RELEASE NOTES

Dawiso 2026.4 LTS

Long-Term Support Release

November 2026

Contents

Administration & Integrations **5**
Checkout Routing 5
Data Source Watching 6
Metamodel Admin 7
Package Installation 7
AI & LLM Support **8**
AI Chat 8
DawisoBot Preview Cards 9
Model Context Protocol 9
Tool Management 10
API Tables Improvements **11**
Filters 11
Manage Columns 12
Table Row Count 12
Fullscreen Tables 12
Automation **13**
Expressions 13
Triggers 13
Actions 13
Manage Automation Rules 14
Customizable Layouts Using Widgets **14**
Homepage & My Customizations 14
Dashboard Authoring 16
Object Detail Layout 16
Dashboards **18**
Page & Dashboard Permissions 18
Data Product Standard **19**
Data Market 19
Data Product Creation 20
OSI Export 21
Diagrams **22**
AI Diagram Creation & Editing 22
Drawing & Annotation 23
Entity Relation Diagrams 24
Knowledge Graph 24
Lineage 25
Autosave and Clearer AI Chat Status 25
Objects You Can't Access Stay Hidden 25
Rename and Move Published Objects Safely 26
Other Diagram Updates 26

External Link Checking....26
Import/Export Improvements....27
Export Reliability....27
Sample Data....28
Download Progress and Page Breaks....28
Restrict Bulk Export by Role....28
Split Exports and Alphabetical Templates....28
Stuck Exports Time Out Automatically....28
Ingestion....28
Interface Redesign....29
Login Page....30
Users & Permissions....31
Objects....31
Character Limit on Object and Title Names....32
Object Name Patterns....32
Change Object Type....32
Choose a Clone or Move Destination....33
Object Numbering....34
Hierarchy & Inheritance Reliability....34
Comment Moderation and Revision Filtering....35
Live-Updating Relations....35
Bulk-Remove Relations....35
Link Attachments....36
Required Attribute Before Embedding....36
Workflow Transitions....36
Workflow Transitions as Cards....37
Change an Application's Workflow....37
Impact Analysis Improvements....37
Onboarding & Guided Tours....38
First-Login Walkthrough....38
Guided Tours....38
Onboarding Checklist....38
Onboarding Hints....39
Packages....39
Attribute Validation....39
Components as Widgets....40
Metadata Sync....40
Package Editor....40
Build Custom Applications Without Package JSON....42
Packages Updates....42
Packages: Marketplace Improvements....43
Enabled by Default....43
Prerequisite Resolution....43

Marketplace Checks for Updates Right After Deployment....44
Reserved core Prefix....44
Packages: New data Type....44
Automatic Installation....44
Export Object Data....44
New Schema....45
Product Tours and Catalog Objects....45
Performance....46
Performance Optimizations....46
Large Rich Text Loads Safely....47
Operational Monitoring....47
Security & Permissions....47
Unified Platform....47
Gate....48
Identity Service....49
UX/UI Enhancements....49
Table Filters Are Easier to Read....49
Fullness Score Detailed Overview....49
Navigation....50
NPS Survey....50
Spaces Icons....50
Tooltips Improvement....50
User Relation Actions....51
WYSIWYG HTML Editor....51
Formatting Tools....51
Attachment Behavior....51
Diagram Embeds....52
Help Me Write....52
Fixes....53
Filters That Silently Returned Too Few Results....53
Permission Recalculation After Identity Service User Changes....53
Word and PDF Exports Keep Their Spacing....53
Smaller Fixes....53

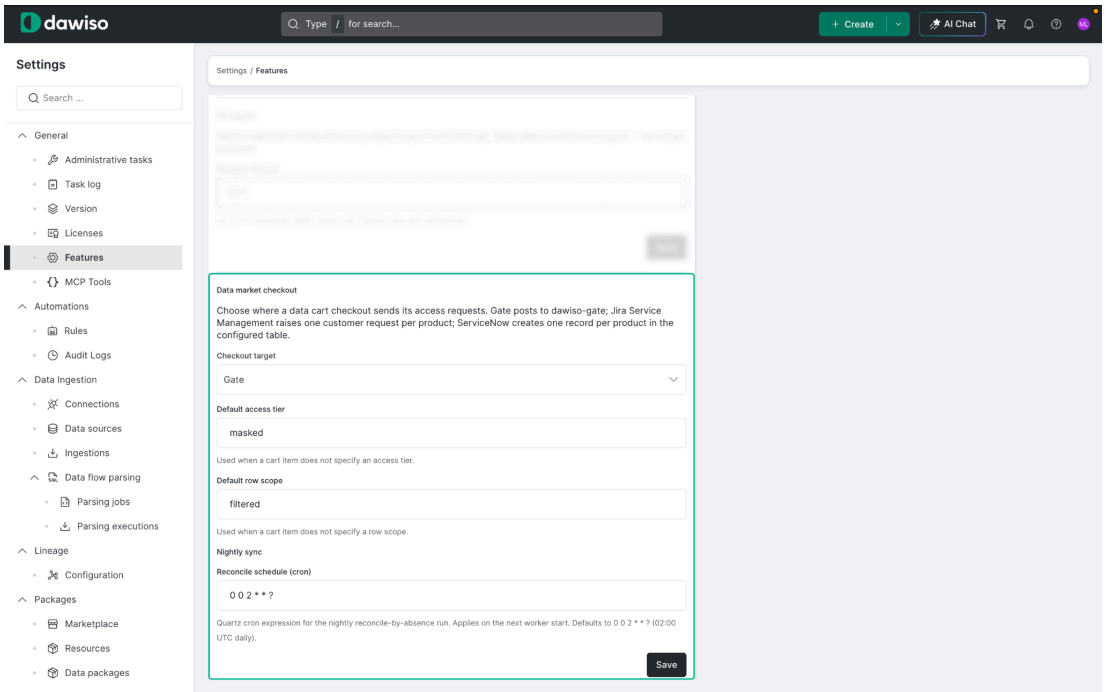
This Long-Term Support (LTS) release note for Dawiso 2026.4 consolidates every platform update since the 2026.2 LTS. The whole platform was redesigned and optimized for a cleaner, faster experience, with major new capabilities including Data Product Standard for governed data products and Unified Platform, connecting Dawiso to Gate through a shared identity service. Automation rules can now reach outside Dawiso to call external systems and move between environments as a bundle. Together, these updates also bring a more capable AI Chat, interactive API tables, customizable page layouts, more control over Word, PDF, and Excel exports, and a long list of reliability and performance fixes.

Administration & Integrations

This release adds a set of new administrative settings, covering checkout routing, data source monitoring, metamodel visibility, and package installation safety.

Checkout Routing

Requesting access to a data product goes through a cart-style checkout process. Cart checkout can route to Jira Service Management or ServiceNow instead of Dawiso Gate. In Jira Service Management, each product becomes its own customer request. In ServiceNow, each one becomes a record in a table you choose. Switching between Gate, Jira Service Management, and ServiceNow takes effect immediately, with no restart needed.

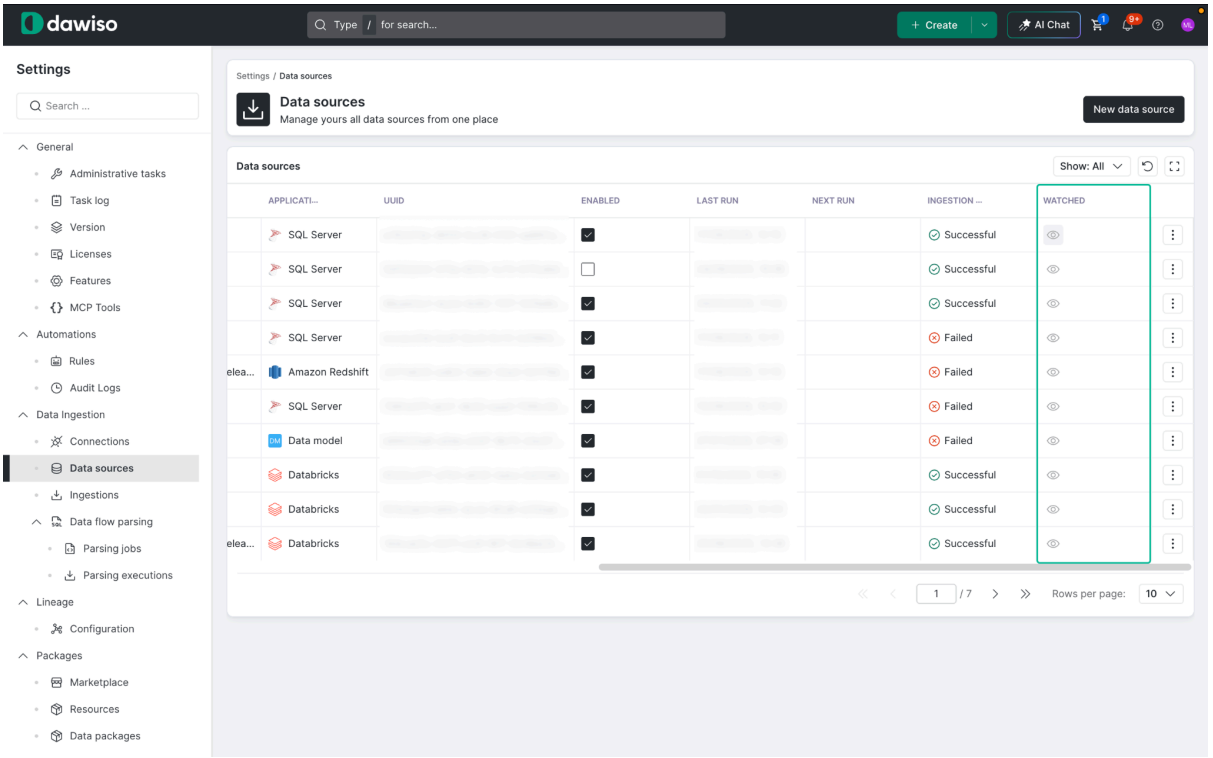


Checkout target picker set to Gate

Switching targets doesn't carry over requests already raised. Request status only flows back into Dawiso for Gate. For the other two targets, **My Access** shows an explanatory note in place of a status list. The nightly reconcile job behind this routing has no configurable retire safety ceiling. Only its schedule is configurable. See Data market checkout targets for the full configuration reference.

Data Source Watching

Administrators can watch a data source to be told when its ingestions fail, instead of checking the ingestion list for problems.



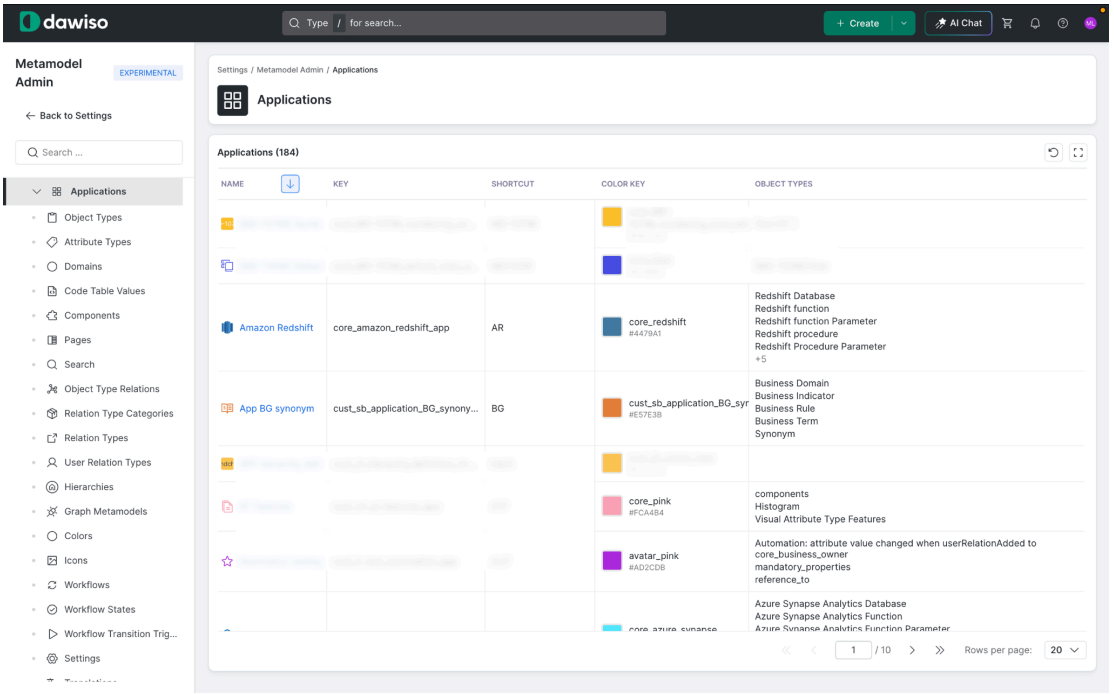
Data sources list with a Watched column of eye icons

A newly created data source is watched by its creator right away, while one that predates this feature stays unwatched until an admin opts in. A failed ingestion run notifies every watcher by email and in the notification bell's **Watching** tab, naming the data source, the failure state, and when the run started and finished.

