

Prioritize Documentation Updates

September 10, 2026

Prioritize Documentation Updates

Use change and freshness information to identify documentation that is most likely to need attention. This helps you focus maintenance work on updates that reflect recent product changes and keep User Guides reliable.

Overview

Documentation can become outdated when the underlying product changes, even if the documentation itself has not been edited. Atloria helps you review project changes and the project changelog so you can decide which documentation updates should be handled first.

Use this process when you are planning a documentation maintenance session, reviewing a new change, or preparing documentation for export. Start with the most recent or highest-impact changes, then compare them with the affected User Guides. This approach helps you avoid spending time updating documentation that is still current.

The **Change** area is useful for reviewing individual changes in detail. The **Changelog** provides a broader record of project updates. Together, these views help you establish which documentation areas deserve priority.



Screenshot pending: The project Change area showing available changes to review.

Prerequisites

Before you begin, make sure you have:

- Access to the relevant project in Atloria.
- Permission to view the project's **Change** area and **Changelog**.
- An understanding of which **User Guides** belong to the project.

- A maintenance goal, such as reviewing recent changes, preparing a release, or improving outdated documentation.
- Any available context from the team that made the product change.

Step-by-Step Instructions

1. Open the project that contains the documentation you want to maintain.
2. Select **Change** to open the project's change history.
3. Review the available changes and identify the newest changes first.
4. Open a change that appears likely to affect user-facing behavior, workflows, configuration, or screens.
5. Read the change details and note which documentation topics may be affected.
6. Select ← **Previous Change** when you need to compare the current change with the prior recorded change.
7. Select **Change branch** if you need to review changes associated with a different branch.
8. Open the project **Changelog** to review the broader sequence of documented updates.
9. Select **View Changelog** when it is available from the project documentation area.
10. Compare the changes in the **Changelog** with the topics covered by your **User Guides**.
11. Identify User Guides that describe behavior affected by a recent change.
12. Open the affected **User Guides** and review whether their instructions still match the current product behavior.
13. Prioritize guides that describe changed workflows before guides that cover unchanged areas.
14. Prioritize guides that have not been reviewed recently when they relate to a changed feature.
15. Select **Save Changes** after you update documentation settings or changelog-related information.
16. Select **Export Documentation** only after you have reviewed and updated the highest-priority documentation.

 *Screenshot pending: The Changelog view used to compare recent changes with documentation topics.*

Tips and Best Practices

- Start with changes that affect the user interface. A changed screen, button, workflow, or configuration option can make step-by-step instructions inaccurate quickly.
- Review changes in context. A single change may affect multiple documentation topics, such as a User Guide, an onboarding procedure, and a troubleshooting page.
- Use the **Changelog** for planning and the individual **Change** view for detail. The changelog helps you see what has changed over time, while a specific change helps you understand what needs to be updated.
- Treat documentation freshness as a prioritization signal, not the only decision factor. A recently edited guide may still need updates if the product changed after the guide was reviewed.
- Focus first on documentation that users depend on to complete common or high-impact tasks. For example, prioritize guides for core workflows before less frequently used reference material.
- Review documentation after changes move across product areas. Changes can involve frontend behavior, backend behavior, configuration, API-related behavior, documentation, or infrastructure. User-facing documentation is most likely to need updates when a change alters what users see or do.
- Keep change notes specific when possible. When reviewing a change, record the affected guide name and the section that needs review. This makes it easier to divide maintenance work across the team.
- Use **View Your Documentation** → to return to the documentation area after reviewing changes and deciding what to update.

Related Pages

- [Review Project Changes](#)
- [Use the Changelog](#)

- [Update User Guides](#)
- [Export Documentation](#)
- [Request Documentation Changes](#)
- [Configure Changelog Settings](#)

Troubleshooting

You cannot find the changes that should be reviewed

Select **Change branch** and confirm that you are viewing the branch associated with the work you need to document. Changes may differ between branches.

A change does not clearly identify the affected documentation

Review the related **Changelog** entries and compare the change with existing **User Guides**. Focus on guides that describe the same workflow, screen, or user action.

You are unsure whether a guide needs an update

Prioritize the guide for review if the change affects what users see, the order of a task, an available option, or an expected result. If the guide still matches the current behavior after review, no update is necessary.

Your updated documentation is not ready to share

Confirm that you selected **Save Changes** before using **Export Documentation**.

Review the changed guide one more time to ensure its instructions reflect the current workflow.