

Today

A few things to look at today

Review 2 changes, fix 3 broken links, and check 1 failing test.

Review changes

Fix links

Recent changes

Version History

Restore across workspaces now needs admin approval.
Today, 9:12am

Runs

New filter: show failed runs only.
Yesterday, 4:51pm

Editor

Clarified autosave behavior.
Yesterday, 11:03am

Next release snapshot

Release 2026.1 (draft)

Critical requirements checked



82% (41 of 50)

View snapshot

Export PDF

Checks that need love

Version History

One end-to-end check is failing.
Last run: 18 min ago

Restore older version

No check yet for the decline flow.
Suggested: add a unit test

Permissions

Manual check is due.
Last checked: 2 weeks ago

Automation Designer > Version History

Product map

[Automation Designer](#)[Runs](#)[Version History](#)[Compare versions](#)[Restore older version](#)[Notes and labels](#)[Permissions](#)

Version History

[Requirement](#)[User-facing](#)[Critical](#)

Users can view, compare, and restore previous versions of an automation.

Quick status

Defined: ready

Implemented: yes

Checked: mostly (1 check failing)

[Open requirement](#)[Draft Jira stories](#)

Inside this requirement

- [Compare versions](#)
- [Restore older version](#)
- [Notes and labels](#)
- [Permissions](#)

Automation Designer > Version History

[Definition](#)[Versions](#)[Implementation](#)[Verification](#)[Delivery work](#)

What this does

Users can view, compare, and restore previous versions of an automation.

Why this exists

This helps people understand changes over time and recover safely from mistakes.

Not included within this requirement

- Branching or merging versions
- Restoring only part of an automation

How we know it works

- Users can see a list of versions with author and timestamp.
- Users can compare any two versions and clearly see what changed.
- Restoring creates a new version and does not overwrite history.
- Only admins can restore across workspaces.

At a glance

[Ready to build](#)[Implemented](#)[1 check failing](#)

Linked implementation
4 locations

Linked verification
7 checks

Quick actions

[Edit requirement](#)[Export PDF book](#)

Automation Designer > Version History

Definition

Versions

Implementation

Verification

Delivery work

Requirement versions

Compare or restore any version. Nothing is ever deleted.

Compare versions

v4 (current)

Added admin approval for cross-workspace restore

Jordan D. | Today 9:12am

Current

v3

Improved diff view for nested steps

Priya S. | May 18

View diff

Restore

v2

Added notes and labels per version

Jordan D. | Apr 02

View diff

Restore

v1

Initial timeline + restore

Alex K. | Jan 10

View diff

Restore

Review changes

What changed (v3 to v4)

- Added** Restoring across workspaces now needs admin approval.
- Added** A restore request can be declined, and the requester is notified.
- Clarified** Restoring always creates a new version, even if identical.

Why

Support asked for safer restores after a few accidental rollbacks between workspaces.

What to review

This is the short checklist for a safe update.

Implementation: restore_service.py
Add admin approval check

Implementation: RestoreDialog.tsx
Show approve/deny UI

Verification: e2e restore check
Fix failing approve/deny path

Verification: unit permissions test
Add decline edge case

Child requirement: Permissions
Confirm it still matches

Approve and make current

Create Jira follow-ups

Draft Jira stories

Proposed drafts

Push drafts to Jira

These are drafts. A PO can edit them before they hit the real backlog.

Story 1: Version timeline UI

Users can see versions with author, time, and notes.

Acceptance: list loads fast for 200 versions.

Edit

Story 2: Compare versions

Users can pick any two versions and see a clear diff.

Acceptance: show changes only toggle.

Edit

Story 3: Restore + audit

Restore creates a new version; log who restored and why.

Acceptance: restore is permission-checked.

Edit

Story 4: Admin approval

Cross-workspace restore requires admin approval.

Acceptance: requester is notified on approve/deny.

Edit

Implementation

Where it lives in code

[Rescan now](#) [Open mapping PR](#)

Links refresh after merges. If something moved, we will ask you to confirm it.

Status	Path	What it does	Last checked
Up to date	automation/versioning/history_service.py	Loads the timeline	Today
Up to date	automation/versioning/restore_service.py	Restores an older version	Today
Needs a look	ui/version_history/RestoreDialog.tsx	Restore UI and admin approval	After last merge
Up to date	api/routes/version_history.py	API endpoints	May 19

Verification

Checks and evidence

Add test link

This requirement is mostly checked. One end-to-end check is failing right now.

End-to-end	e2e/version_history_restore.spec.ts Failing: approve or deny path	18 min ago
Integration	api/test_version_history_routes.py Passing	18 min ago
Unit	unit/history_service.test.py Passing	18 min ago
Unit	unit/restore_permissions.test.ts Passing	Yesterday

Release 2026.1 snapshot

Release readiness

A snapshot is a point-in-time view of what the product meant and what checks passed.

Critical requirements checked



41 of 50 (82%)

What changed in this release

Version History	Admin approval for cross-workspace restore
Runs	Failed runs filter
Editor	Autosave clarification

Export release PDF

Compare to last release

Export

Requirements book

Choose what to include, then export a clean PDF you can share.

Include

- ☒ Tree (always)
- ☒ What this does and why this exists
- ☒ Not included within this requirement
- ☒ How we know it works
- ☐ Related delivery work
- ☐ Implementation links
- ☐ Verification links

Export PDF

Preview

Automation Designer

Version History

Users can view, compare, and restore previous versions of an automation.