Canceled runs don't notify, and a run stuck reporting the same failure repeatedly only notifies once. See Watching data sources for the full reference.

Metamodel Admin

Metamodel Admin adds a read-only, searchable inventory of an instance's whole metamodel: applications, types, attributes, relations, domains, and workflows, all in one place. Finding which object types reference a given attribute, for example, no longer means querying the database directly. It's a filter away in the new browser, available to administrators. See Metamodel Admin for the full reference.



Metamodel Admin's Applications screen

Package Installation

Package installs are both safer and more automatic:

- **Destructive-patch validation**: Installing a patch that would silently drop existing metamodel configuration fails validation instead of completing unnoticed. See Patch Packages: Advanced Guide for the technical detail.
- **Automatic metamodel propagation**: Installing a package propagates its metamodel changes automatically, removing the need for a separate manual template refresh.
- **REST endpoint for user sync**: User sync can also read organizational structure from a REST endpoint, alongside its existing input options.

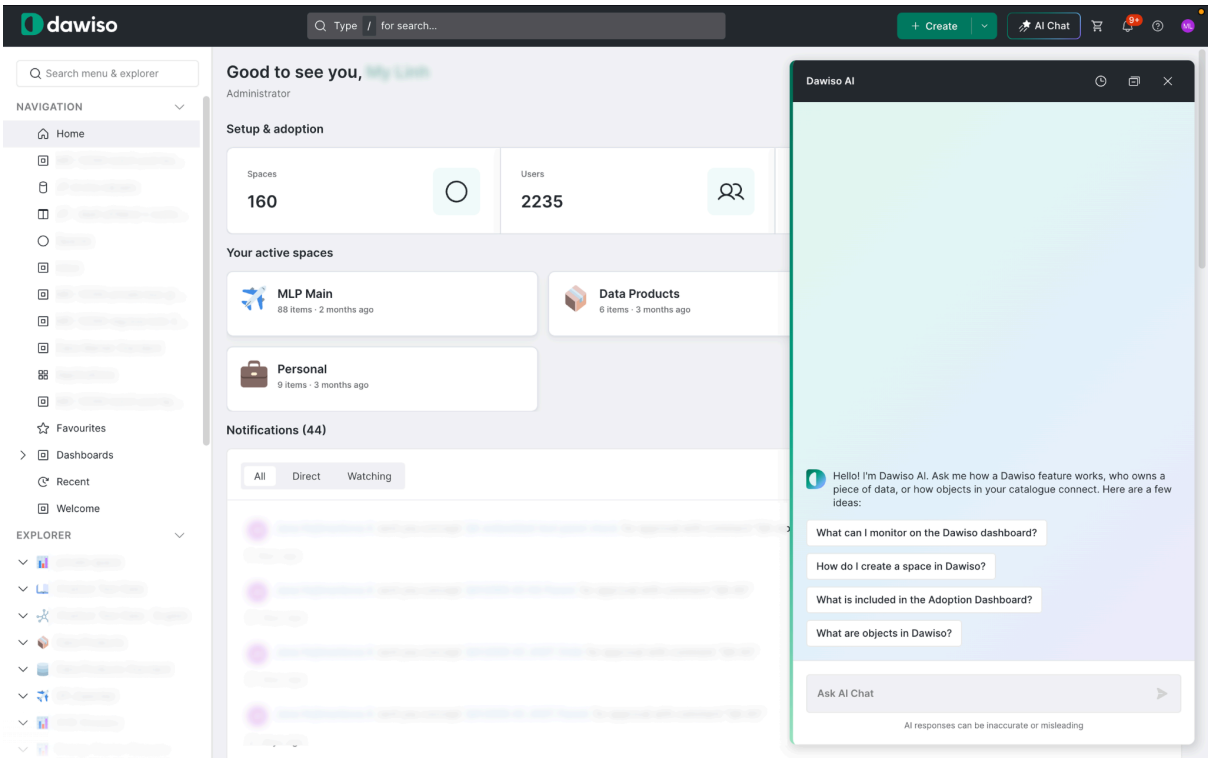
AI & LLM Support

AI and LLM support expanded across both releases. The **AI Chat** panel shipped in early access and gained page-context awareness, diagram editing, and connection-test troubleshooting, DawisoBot in Slack can show preview cards, and the Model Context Protocol tool surface was rebuilt into narrower, role-filtered tools.

AI Chat

AI Chat is available from the top navigation bar, where you talk to Dawiso AI about your catalog. This feature is in early access, and token usage is currently free. Continuing to use it once early access ends will require a license update.

Responses stream in as Markdown, with clickable citations linking straight back to the objects they're drawn from. Minimizing the panel or refreshing the page doesn't end the conversation, and each answer can be rated with a thumbs up or down. Every query is limited to spaces the asker already has read access to. An object outside that scope never surfaces in a response or its citations.



AI Chat panel open over the Dawiso dashboard, with suggested starter questions

The chat knows which object, space, or application page you're viewing, so phrases like "this table" or "this page" resolve to it directly, instead of asking you to clarify. When a connection test fails while you're setting up a connector, the **Explain with AI** button grounds its answer in the connector's own Dawiso setup guide. It names the field most likely at fault and links to the guide, instead of giving generic troubleshooting advice.

DawisoBot Preview Cards

Pasting a link to a Dawiso object in Slack turns into a preview card showing its name, path, description, type, space, state, and owner, built using the pasting user's own Dawiso permissions. See DawisoBot for Slack.

Model Context Protocol

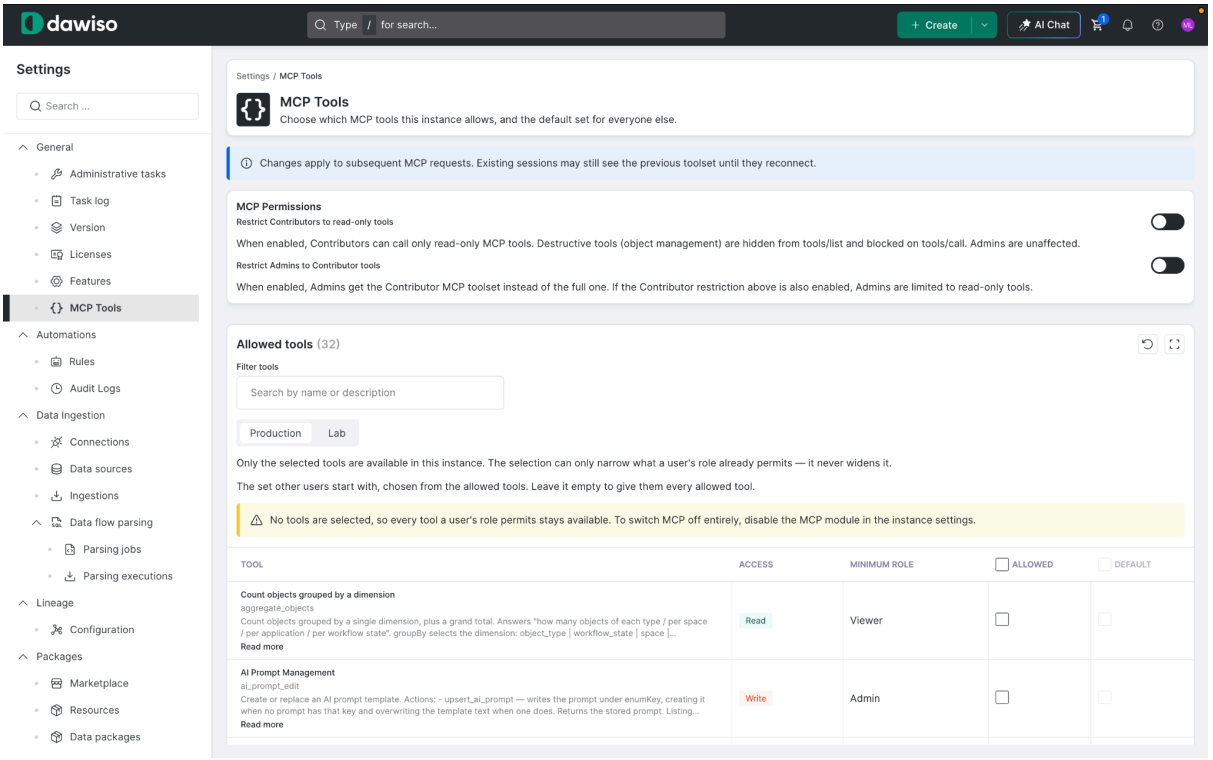
MCP's broad, multi-action tools were rebuilt into narrower, role-filtered ones, each handling exactly one kind of read or write. Calling `tools/list` from a **Viewer**, **Contributor**, or **Admin** session no longer returns the same catalog. Each role only sees the tools it's allowed to use. AI agents already integrated using the previous tool names won't pick up the change automatically. Each one has to reconnect so `tools/list` reflects the new set. Existing integrations should treat this as a breaking change. See Dawiso Model Context Protocol (MCP) for the full tool reference.

Additional improvements round out the MCP update:

- **Local user management**: An admin-only tool can add, update, remove, or reset the password for local Dawiso users, for environments using in-memory authentication rather than SSO.
- **Single-call attribute updates**: `update_object` writes several attributes in one call instead of one call per attribute.
- **Object creation without a concept**: `object_management` can create objects even for types that normally require an associated concept first.
- **Leaner responses**: Tool descriptions are shorter, with parameters documented in their schema, and every tool shares one response envelope. Results page at 20 items by default, capped at 100, with `mode` and `fields` parameters to control how much comes back. See Dawiso Model Context Protocol (MCP) for the full reference.

Tool Management

Administrators manage the environment-wide tool allowlist from **Settings > General > MCP Tools**, with separate **Production** and **Lab** tool sets and two switches restricting Contributors to read-only tools or Admins to Contributor tools. From **My AI tools**, each person can narrow their own available tools within whatever the administrator allows, with sortable, filterable tool tables for both production and lab tools. See MCP Tools Management for both settings pages in full.



MCP Tools settings page with the MCP Permissions switches and the Allowed tools table

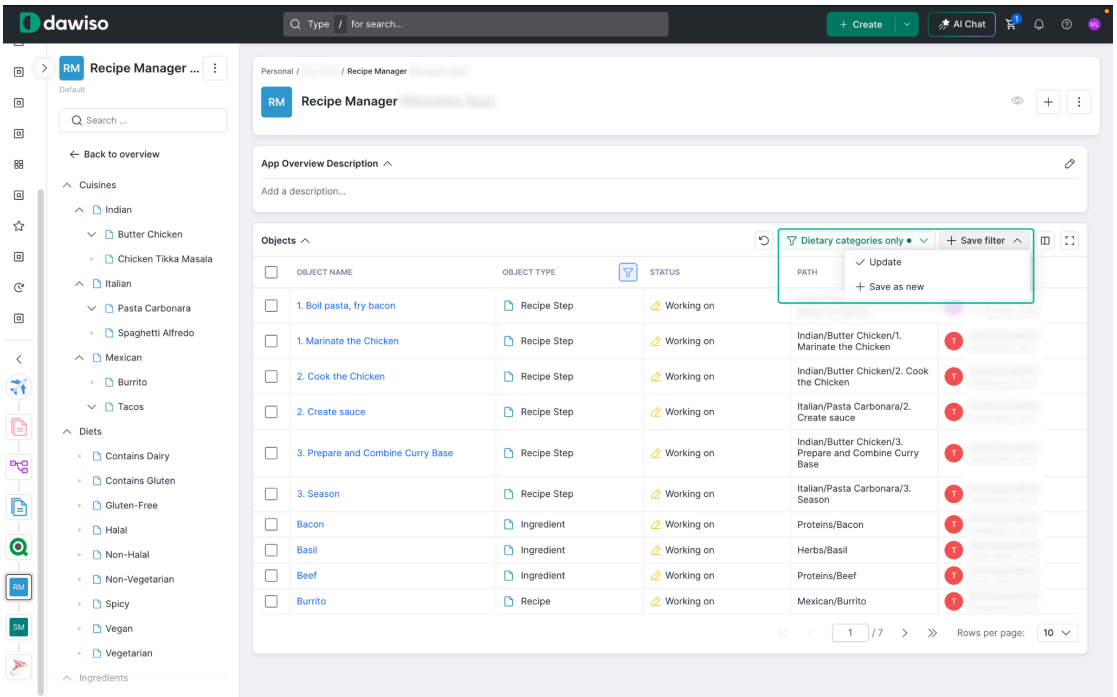
API Tables Improvements

Tables built with the API cards and API tables component gained more control across both releases: better filters, column management, row-count control, and a new fullscreen view.

