go to the release or version you want to validate.
2. Check the page header for the active release label, version name, or project context so you know you are reviewing the right content set.
3. Open the **Analytics** area for that project or release and confirm you can view the dashboard.
4. Open the **Audit History** or activity log view and make sure entries load for the same project.
5. Find the **Export** action in the project view, release view, or related toolbar and confirm export options are available.
6. Decide the scope of your review before you continue. For example, you may be validating one documentation version, one release branch, or one published set.
7. Confirm the export outputs you need are available from the menus you can access, such as a metrics export, an audit log export, or a documentation package export.

[SCREENSHOT: Project release view showing the release label in the header, with Analytics, Audit History, and Export actions visible]

If one of these areas is missing, stop and confirm you are in the correct workspace. In Atloria, release checks are most reliable when the same project and release context is used across all three areas.

Reviewing project analytics to confirm the release is ready to share

Use the Analytics view to judge whether readers are finding and using the documentation the way you expect. In Atloria, the **Analytics & Insights** screen is available from the admin area, and project analytics may also be available from the project context depending on your workspace. When you open analytics, start by checking the filters at the top of the screen. If there is a date range, release filter, version selector, or similar control, set it before reading any numbers. That keeps your review focused on the release candidate instead of older project activity.

1. Open **Analytics & Insights** from the relevant project or admin workspace.
2. Set the date range, release filter, version selector, or other visible filter so the dashboard reflects the release you are validating.
3. Review the summary cards and charts for signs that the updated documentation is performing as expected.
4. Look closely at reader-facing patterns such as page views, search activity, completion trends, or pages with noticeably low engagement.
5. Compare newly updated pages with pages that were not changed in this release. A large difference can point to missing links, unclear titles, or content that still needs revision.
6. Narrow the dashboard using any available filters, segment controls, or module selectors so you are reviewing only the documentation set included in this release.
7. Note any unusual results that may need a second check in the audit trail, such as a sudden drop in engagement or unexpected search behavior.

[SCREENSHOT: Analytics dashboard with date range and release filters highlighted]

Treat analytics as evidence of release readiness, not just a traffic report. If readers are not reaching important pages, or if search behavior suggests they cannot find updated content, pause before approving the release. Those signals often become easier to explain once you compare them with the activity records in the next step.

Checking audit history to verify what changed in the release

After reviewing analytics, switch to the **Audit History** or activity log for the same project. This view helps you confirm exactly what changed, who changed it, and when the change happened. The most useful audit review is narrow and deliberate. Instead of scanning the full timeline, filter the records so they match the same release, date range, and content scope you used in analytics. That makes it much easier to verify whether the final approved edits are the ones actually included in the release.

1. Open the **Audit History** or project activity log from the same project or release context.
2. Apply filters such as release version, date range, user, or content item so the list matches the release candidate under review.
3. Review entries for important release actions, including page edits, metadata changes, permission updates, publication events, and export-related actions if they appear in the log.
4. Open individual audit entries when more detail is available and check the timestamp, the user name, and the affected content item.
5. Confirm that the latest approved changes appear in the timeline and that they happened before the release was prepared for sharing.
6. Look for unexpected activity after final review, such as a late page edit, a visibility change, or a permission update that could affect what readers see.
7. Record any entries that need clarification before release approval.

[SCREENSHOT: Audit History view filtered by date range and release, with entry details panel open]

A clean audit trail should support the release story you expect: the right pages were updated, the right people made the changes, and no unplanned edits appeared afterward. If the timeline does not match the release you reviewed in analytics, adjust the filters and check the project header again before moving on.

Cross-checking analytics and audit records before approving the release

This is the point where release validation becomes a decision rather than a review. Analytics tells you how the documentation is being used or discovered, while the audit trail tells you what changed behind that result. In Atloria, you should compare both views using the same release label, page title, and date range whenever possible. If you switch between broad project data and release-specific activity, it becomes much harder to judge whether a problem belongs to the current release or to older content.

1. Keep the same project and release selected while moving between **Analytics** and **Audit History**.
2. Match unusual analytics patterns to related activity records. For example, if a page suddenly loses engagement, check whether it was renamed, moved, hidden, or replaced in the audit timeline.
3. Use the same page title, release label, or documentation item name in both views so you are comparing the same content.
4. Check whether search activity points to pages that were recently edited or replaced.
5. Confirm that the pages receiving attention are the pages you intended to promote in the release.
6. Decide whether the release is ready based on combined evidence: analytics should support discoverability and usefulness, and audit history should confirm the approved change set.
7. Write down any mismatch that requires follow-up before sharing the release.

A few examples of discrepancies worth pausing on:

- Analytics shows traffic going to an older page, but the audit trail shows a replacement page was published.
- Search activity increases for a topic, but the audit trail shows the related page title or metadata changed recently.
- A key page has very low engagement, and the audit trail shows it was moved or updated late in the cycle.
- Audit records show permission or visibility changes that may explain why readers are not reaching the intended content.

When both views support the same conclusion, you have stronger evidence that the release is ready to share with stakeholders.

