

Validating Audience Targeted Publishing

August 25, 2026

Confirming the release is ready for audience-targeted validation

Before you start testing, make sure you are checking a release that is already available in Atloria's published documentation view. If the version is still being reviewed or has not been published yet, the public page and version selector will not reflect the final audience experience. If you need help with the publishing flow itself, use [Publishing Documentation for Specific Audiences](#) first, then return here to validate the published result.

Gather the exact pages you plan to test and confirm each one has audience targeting applied in its publishing setup. In practice, that means you should already know which audience each page is meant for and which parts of the page are expected to change. This is especially important for pages that include audience-specific notices, different screenshots, alternate links, or navigation entries that only appear for one audience.

Before opening the public site, prepare a simple validation table so you can compare expected results against what you actually see.

Page	Expected Audience	Expected Version	What Should Be Visible
Getting Started page	Selected audience	Published release	Main heading, audience-specific callout, correct links
Setup page	Selected audience	Published release	Matching navigation item, expected body sections
Troubleshooting page	Selected audience	Published release	Correct related links and visible notices

Also confirm you know:

- The standard public URL for the page
- The versioned documentation URL for the same page
- The audience-specific path pattern your team uses for published content

- Whether the audience is selected through a visible switcher, a dedicated path, or another public-facing route pattern
- Which browser session you will use for testing

[SCREENSHOT: validation notes showing page name, audience, version, and expected visible elements]

Reviewing how the targeted page should appear before testing

A useful validation pass starts with knowing exactly what should change on the page. Open your source page in Atloria and review the visible content structure before you test the public output. Focus on the page title, main heading, section headings, notices, tabs, screenshots, related links, and any right-side table of contents entries that may appear or disappear for a specific audience. If you do not define these expected differences first, it becomes much harder to tell whether the published page is correct.

Create a side-by-side expectation for two views of the same page:

- The canonical public view
- The versioned audience path for the release you are validating

For each view, note the exact items you expect to see. For example, the page may keep the same H1 but show a different warning box, a different screenshot, or a different internal link for one audience. In some cases, the page should also change how it appears in left navigation, breadcrumbs, search results, or related-links sections. If a page is meant to be hidden from a certain audience, record that too so you can verify it does not appear where it should not.

Define your pass criteria before testing. A page should count as valid only when all of the following are true:

- The correct audience-specific body content is visible
- The correct version is shown in the version selector
- Content meant for other audiences does not appear
- Breadcrumbs and navigation match the intended page
- Internal links open the correct destination without losing the audience context

[SCREENSHOT: published page with callouts identifying H1, breadcrumb, left navigation, and targeted content block]

This preparation step reduces guesswork and gives you a clear checklist when you move into public and versioned validation.

Validating the page in the public documentation view

1. Open the public page in a clean browser session. Use an incognito or private window so previous browsing does not affect what you see. This matters when Atloria remembers a prior audience selection or loads an older page state from cache.
2. Confirm that you landed on the correct page. Check the browser address, page title, H1 heading, breadcrumb trail, and the matching entry in the left navigation. These should all point to the same published page. If the breadcrumb or navigation label does not match the page you intended to test, stop there and record it as a navigation issue.
3. Review the audience-targeted content in the body of the page. Look for the exact sections you listed earlier: audience-specific paragraphs, notices, screenshots, tabs, badges, or links. Compare what is visible against your validation notes. If a callout should appear for one audience, make sure it is present and placed in the expected section of the page.
4. Check for content leakage. Scroll through the full page and look for anything that should not be public for the selected audience. This may include alternate instructions, hidden screenshots, links intended for another audience, or extra sections that should have been excluded from the published view.
5. Test the page links. Open the most important links from the page body, breadcrumbs, related links, and left navigation. Make sure each one opens the correct public destination and keeps you in the same audience context instead of sending you to a generic or mismatched page.

[SCREENSHOT: public documentation page showing breadcrumb, left navigation, and audience-specific content area]

If the page passes these checks in the public view, you are ready to compare it against the versioned audience path.

Checking versioned audience paths and version switching behavior

1. Open the versioned URL for the same page you just tested in the public view.
Once the page loads, check the version selector immediately. It should show the exact release version you planned to validate. If the wrong version appears by default, note that before testing anything else.
2. Review the address and confirm the page follows the expected audience-specific route pattern for that version. The versioned page should not drop you onto a generic page, a different audience path, or a different document entirely. Compare the page title, H1, breadcrumb, and left navigation with your validation notes to confirm you are still looking at the same content in the correct release.
3. Switch between available versions using the version selector. After each change, watch for two things: whether the page stays on the same document, and whether the audience-targeted content updates correctly for that version. A correct result keeps the page context intact while changing the release version. An incorrect result may send you to the wrong page, remove audience-specific content, or show content from another audience.
4. Compare the versioned page with the standard public page. For the same audience, the visible headings, notices, links, and content blocks should remain consistent unless your team intentionally changed them between versions. If the public page shows one callout and the versioned page shows another for the same release, record that mismatch.