Filters

Table filters gained three separate improvements:

- **Range filters on numbers**: Aggregated numeric columns, such as karma or view counts, can be filtered by a `from–to` range instead of an exact value. See API cards and API tables.
- **Saved filters**: A filter can be named, saved, and reused, and whichever one you used most recently loads again automatically when you reopen the table.
- **Searchable relation filters**: A relation column's filter is a searchable multiselect instead of free text. Open it to load the related objects, search by name, and scroll for more as they load in. Picking several objects in one column matches rows with any of them, and picking objects across two different relation columns narrows to rows matching both. **Select all** only grabs what's already loaded, so scrolling to the bottom first is still necessary to select everything.



A saved filter's dropdown, showing Update and Save as new options

Manage Columns

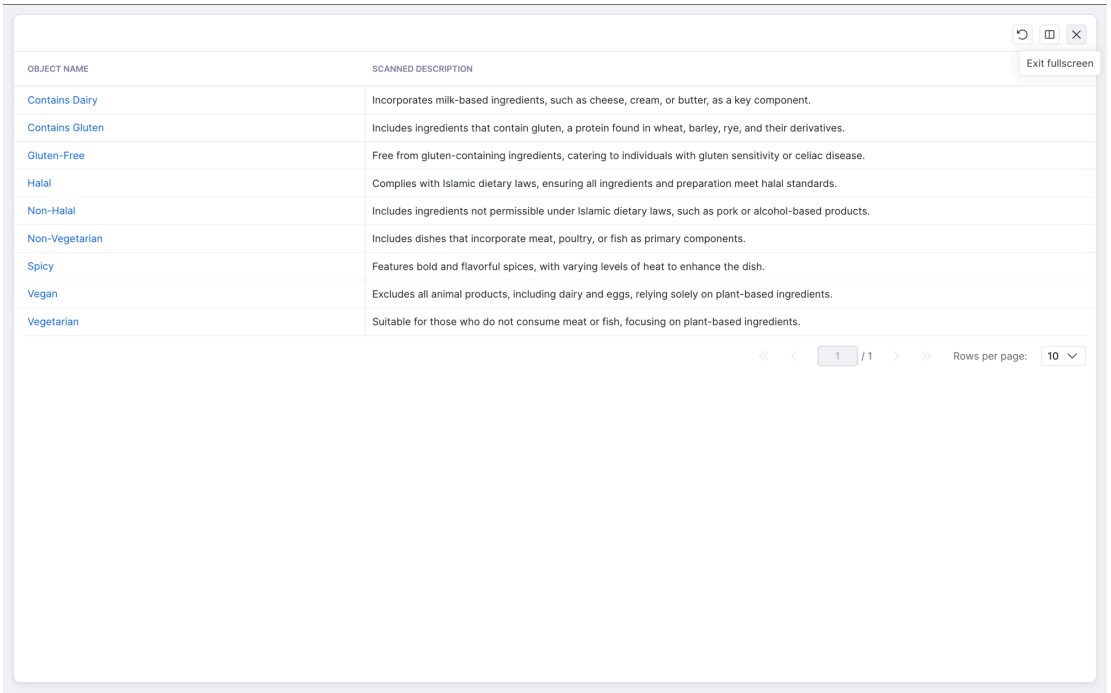
Any table listing Dawiso objects has a **Manage columns** control built into it, for hiding, reordering, searching, and showing columns without leaving the view. Metamodel attributes and user relations can be added as columns too, beyond the table's default set. **Restore default columns** reverts any customization back to how the table started. See API cards and API tables for the full reference.

Table Row Count

Any table built with the API cards and API tables component, dashboards included, can start out at a page size the table's author chose instead of a fixed default. Once a user picks their own rows-per-page value, that choice sticks and overrides the configured one from then on. See API cards and API tables for the full reference.

Fullscreen Tables

Every table across Dawiso, such as API tables, relations tables, and sample data tables, can expand to fill the whole page with a **Fullscreen** button in its toolbar. The table keeps its page, sorting, filters, and column widths when you switch in and out.



A table expanded to fill the page, with the Exit fullscreen button in the top-right corner

Automation

Automation rules now allow for more flexibility thanks to triggers catching more indirect changes, dynamic expressions gaining a rich-text formatting helper, and new actions calling out to external systems. Automation rules can now also move between environments as a bundle.

Expressions

The `.stripHtml()` function is available in dynamic expressions. It strips HTML tags from a value and collapses it into single-line plain text. Use it when passing a rich text attribute into `append()` or `appendLine()`. Without it, the attribute's `<p>` tags break the line. See Functions for the full list of expression functions.

Triggers

The `userRelationAdded` and `userRelationDeleted` triggers also fire on role changes at the space level, on inheritance to descendants, and on workflow transitions. Previously, only a direct edit fired them.

- `userSyncFinished`: Reacts the moment a user synchronization finishes, so a rule can act on exactly the users that changed. The task log for identity-service synchronizations also shows who was added, changed, or removed in each run.

Actions

- `changeRelation`: Connects or disconnects objects without a manual edit.
- `setUserRelation`: Assigns or reassigns people to roles like steward or owner.
- `renameObject`: Renames an object from an expression, for example combining its parent's name with one of its own attributes.
- `callRestApi`: Reaches out to an external system, for example to open a ticket in an ITSM tool or register a newly published data product in an outside portal. A REST API connection keeps the address and credentials it needs in one secure place, so a rule never needs a password written into it.

Manage Automation Rules

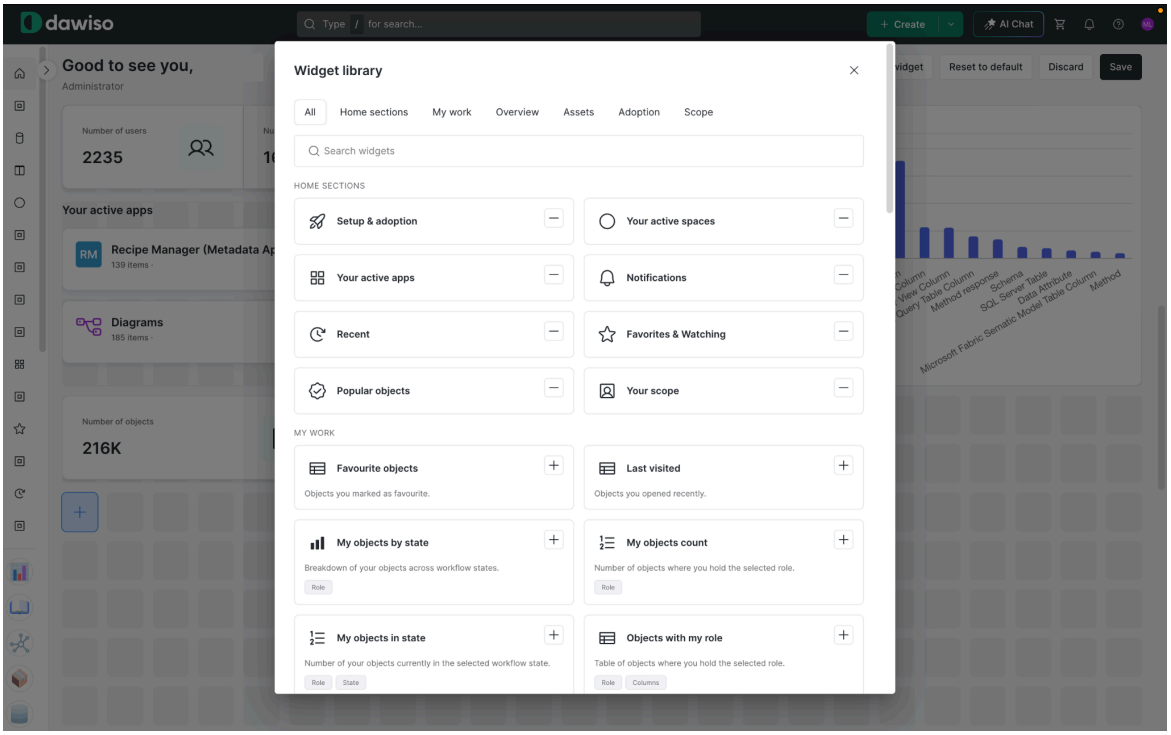
Automation rules can move between environments as a bundle. Export them from one instance and import them into another, for example from a test environment into production, without rebuilding every rule by hand.

Customizable Layouts Using Widgets

Dawiso's homepage, dashboards, and object detail pages all let you rearrange their content directly, instead of relying on a fixed layout. The homepage and dashboards use a free-canvas grid of widgets that can be added, moved, and resized. Object detail pages use a similar drag-and-drop approach within a fixed two-column structure. A dashboard's widget catalog, for example, includes a **Broken links** widget tracking external links that no longer resolve, covered under External Link Checking.

Homepage & My Customizations

The left navigation menu works the same way in Customize mode, where items can be reordered or nested under one another by dragging a handle. Changes on either surface stay as an unsaved draft until you **Save**, with **Discard** and **Reset** available to back out.



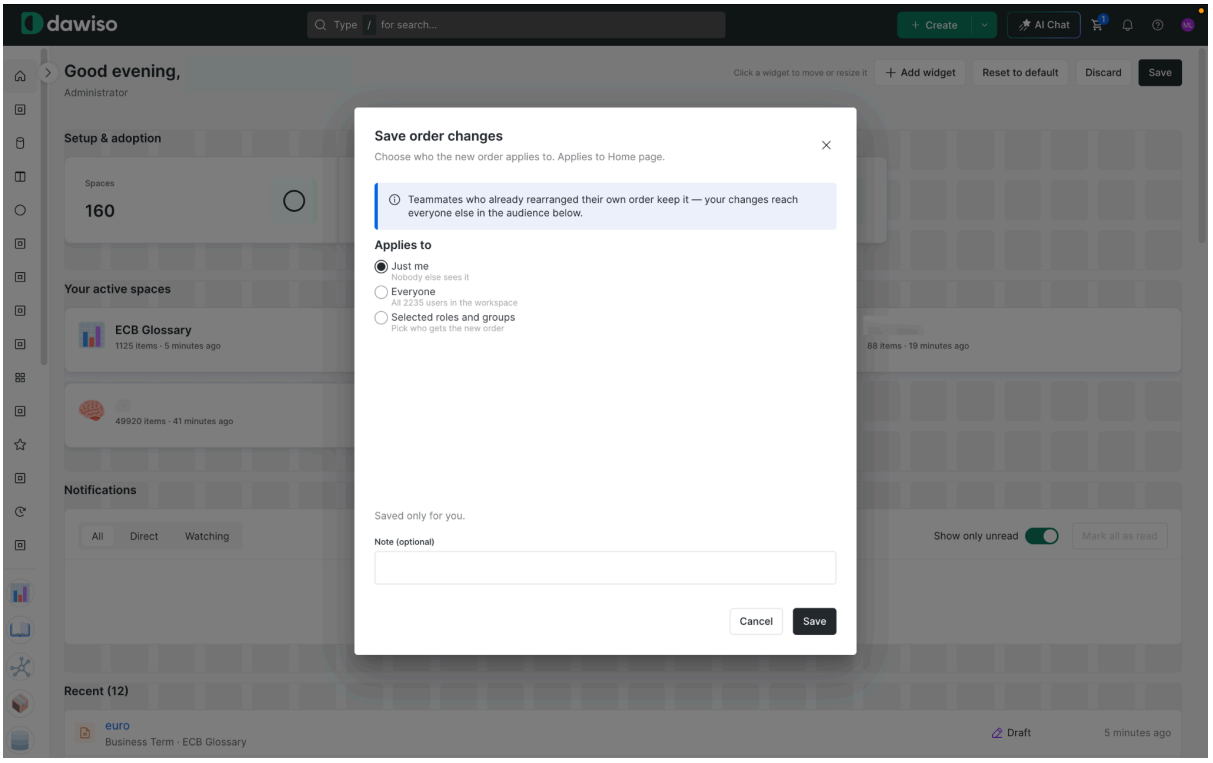
The widget catalog opened from a homepage in Customize mode

See Homepage layout and widgets and Customize navigation for the full reference.

