

Project versions

September 15, 2026

Project versions

Project versions at a glance

A documentation version is a complete snapshot of your documentation at a point in your codebase. Use version history to keep releases distinct, compare two snapshots to see what changed, and review regenerated or carried-forward content before publishing.

If you need to...	Go to
Check the versions belonging to a project	Find version history
Understand the counts and actions on a version row	Understand the version history screen
Inspect regenerated content before publishing	Review a version before publishing
See which documents changed between snapshots	Compare two versions
Change a version's status or handle approval	Manage version status and follow-up actions
Work with project changes	Project changes and audit
Review version screenshots	Review version screenshots
Understand the generation flow	Documentation generation lifecycle
Read the published result	Published documentation

Choose the task above, then prepare with **Before you start**.

Before you start

Prerequisites

- Open the project whose versions you need to inspect.
- Have at least two versions when you need to compare them.

- Check the version's status, the snapshot's lifecycle state, before choosing an action. The available status values are Draft, Active, Published, Deprecated, and Archived.
- Use an administrator role for **Approve** and **Reject**.
- Check whether a version is already scheduled before choosing **Schedule go-live....**
- Check whether flagged documents exist before publishing from the review surface.
- Check whether screenshot coverage is below 80% when the review surface presents a warning.

Once these conditions are clear, continue to [Find version history](#).

Find version history

Version history is the project view where you inspect saved documentation snapshots and choose whether to review, compare, or continue with a version action.

Steps

1. Open the project whose documentation snapshots you need.
2. Select **Versions** in the project workspace.
3. Select **Compare versions** to open the comparison surface when at least two versions are available.
4. Select **Back to versions** on the review or comparison surface to return to version history.

Result: The project's version history is on screen and ready for you to read the table.

Understand the version history screen

The version history screen is a list of documentation snapshots with their status, creation details, document counts, screenshot counts, category counts, and source information.

The table uses these columns:

Column	What it tells you
Version	Which snapshot the row represents
Status	The snapshot's current lifecycle state
Created	Who created the snapshot and when
Docs	The number of documents in the snapshot
Screenshots	The number of screenshots in the snapshot
Categories	The number of categories in the snapshot
Source	The commit hash or branch associated with the snapshot

The header includes **Generate new version**. Select it when you need a new snapshot from the project's current documentation source. The row actions depend on the version's status; use the status column before selecting one. Continue to the review or comparison procedure when the row requires closer inspection.

Review a version before publishing

A pre-publish review is the checkpoint for deciding how regenerated content and carried-forward content should enter the next published snapshot.

A flagged document is a document that needs a choice between keeping the previous edits, using the new AI version, or discarding the flagged content.

Prerequisites

- Open a version's review surface.
- Check the Needs review items before selecting a publishing action.
- Check the **Screenshots** summary. A warning appears when coverage is below 80% and at least one screenshot exists.

Steps

1. Select **View diff** on a flagged document when a regenerated version is available.

2. Select **Hide diff** to close the side-by-side comparison.

3. Choose one action for each flagged document:

If you need to...	Select
Keep the previous edited content	Keep edits
Replace it with the regenerated content	Use AI version
Remove the flagged content	Discard

4. Select **Keep all edits** or **Use all AI** when the same decision applies to every flagged document.

5. Choose one next action:

If you need to...	Select
Address the screenshot summary before publishing	Manage screenshots →
Leave the version in draft form	Save as draft
Ask a reviewer to approve the version	Request approval

6. Select the publish action labelled with the version number when there are no flagged documents to resolve.

Result: The review surface reflects your document decisions, screenshot status, and selected publishing path. Continue to [Compare two versions](#) when you need to inspect changes between snapshots.

Compare two versions

Comparison shows how the documents in two documentation snapshots differ, so you can inspect the change before choosing a publishing or follow-up action.

Prerequisites

- Have at least two versions available.