Use this comparison checklist while switching versions:

Checkpoint	Public View	Versioned View
H1 heading	Matches expected page	Matches same page
Version label	Current published state	Selected release shown
Audience-specific content	Correct	Correct
Internal links	Open correct destination	Keep version and audience context

[SCREENSHOT: versioned documentation page with version selector open]

Recording validation results across audiences and pages

As you test, record each result page by page rather than waiting until the end. Audience-targeted publishing issues are much easier to fix when you can point to the exact page, audience, and version where the mismatch appeared. A simple tracking sheet is enough as long as it captures what you tested and what failed.

For every page and audience combination, log:

- The page name
- The tested public URL
- The tested versioned URL
- The selected audience
- The selected version
- The browser session used, such as normal window or incognito
- A pass or fail result
- Notes describing what you saw

When a page fails, describe the exact visible issue in Atloria's published view. Good examples include:

- Breadcrumb shows the wrong page name
- Left navigation includes a page that should be hidden
- Audience-specific notice is missing
- Wrong version appears in the version selector
- Internal link opens the wrong destination
- Screenshot for another audience is visible

Use screenshots whenever the mismatch is visual or hard to describe in one sentence.

Page	Audience	Version	Result	Issue Found
Setup page	Audience A	v1	Pass	—
Setup page	Audience B	v1	Fail	Wrong callout displayed
Troubleshooting page	Audience A	v1	Fail	Broken related link

[SCREENSHOT: failed validation example showing incorrect callout or wrong breadcrumb]

Also note the scope of the issue:

- One page only
- All pages for one audience
- Multiple pages in one version
- All tested pages in the release

That scope helps your team decide whether the problem is a page-level publishing issue or a broader release problem.

Fixing common validation failures in targeted publishing

When audience-targeted validation fails, start with the visible symptom and trace it back to the publishing choices used for that page. In Atloria, most issues fall into a few repeat patterns.

If audience-specific content appears in the wrong public view, first reopen the page's audience targeting setup and confirm the page is assigned to the intended audience. Then compare the published result with your latest page changes. If the page still shows an older variant, the public output may not yet reflect the newest publish. Republish the page or release, then test again in a clean browser window.

If the versioned audience path opens the wrong page or the wrong release, check the selected entry in the version selector and confirm you are using the correct versioned URL for that document. A mismatch here often shows up as the right page title with the wrong body content, or the right audience with the wrong version label.

If hidden content is still visible after release, inspect the published page carefully and compare it with the intended audience view. Look for alternate screenshots, extra instructions, or links that should have been excluded. Then republish the corrected page and retest both the public page and the versioned page.

If links break when switching audiences or versions, test each link from the visible page again and confirm it points to the matching published destination. Audience-specific pages should link to the equivalent audience-aware destination whenever one exists. If a link sends readers to a generic page, the audience context may have been lost.

[SCREENSHOT: comparison of correct audience view and incorrect audience view]

After each fix, repeat the same validation steps on the affected page instead of assuming the issue is resolved across the whole release.

Overview

Audience-targeted publishing validation in Atloria is the check you perform after publishing to confirm that each audience sees the right documentation in the right place. You are not reviewing draft content here. You are reviewing the actual

published experience across the public documentation site and the versioned documentation view.

This validation focuses on a few visible outcomes:

- The correct page opens for the selected audience
- The correct release appears in the version selector
- Audience-specific sections display when they should
- Content for other audiences stays hidden
- Navigation, breadcrumbs, and links remain consistent

This guide assumes you already know how to publish documentation for a defined audience. If you need to revisit that setup, refer to [Publishing Documentation for Specific Audiences](#). The goal here is different: confirm that the published result matches your expectations before you move further into release checks.

In Atloria, this work usually happens after a version is published and before broader release sign-off. You will compare the same page in two places:

- The standard public documentation view
- The versioned audience path for the release

That comparison helps you catch issues that are easy to miss in editing screens, such as:

- A page showing the wrong audience-specific notice
- A hidden section leaking into the public page
- A version selector pointing to the wrong release
- A left navigation item appearing for the wrong audience
- An internal link dropping readers out of the intended audience path

[SCREENSHOT: public page and versioned page open side by side]

If you treat validation as a page-by-page check instead of a quick spot check, you will find release issues earlier and produce a more reliable published experience.

Prerequisites

Before you begin this validation workflow in Atloria, make sure the following items are already in place:

- A published documentation release that includes the pages you want to test
- At least one page with audience-targeted content already configured

- Access to the public documentation site for the project
- Access to the versioned documentation view, including the version selector
- A list of the audiences included in the release
- The expected public page locations and versioned page locations for each test page
- Notes describing what should appear for each audience, such as headings, notices, screenshots, or links

It also helps to prepare a small validation worksheet before you start. Include:

- Page name
- Audience name
- Expected version
- Expected navigation label
- Expected breadcrumb
- Expected visible content blocks
- Key links to test

Use a clean browser session for final checks. If you test in a browser window that already contains older page state, you may see outdated audience results and record a false failure.

If you are still setting up audiences or organizing audience rules across projects, review:

- [Defining Audiences for Targeted Documentation](#)
- [Managing Project Audiences for Targeted Documentation](#)
- [Using Public Navigation with Audience Specific Content](#)

If your release includes multiple versions, make sure you know which version is meant to be the public default and which versions should remain available in the version selector. The next step after completing this validation is [Validating Audience Targeted Publishing Before Release](#).