Administrators can publish a layout to everyone, or to selected license roles and groups:

- A homepage, navigation, or Space layout can be published this way.
- Each publication is versioned and can be previewed before anyone switches to it.
- Every user's **My customizations** screen lists what's available to them on each surface: their own arrangement, any published layout they're entitled to, or **Automatic**.
- A **Used by you** badge marks whichever one currently applies.
- A layout a user picks for themselves stays in effect even after a higher-priority layout is published, until they switch back to **Automatic**.

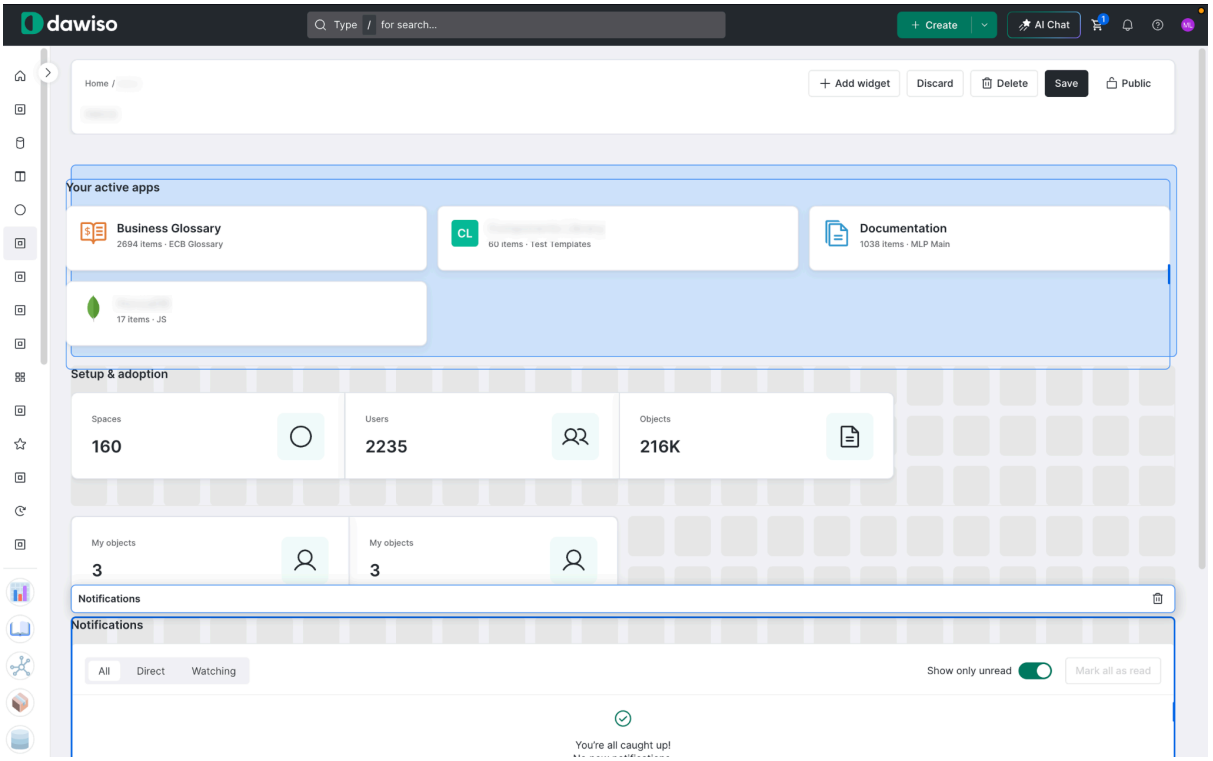
Copying and pasting a widget on the homepage lands it wherever the pointer sits: on an empty cell, on top of another widget (pushing it and everything below down), or below the copied widget when pasted without pointing anywhere in particular. See My customizations and Published customizations for details.



Save order changes dialog with Applies to options: Just me, Everyone, and Selected roles and groups

Dashboard Authoring

Administrators can create a named dashboard from **Customize** mode. Saving it creates an installable package behind the scenes, versioned on every edit, so the dashboard can move between environments like any other package. The dashboard becomes a page in the left navigation. Creating, editing, or deleting it still requires administrator permissions, and shows clear progress while Dawiso finishes the job in the background, with a **Refresh page** option if it takes too long. Viewing it follows the dashboard's own page permissions. Custom dashboards always open at a `/dashboard/` address. Older `/page/` links redirect automatically. See Dashboards for the full walkthrough.



A dashboard open in Customize mode, with Add widget, Discard, Delete, and Save controls

Object Detail Layout

Instead of hand-editing package files, administrators can drag, add, and remove an object type's detail-page components right on a live object, with every change reflected immediately. Access it from the object's **three dots** menu via **Edit template components**. The result saves as the new layout for every object of that type.

- Components can be moved, duplicated, renamed, configured, or removed, and dragged anywhere within the two-column layout or into a panel.
- Rows within attribute lists, relation lists, and related-user lists are addressable individually. Each gets its own toolbar, moves independently of its siblings, and multiple rows can be selected and moved together as a range, or promoted into a standalone component.
- Tables can be configured to list either child objects or related objects, with column and relation-type narrowing.
- Relation and role rows can get a per-language display name that overrides their default label.
- New attributes and user roles can be created directly from the editor's dialogs instead of requiring a package edit.

A few rules keep the layout coherent:

- Panels and diagrams only sit at the top level of a column. A relations table is the one exception that can also go inside a panel.
- Adding a component at a column's top level wraps it in a panel that needs its own name, except for a standalone rich-text attribute or diagram, since those already draw their own frame.
- Components shared from a package, such as the **Ownership** panel, detach into the object type's own copy automatically the moment they're edited at the row level, so later package updates no longer reach it.

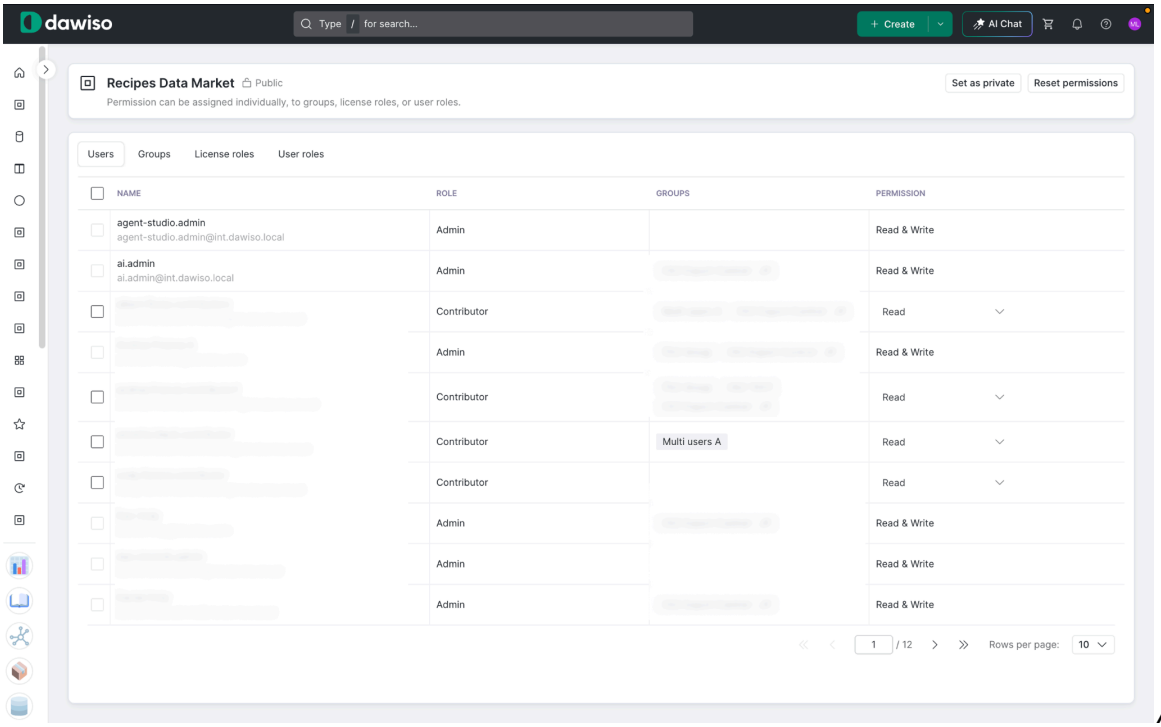
See Object detail layout for the full reference.

Dashboards

Dashboard pages gained finer-grained permissions, and creating or deleting a dashboard is clearer about its own progress. For rearranging dashboard content itself, see Customizable Layouts Using Widgets.

Page & Dashboard Permissions

Each dashboard page can be set to **Public** or **Private**. A **Private** page stays invisible to everyone, administrators included, until access is explicitly granted. Regardless of that setting, access can be granted per user or group at **Read** or **Read & Write** level. See Dashboards for the full permissions reference.



A dashboard's permission screen listing users with their role and Read or Read & Write permission

Data Product Standard

Data Product Standard is a new application type that packages data into governed, ready-to-consume products, instead of free-form catalog entries.

Every product is assembled from the same typed pieces:

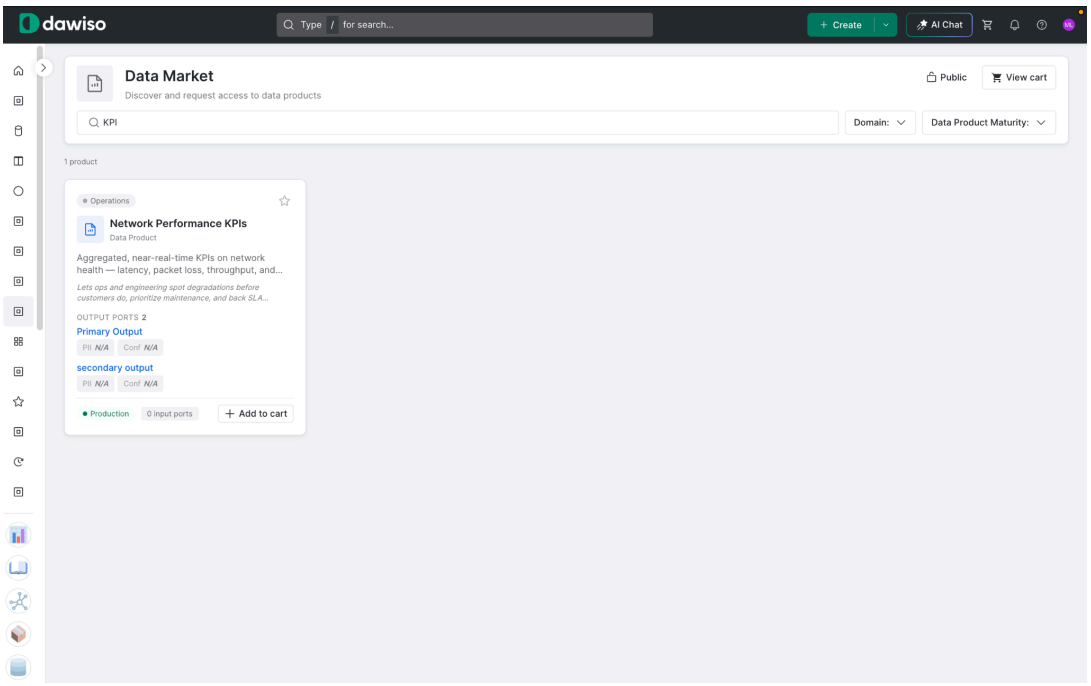
- Ports for exposing or consuming data
- A contract covering what's delivered and its schema
- Quality and service-level guarantees
- An accountable owner

The whole model maps to the Open Data Product Specification, the Open Data Contract Standard, and Data Mesh port conventions, so a product described here exports cleanly to those formats too. Products also connect to each other through their ports, and Dawiso draws each producer-to-consumer link as lineage, surfacing cross-team dependencies that would otherwise stay hidden.

Data Market

Once published, a data product surfaces in the Data Market for anyone in the organization to explore and formally request, instead of tracking down an owner informally. Each product also carries a visible classification for personal-data content and who's allowed to see it, so sensitivity is clear before anyone asks for access.