Steps

1. Open the comparison surface.

2. In **Select version A...**, choose the first version.
3. In **Select version B...**, choose the second version.
4. Select **Compare**.
5. Filter the results with **Filter documents by change status**:

If you need to see...	Option
Every result	All
Documents added in the selected change	Added
Documents removed in the selected change	Removed
Documents modified in the selected change	Modified
Documents unchanged in the selected change	Unchanged

6. Select **View** → on a document row to open its difference.
7. Select **Close** to leave the difference view.

Result: The comparison table is filtered to the change status you selected, or the selected document difference is open for inspection. Continue to [Manage version status and follow-up actions](#) when you are ready to act on the version.

Manage version status and follow-up actions

Version status tells you where a snapshot is in its publication lifecycle: Draft is newly generated, Active is the current working version, Published is publicly accessible, Deprecated is marked for removal, and Archived is read-only and hidden.

Prerequisites

- Open a version row and read its status.
- Use an administrator role for **Approve** and **Reject**.
- Keep a Draft or Published version unscheduled before selecting **Schedule go-live...**

Steps

1. Choose the action that matches the version's status or the work you need to complete:

If you need to...	Select
Activate a Draft or Published version	Activate
Schedule a Draft or Published version that has no scheduled activation	Schedule go-live... , then choose a future date and time and select Schedule
Leave the schedule dialog without scheduling	Cancel
Ask for approval on a Draft version	Request approval
Approve a pending version	Approve
Reject a pending version	Reject , enter a reason, then select Confirm reject
Publish a Draft or Active version	Publish
Mark a Published version for removal	Deprecate
Make an Active, Published, or Deprecated version read-only and hidden	Archive
Permanently remove a version that is not Active	Delete

Result: The version history reflects the lifecycle action you selected, or the schedule, approval, or deletion dialog remains available for the next decision. If a message appears, use

[When version work goes wrong.](#)

When version work goes wrong

Use the exact message on screen to choose the correction.

Message	When it appears	What to do
A reason is required to reject.	Reject is selected without a reason	Enter a reason before selecting Confirm reject .
Pick a valid date and time.	The scheduled value is not a valid date and time	Enter a valid value in the schedule field.
Scheduled time must be at least 2 minutes in the future.	The scheduled time is too soon	Choose a time at least two minutes ahead.

Message	When it appears	What to do
Mark this job as completed?	Mark complete is selected on a stuck generation job	Confirm only when the job is ready to be marked complete.
Failed to request approval	An approval request fails	Check the version state and try the approval action again.
Failed to approve	Approval fails	Check the approval state and try again.
Failed to reject	Rejection fails	Check the rejection reason and approval state, then try again.
Failed to schedule	Scheduling fails	Check the date and time, then try again.
Failed to cancel schedule	Cancelling a schedule fails	Reopen the version history and check whether the schedule remains.
Action failed	A version action fails without a more specific message	Check the version status before retrying.
Resolve failed	A flagged document decision fails	Reopen the review and check the flagged document.
Bulk resolve failed	A bulk flagged-document decision fails	Reopen the review and resolve the documents individually.
Publish failed	Publishing from review fails	Reopen the review and check whether flagged documents remain.
Review failed to load. Reload the page to try again.	Review data does not load	Reload the review page.
Approval requested. A reviewer will be notified.	Request approval succeeds	Wait for the reviewer to act before publishing.
Screenshot coverage is below 80%. Consider capturing missing screenshots before publishing.	Screenshot coverage is below the threshold	Select Manage screenshots → before publishing.
Compare failed	The comparison request fails	Check both selected versions and try Compare again.

Message	When it appears	What to do
Failed to generate changelog	Changelog generation fails	Return to the comparison and try the changelog action again.

When the correction is complete, continue to [What to do next](#).

What to do next

After you inspect a version, continue with the page that owns the next piece of work:

- Select **View changes** to continue with [Project changes and audit](#).
- Select **Manage screenshots** → to continue with [Review version screenshots](#).
- For generation and status flow, continue with [Documentation generation lifecycle](#).
- For the public result, continue with [Published documentation](#).
- To locate the project first, use [Projects overview](#).

The lifecycle descriptions distinguish the outcomes: Published is publicly accessible, Deprecated is marked for removal, and Archived is read-only and hidden.