Exporting validation evidence for stakeholders and release records

Once the release checks are complete, export the evidence you need for review, approval, or archiving. In Atloria, use the **Export** action from the project view, analytics view, or related release area, depending on where the export options are available to you. The most important part of this step is applying filters before you export. If you export first and filter later, you may end up sharing files that include unrelated projects, older releases, or extra activity that confuses reviewers.

1. Open the **Export** menu from the current project, release, or analytics view.
2. Choose the export type that matches your review need, such as a metrics export, an audit log export, or a packaged documentation export.
3. Before generating the file, apply the release-specific filters shown on screen, including the version, date range, page selection, or activity scope.
4. Create separate exports when stakeholders need both performance evidence and a change record.
5. Download each file and confirm the contents reflect the release candidate you reviewed.
6. Save the files using the release identifier shown in Atloria so each export can be traced back to the correct version.
7. Share the exported files with reviewers or store them with your release records according to your team's process.

[SCREENSHOT: Export menu open with release-specific filters visible before download]

A simple naming pattern helps when you are handling multiple release candidates. Include the visible release label from Atloria in each saved file name so reviewers can match the export to the exact candidate under discussion. If you are exporting both analytics and audit evidence, keep them as separate files unless your team has a standard package format for release reviews.

Fixing common issues when analytics, audit data, or exports do not match the release

When the numbers, activity records, or exported files do not line up, start with the visible filters and the release label in the header. In Atloria, mismatches usually come from reviewing the right project with the wrong date range, or the right release with the wrong export scope. Before you assume the release data is incorrect, confirm that every screen is pointed at the same version and time window.

Use the checks below to resolve the most common problems:

- **Analytics does not reflect the current release**
  - Reopen the **Analytics** view and verify the selected date range.
  - Check for a release filter, version selector, or project scope control.
  - Make sure you are not looking at full project history when you only need the current release.

- **Audit history is missing expected changes**
  - Review the filters in **Audit History**.
  - Expand the date range if the final edits happened earlier than expected.
  - Check whether the log is filtered to the wrong user or content item.
- **Exported files contain too much or too little information**
  - Return to the source screen and reapply the version, page, and date filters before exporting again.
  - Confirm you selected the correct export type for what you need to prove.
  - If you need both performance data and change records, export them separately.
- **Stakeholders question whether the export is current**
  - Compare the export timestamp with the latest visible audit entry.
  - Check the active release label in the project header.
  - Regenerate the export if the file was created before the most recent approved change.

[SCREENSHOT: Side-by-side view of release header, filters, and export timestamp for validation]

If the mismatch remains after you verify filters and release context, repeat the review from the project header outward: release label first, then analytics filters, then audit filters, then export settings. That order usually reveals where the scope changed.

Overview

This document focuses on one release task in Atloria: proving that a documentation release is ready to share. Instead of relying on a single approval signal, you use three connected areas of the workspace together:

- **Analytics** to understand how readers are finding and using the documentation
- **Audit History** to verify what changed, who changed it, and when
- **Export** actions to capture evidence for stakeholders, release records, or approval discussions

This workflow is most useful after your content has already been prepared as a release candidate. If you still need to move edits into a versioned release, return to Moving Document Updates into Versioned Releases before using the steps here.

The release validation process in Atloria is not just about checking whether pages exist. You are confirming that the intended pages are the ones readers can discover, that the final approved edits are the ones recorded in the activity timeline, and that you can produce clear evidence for anyone reviewing the release. This is especially helpful when a documentation manager, project administrator, or stakeholder needs a record of why a release was approved.

You will work across project and admin views that expose **Analytics & Insights**, **Security & Audit** or **Audit History**, and **Export** actions. Some workspaces may show these tools in different navigation areas, but the goal stays the same: keep the release scope consistent across all three. When the release label, filters, and exported files all match, you have a dependable release record that can support publication, review, and follow-up decisions.

Prerequisites

Before you begin, make sure the following conditions are true in Atloria:

- You can sign in and open the target project workspace.
- The documentation release already exists as a version, release candidate, branch-based release, or published set that you can identify by name in the header.
- You know which release you are validating and which pages or documentation area belong to that release.
- You can open the **Analytics** area relevant to your project or admin workspace.
- You can open **Audit History**, the activity log, or the related audit view for the same project.
- You can access the **Export** action from the project, release, or analytics context you are using.
- You know what evidence your reviewers need, such as analytics results, audit records, or a documentation package export.

It also helps to have these details ready before you start:

Item to confirm	Why it matters
Release label or version name	Keeps analytics, audit review, and exports tied to the same candidate
Date range for review	Prevents older activity from being mixed into current release checks
Pages or content scope	Helps you focus on the exact documentation set being approved

Item to confirm	Why it matters
Reviewer expectations	Tells you whether to export metrics, audit records, packaged docs, or all three

If you need help getting into Atloria or moving through sign-in and session access, use Signing In to Atloria and Solving Access Problems. After you finish the validation workflow in this guide, continue with Coordinating Technical Reference and Authored Content Workflows.