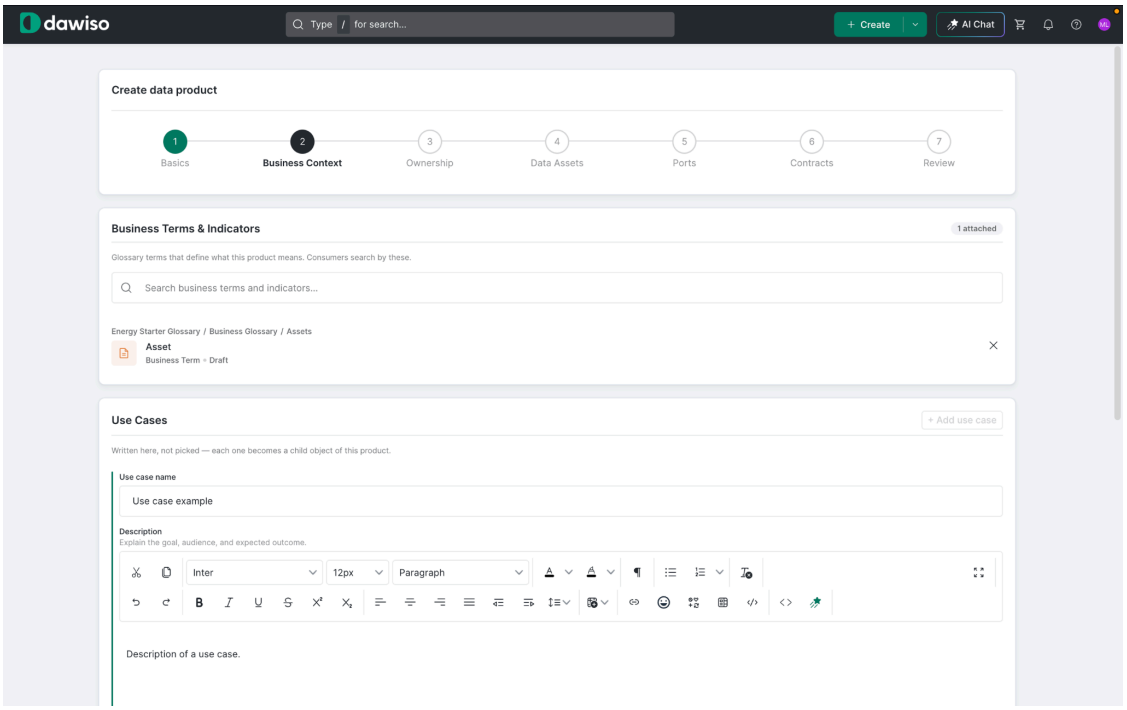
After requesting access, track its status from **My Access**, opened from the profile menu in the top right.



Data Market search results for a data product, showing its output ports

Data Product Creation

Building a product is guided rather than freeform, through a series of steps, including identity, business context (glossary terms, use cases, related products, and labels),ownership, underlying data, and contract terms.



The Business Context step, with Business Terms & Indicators and Use Cases sections

An AI agent can also build a starting draft of an entire product from an object already in your catalog, prefilling identity, ownership, assets, ports, and a contract automatically. Reviewing, adjusting, and publishing that draft is still up to you, but starting from something prefilled removes most of the manual setup. Opening a prefilled draft this way replaces any product you already had in progress, without the usual prompt to resume or discard it, so finish or publish other in-progress work first.

Returning to an unfinished draft blocks progress with a dialog to **Resume draft** or **Start over**, instead of a dismissible banner.

OSI Export

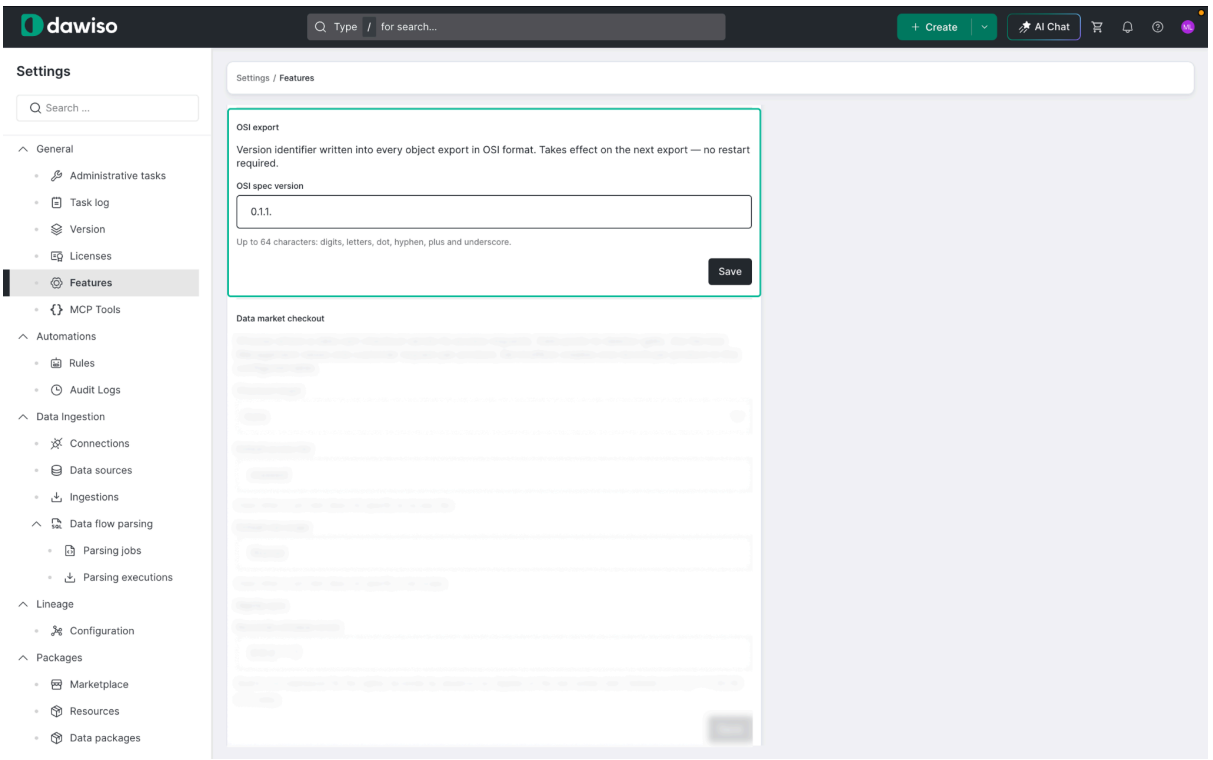
A data product can be exported as **OSI YAML**, following the Open Semantic Interchange standard for tables, columns, and foreign-key relationships. The option sits next to **Export** wherever a data product is reachable:

- Its own detail page
- A multi-select in an app's or space's Objects table
- The **three dots** menu on an application or space

An output port with no attached tables is left out of the file, and an object that isn't a data product, or whose ports were all skipped, is skipped entirely. What got left out is recorded in the export task's log rather than the notification.

The downloaded file works with any OSI-compatible tool, Snowflake included. For Snowflake, Dawiso can also generate a ready-to-run SQL file that loads the export directly. If Dawiso can't safely resolve a single target database and schema, or a data source isn't a plain table reference, it comments the statement out and explains why instead of guessing at a target.

Administrators can also change the specification version written into every export, which defaults to `0.1.1`. See Export a data product as OSI YAML for the full reference.



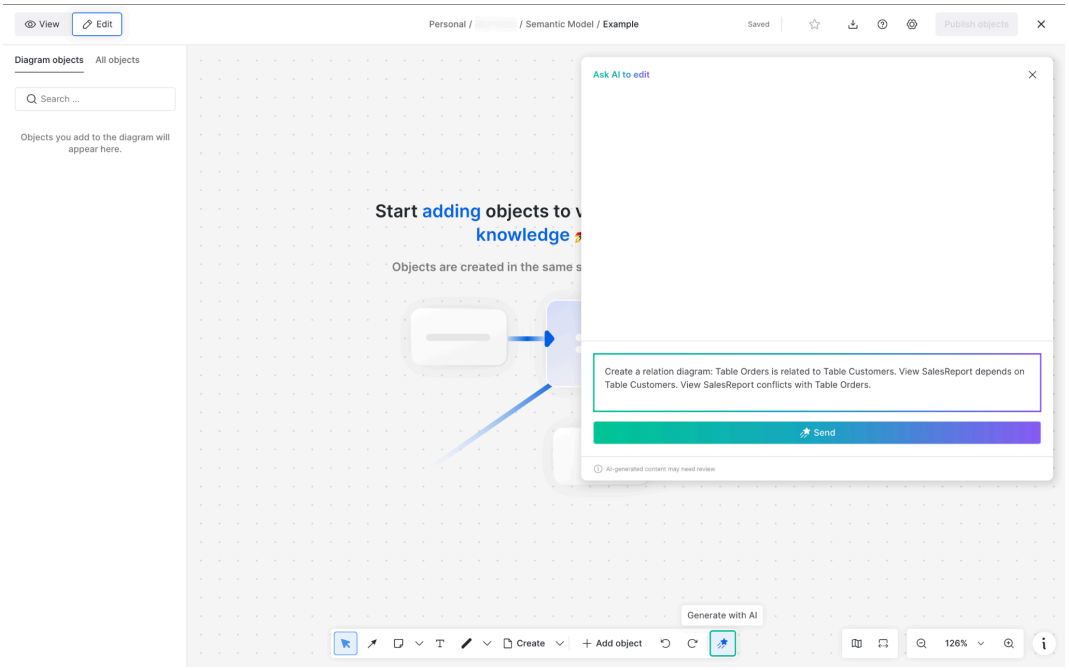
OSI export settings showing the OSI spec version field

Diagrams

Diagrams gained a more capable AI assistant, freehand drawing and annotation tools, a lineage layout built on ELK (Eclipse Layout Kernel, an open-source graph-layout engine) for large, complex flows, autosave, and several reliability fixes.

AI Diagram Creation & Editing

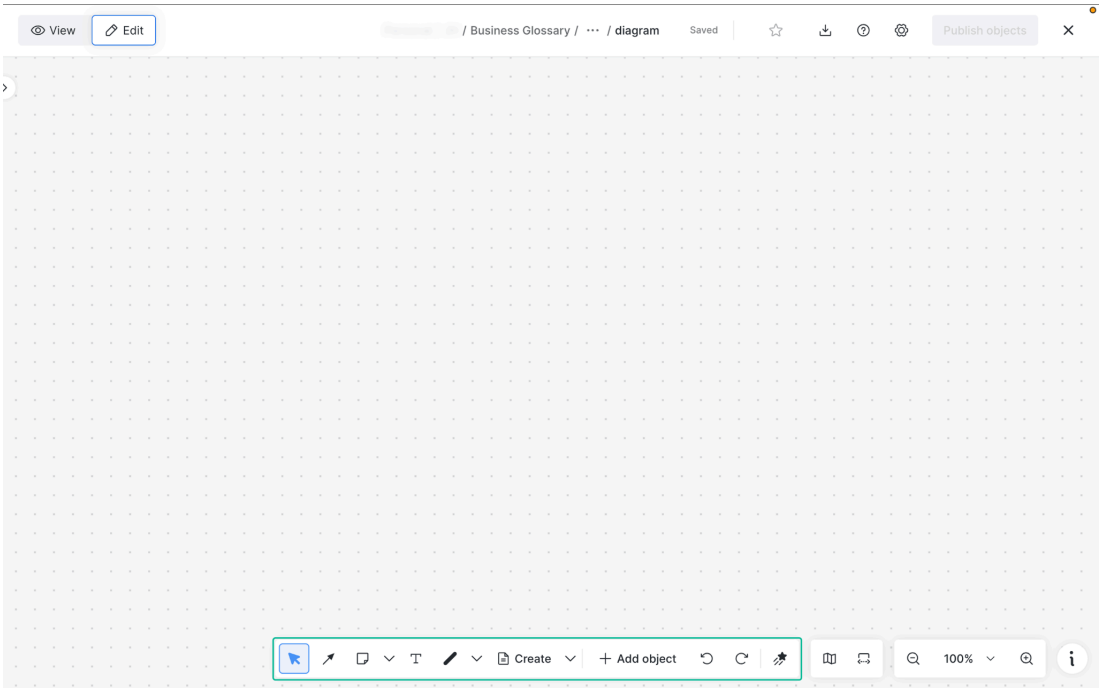
AI-generated diagrams also edit an existing diagram instead of only creating a new one from a description. **Ask AI to edit** applies a plain-language instruction directly to the canvas (create, merge, or change objects and relations), the same way you could do it by hand. A diagram no longer has to be regenerated from scratch to reflect a change. Describing what's different is enough, and the rest of the diagram stays as you left it. This works for conceptual, semantic, and logical model diagrams alike.



The Ask AI to edit panel open over a semantic model diagram, with a prompt describing the relations to create

Drawing & Annotation

Diagram drawing and annotation (notes, shapes, lines, and freehand elements placed directly on the canvas) got smoother and more stable performance. See Objects and elements and Actions in diagrams for the full reference.



A diagram canvas in Edit mode with the bottom drawing toolbar open

Entity Relation Diagrams

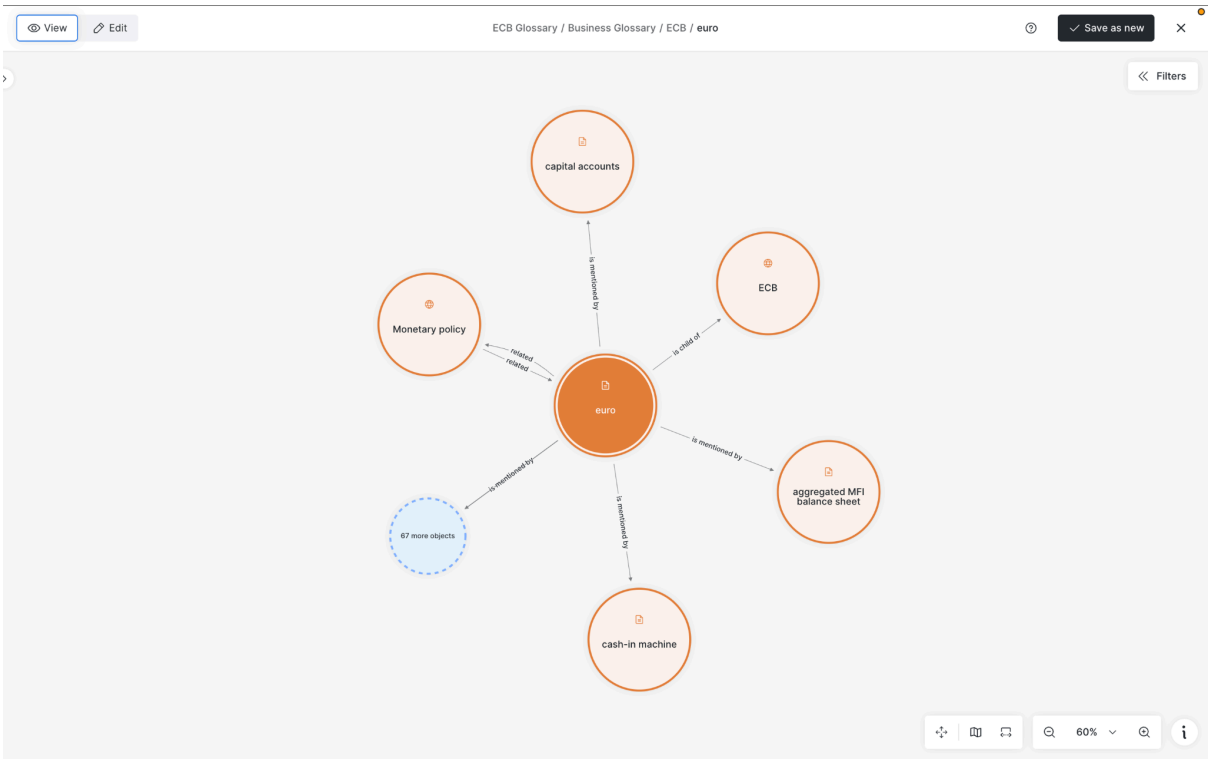
Expanded objects in entity relation diagrams space themselves out automatically. A schema with many tables and columns stays legible as it grows, without manual rearranging.

Knowledge Graph

Knowledge graphs no longer draw every relation of a densely connected object at once. When an object has many relations of the same type, the extra ones collapse into a single grouped node labeled with a count:

- Double-clicking the group opens a paginated, filterable list.
- Selecting objects from it adds them to the canvas individually.
- Once every object in a group has been added, the grouped node disappears.
- Grouping applies at every level of an expansion, not only the diagram's starting object, so even a wide expansion of a heavily interconnected term stays readable.

See Knowledge graph for the full reference.



A knowledge graph with a dashed grouped node labeled "67 more objects"

Lineage

Lineage diagrams gained several improvements:

- **ELK-based layout**: The layout engine switched to ELK. Large diagrams, on the order of a hundred flows, stay readable instead of collapsing into a tangle, and the layout restores the left-to-right reading direction lineage had before.
- **Power BI column-level tracing**: Relations are parsed from Power Query M expressions (the formula language behind Power BI's data transformation steps) down to the originating source systems, not just between Power BI objects themselves, and a parsing failure is written to the run log instead of being silently dropped.
- **Custom object-type colors**: An object type can be assigned its own color, carried into every lineage diagram so a given type is recognizable at a glance across a large flow.
- **Deterministic recalculation**: Lineage recalculates correctly after a package changes the hierarchy its graph is built on, and objects no longer disappear silently after a hierarchy change.
- **Parsing options per application**: Lineage parsing options can be overridden per application, instead of only globally.

Autosave and Clearer AI Chat Status

Modeling diagrams autosave your work, and retry automatically if a save fails. Leaving with unsaved changes asks for confirmation first. The diagram AI chat's status also distinguishes a request that's still working, one that's stopped, one that failed, and one that's timed out after three minutes, instead of lumping the last three together.

Objects You Can't Access Stay Hidden

Diagrams, the knowledge graph, data lineage's Excel export, and the MCP lineage tool all show an object you can't read as **Object without access** instead of leaving it out silently or exposing its name.

Rename and Move Published Objects Safely

Renaming a published object directly on a diagram renames it everywhere, after you confirm the change. Undo and Redo also rename it back and forward. Moving a published object out of a folder correctly reports moving it to the space root, and both folder moves and renames can be undone from the toolbar or with `Ctrl+Z` or `Cmd+Z`.

Folders in conceptual models can nest to any depth, dragged in from the canvas or the palette, or created by resizing a folder over other elements.

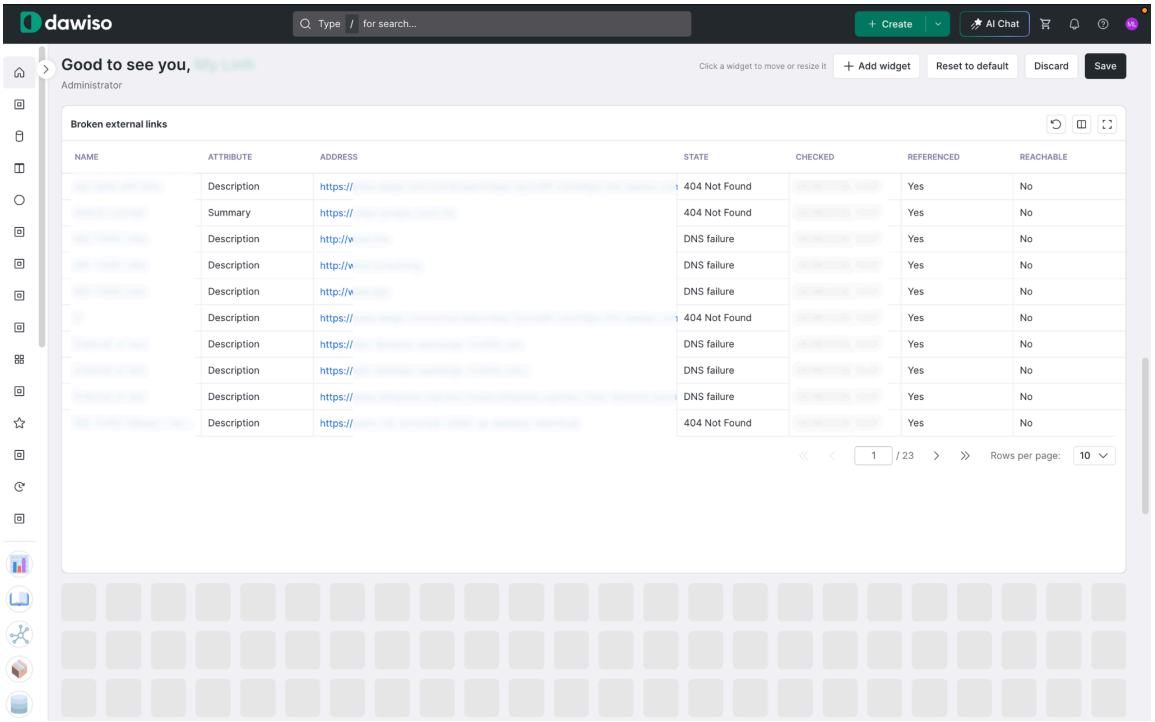
Other Diagram Updates

If an object's configured parent type isn't present in the diagram, graph metamodels fall back to the nearest ancestor that has a node there, instead of the object disappearing from the graph. Misleading origin markers on parent-level edges have been removed, and the object detail page's preview panel no longer renders a diagram by mistake.

External Link Checking

This feature checks the external links inside your object texts and reports which addresses no longer resolve. A standalone checker performs the checks from your own infrastructure, not from Dawiso itself.

Results appear on a **Broken links** widget that you can add to any dashboard. By default it lists addresses that are currently broken and still referenced somewhere, most recently checked first. Each row carries the object and attribute the link sits in, the address and its result, the check date, and both its referenced and reachable status. See External link checker for setup and the full field reference.



Broken external links widget listing addresses with their state, checked date, referenced, and reachable columns

Import/Export Improvements

Objects can carry a small data sample directly, several export paths that previously failed silently got fixed, and bulk exports are more flexible and easier to follow while they run.

Export Reliability

An automatic export that produces an attachment fails the task outright when the result is empty, rather than reporting success on a zero-byte file as it used to. Any attachment already created that way needs to be regenerated. It won't fix itself.

Additional fixes further improve export reliability:

- Word and ZIP exports reliably open in Windows Explorer.
- Generating an Excel Template while looking at a historical version of an object captures that same version, instead of quietly pulling the current one.
- The **Changes** tab shows word-level differences again, restored after a prior regression.

Sample Data

Any object you can edit can carry a small CSV or JSON data sample, viewable as a sortable, filterable table wherever the object itself is shown. Upload it through the UI, the REST API, or an MCP tool that an AI assistant can also read from when answering. A new upload replaces the old sample outright, and nothing preserves what was there before. Sample values stay out of search results, appearing only in that table. See Sample data for the full reference.

Download Progress and Page Breaks

Downloads start in place from the finished-export notification, which also shows your place in the queue while you wait. Page breaks you insert in a rich-text attribute carry through to the Word export in the same spot.

Restrict Bulk Export by Role

Administrators can turn off bulk export for Viewers and Contributors where it isn't needed.

Split Exports and Alphabetical Templates

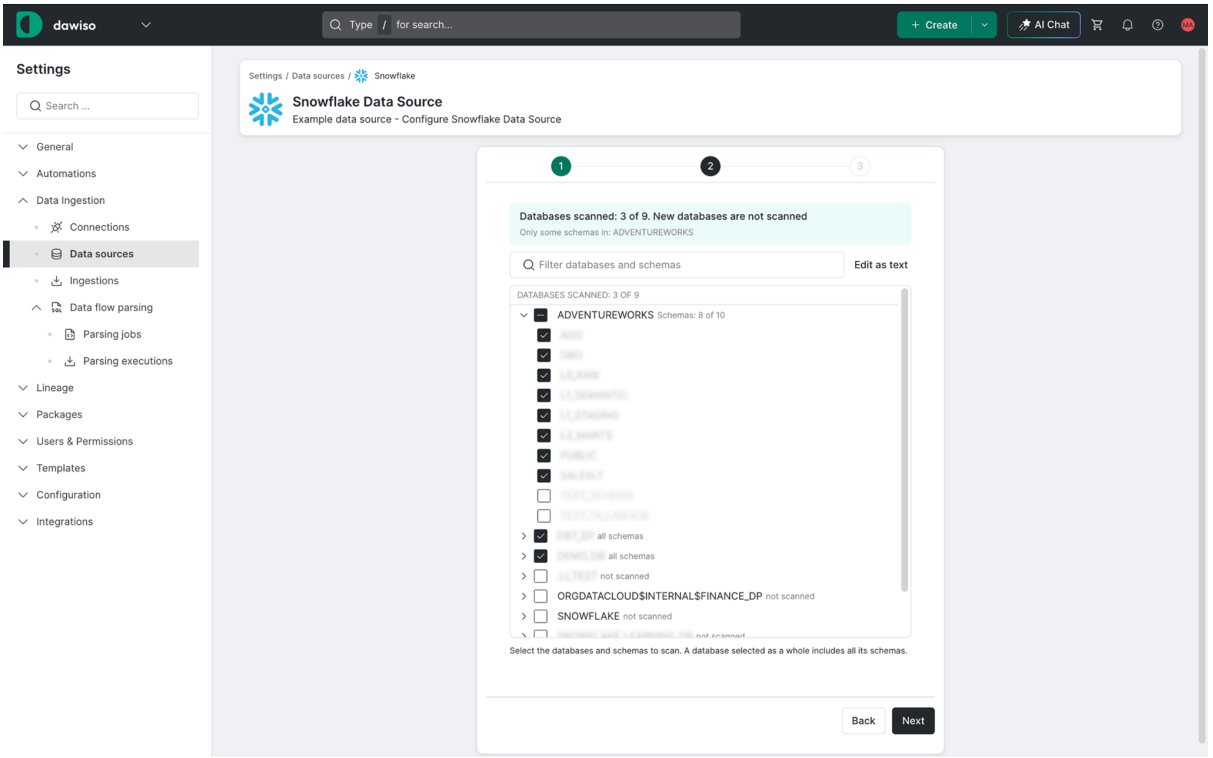
A large export can split into several documents instead of one, grouped either by top-level object or by what you selected, and templates can list every object alphabetically instead of following the hierarchy.

Stuck Exports Time Out Automatically

A stuck export stops itself after 30 minutes if it doesn't finish, instead of blocking every other export on the instance indefinitely. An admin can also cancel a running export directly from the Task log.

Ingestion

Catalogs and schemas for Databricks and Snowflake load live from your connected workspace, so you can select exactly what Dawiso should scan from the current list instead of typing it out. A separate issue that could silently drop notebooks and SQL queries from the Databricks catalog is also fixed, so they're tracked the way they always should have been.

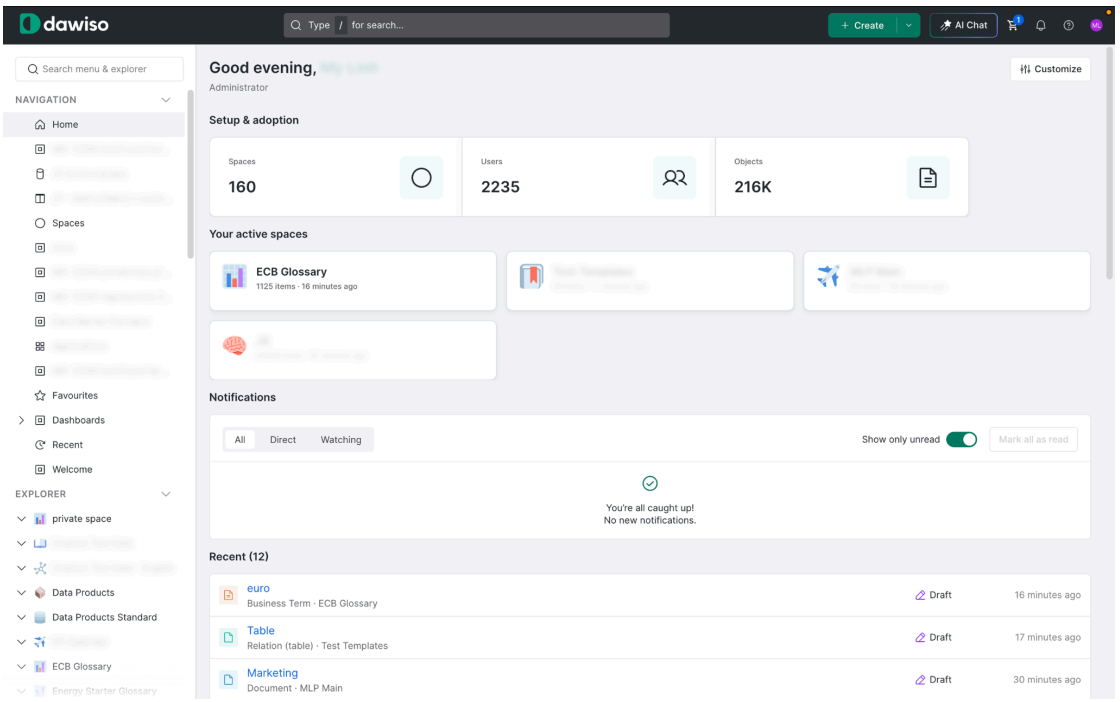


Selecting schemas to scan from a live-loaded list while creating a data source

Ingestion is stabilized and automatically resumes without needing a manual restart. Clicking **Retry** on a data source after a failed transfer now actually retries it, instead of permanently failing until a full **Restart**, which discards already-collected data.

Interface Redesign

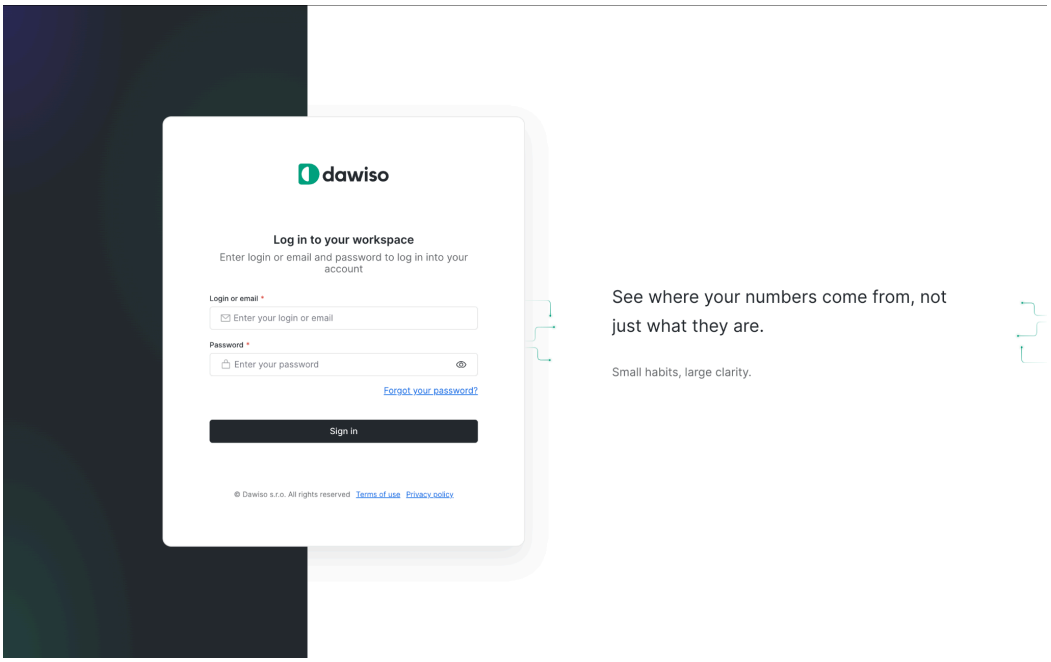
This release introduces a complete redesign of the Dawiso interface, focused on giving your data more room to breathe. The top navigation bar and page header merged into a single, thinner strip, freeing up working space that used to sit under a tall, dark bar. Tables and other components across the interface also became more compact, showing more rows at once without scrolling.



The Dawiso dashboard with the merged, thinner top navigation bar

Login Page

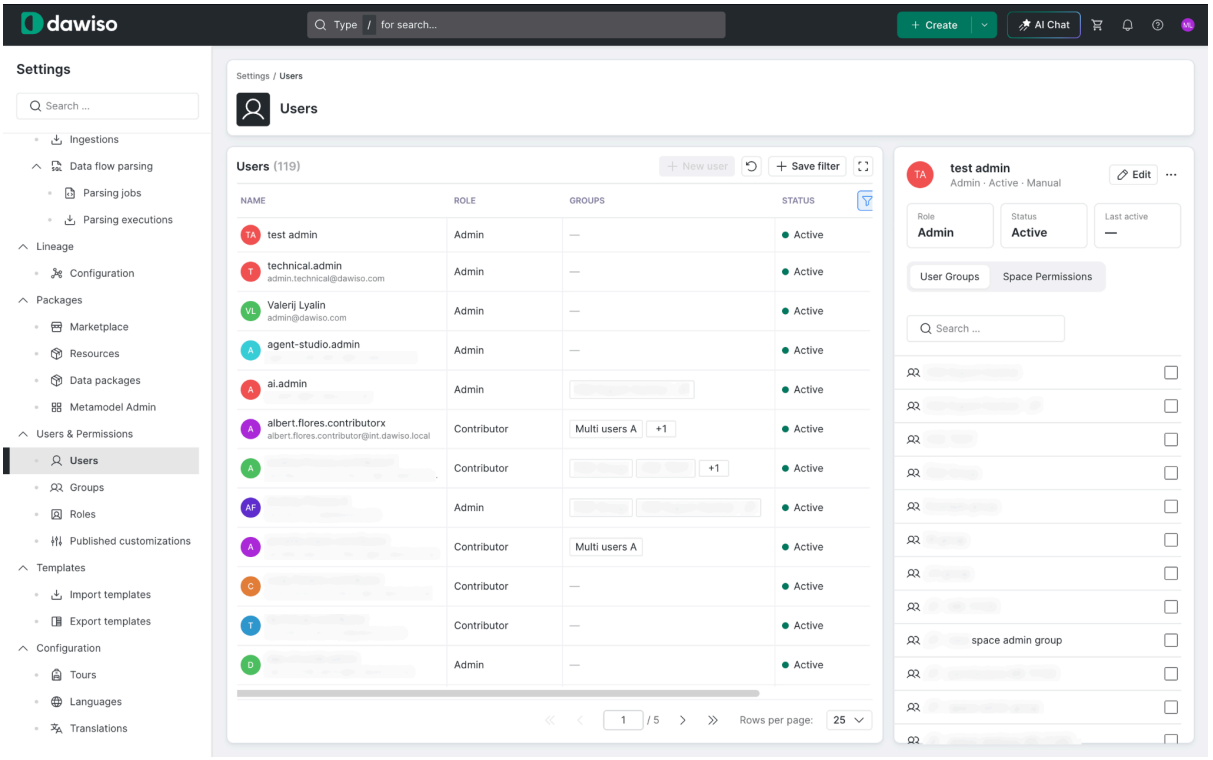
The sign-in page was redesigned to match the rest of the interface. A short tagline rotates below the form every 20 seconds. Administrators can replace the default set of taglines with their own.



The redesigned Dawiso login page, with a rotating tagline and terms-of-use and privacy-policy links

Users & Permissions

The **Users & Permissions** screen uses a master-detail layout: a list of users and groups on one side, and the details of whichever one is selected on the other, replacing the previous full-page navigation between list and detail views.



The Users screen with a user list on the left and the selected user's details, groups, and permissions on the right

Objects

Object identity, structure, and workflow all got more flexible. You can change an object's type without recreating it, choose exactly where a clone or move lands, and object and title names are capped at 200 characters, optionally with a pattern they must match. Moving, renaming, or transitioning an object is far less likely to quietly drop data or in-progress work along the way, and comments, relations, attachments, and workflow transitions each gained their own improvements.

Character Limit on Object and Title Names

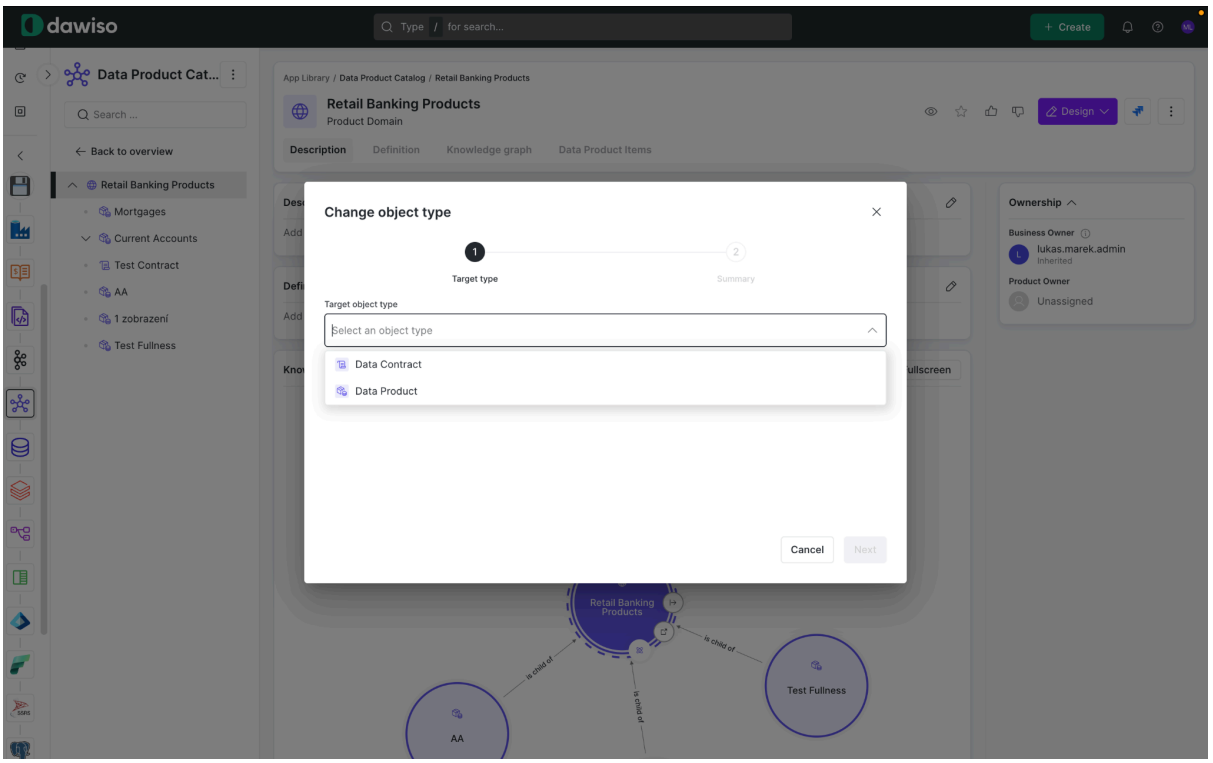
Object and title names are capped at 200 characters, with a live counter in both the inline title editor and the Rename dialog.

Object Name Patterns

Administrators can configure an object type to require names matching a pattern, checked whenever an object is created, renamed, or changes type, in the UI, MCP, and Public API alike. A name that doesn't match is rejected with an error showing the expected pattern.

Change Object Type

An object can change type directly, without recreating it or losing its history. Relations, comments, workflow state, and prior activity all carry over as-is, so reclassifying an object no longer means starting over.



Change object type dialog with a Target object type dropdown

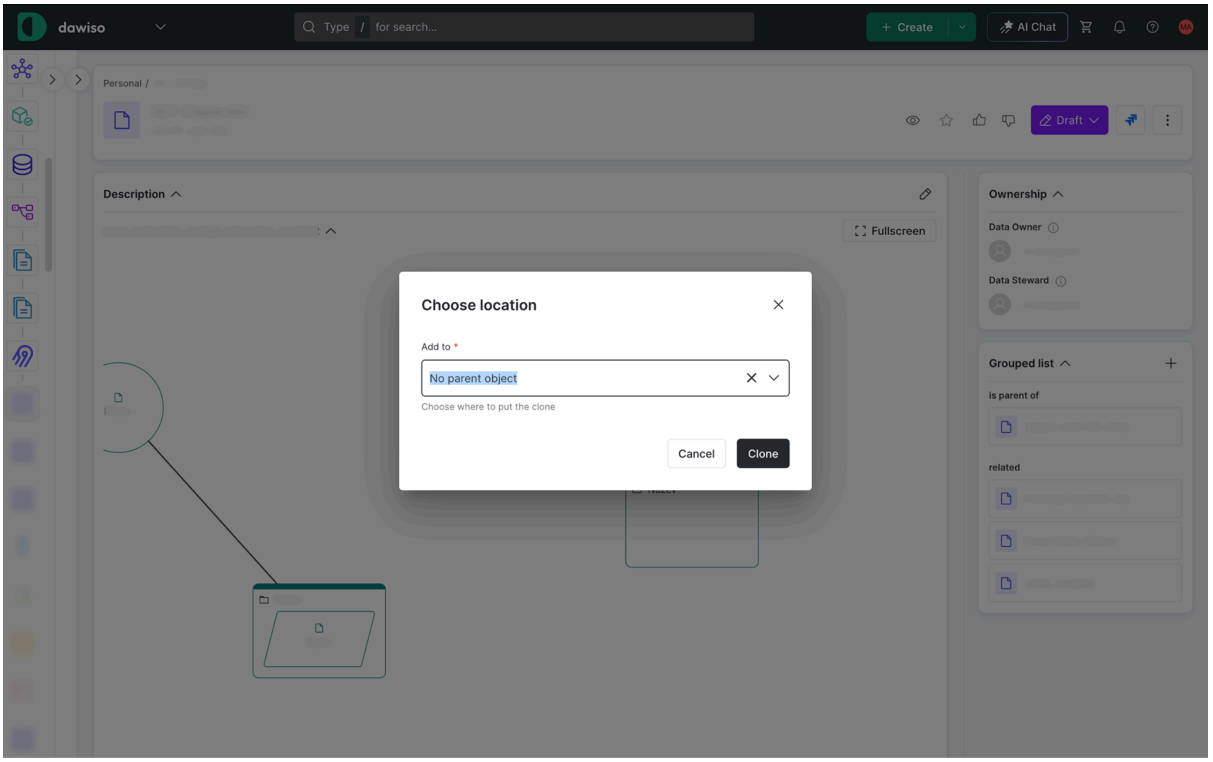
Changing type comes with a few rules:

- Attributes the new type doesn't define map to a compatible attribute, or stay stored but hidden.
- A new parent can be set when the new type doesn't allow the current one.
- Only the type, and the parent where applicable, changes. Everything else about the object's identity stays intact.
- Switching back to the original type restores any values that were hidden rather than mapped.

See Object management for the full procedure.

Choose a Clone or Move Destination

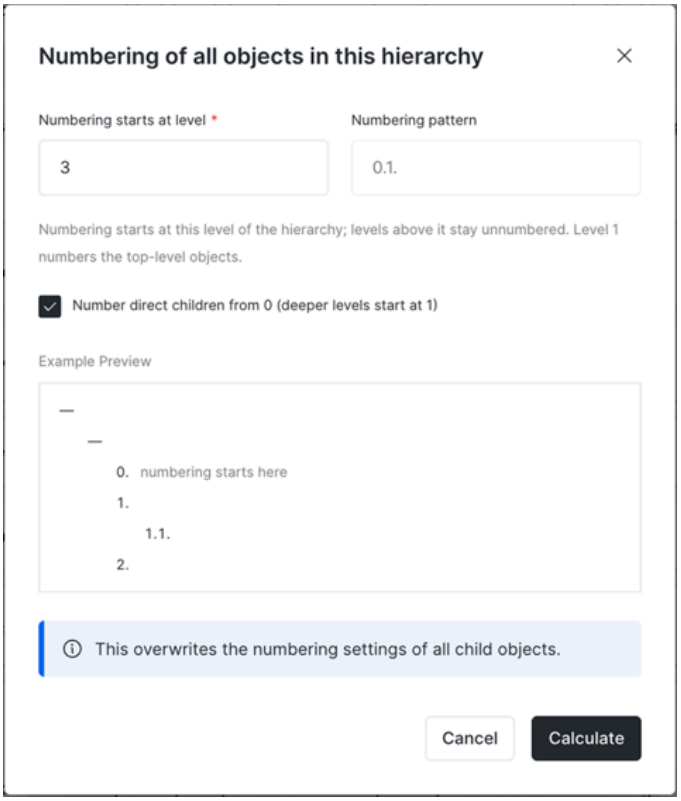
You can choose where a clone or move lands instead of it always going under the current object. Cloning an object shows a location picker with every valid target and its full path. Moving an object to a different application or type walks you through the same destination picker, with child object type and attribute mapping, and a summary that names exactly which attribute's values get overwritten before you confirm.



The clone destination picker showing valid target locations and their full paths

Object Numbering

The **Number direct children from 0** checkbox starts the whole numbering sequence at 0 instead of 1, in one click. A numbering pattern set on an object clears automatically whenever the object is moved, copied, or a parent's numbering is recalculated, and then reverts to inherited numbering. A package that installs an invalid numbering pattern is rejected at install time, instead of later breaking concept publishing or leaving objects with no name in the hierarchy. See Objects for the full numbering reference.



Numbering dialog with Number direct children from 0 checked

Hierarchy & Inheritance Reliability

Several object and package operations no longer lose data silently. Moving an object to a new parent also moves any automatically created relation between it and its previous parent, such as one configured with `is_creatable_on_child_add` (see Features (relation type)). Previously, that relation was left pointing at the old parent instead.

Additional fixes further improve hierarchy and inheritance reliability:

- Renaming a child object no longer deletes its inherited attributes.
- Deleting a package no longer deletes unrelated system types along with it.
- Creating a child object respects the object type's "inherit from parent" role setting.
- Cloning an object preserves its inherited roles and attributes.
- A locked transition button's tooltip explains what's missing (attributes, roles, or relations) before you go looking.
- An unsaved edited attribute on a transition offers **Save**, **Discard**, or **Cancel** instead of losing the change silently.

Comment Moderation and Revision Filtering

Comment moderation is workflow-aware. A role the current workflow state allows can remove another user's comment, and both the author and anyone who replied are notified. Deleted and added comments show clearly when comparing two versions of an object. When an object type has a revision attribute configured, comments can be filtered to just the revision you're interested in.

Live-Updating Relations

Every view of an object's relations updates the moment a relation changes, wherever that change happened, instead of only in the place you clicked. A teammate's edit shows up in your right panel without a reload.

Bulk-Remove Relations

A relations table can enter edit mode to select and remove several relations at once, instead of deleting them one by one. The removal is all or nothing. If it fails, none of the selected relations are removed.

Link Attachments

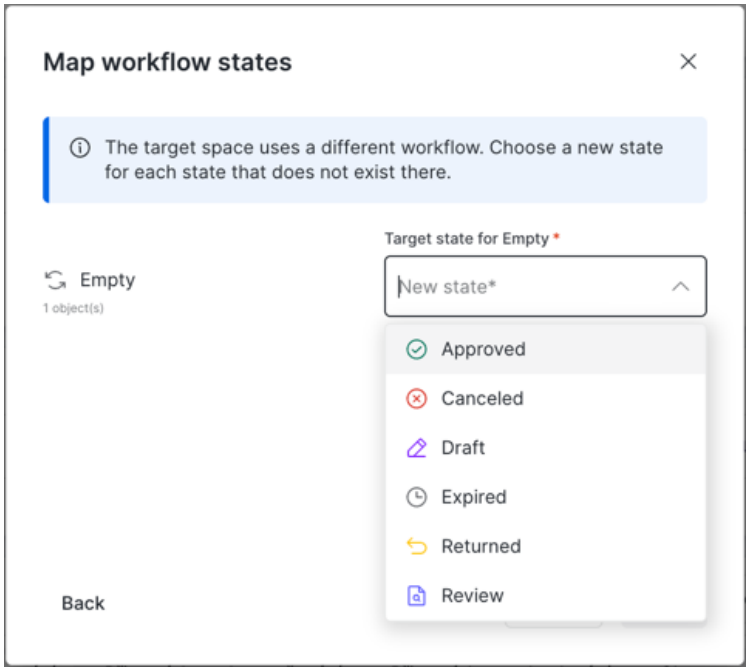
An attachment can be an external link instead of an uploaded file, for example a path on a shared network drive. Dawiso stores only the URL. An HTTP or HTTPS link opens in a new window, and a `file://` link gets a **Copy link** action since browsers usually block opening local file links directly from a web app.

Required Attribute Before Embedding

An administrator can require a specific attribute to be filled in before an object can be embedded in another object's text. Picking an object that doesn't meet this condition names the missing attribute and blocks the embed until it's filled in or a different object is picked.

Workflow Transitions

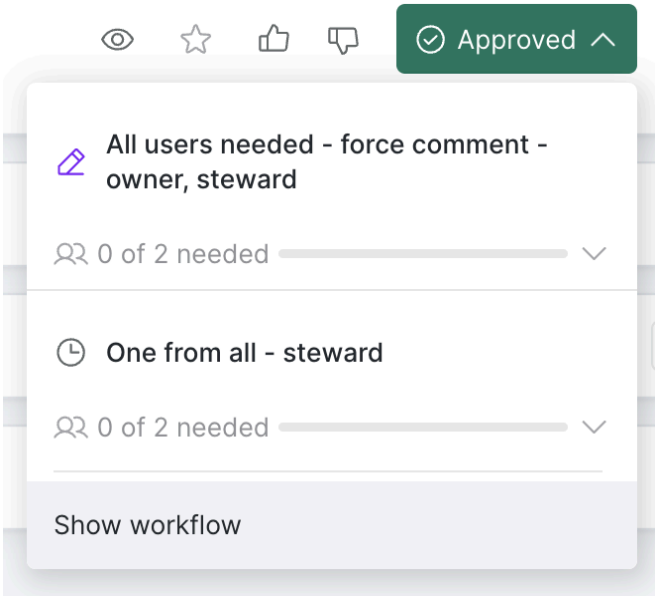
Moving an object to a space with a different workflow maps its state directly in the move dialog. Only states missing from the target workflow require mapping, and the move can't finish until every listed state has one. See Object management for the full move procedure.



Map workflow states dialog with a dropdown of target states

Workflow Transitions as Cards

Clicking a workflow state opens a card for each available transition, with live approval progress, for example **Waiting for approval** with a `0/2` badge, visible to every approver the moment the first one approves.



A workflow state transition card showing approval progress as a 0 of 2 badge

Change an Application's Workflow

Admins and space admins can switch which workflow a space application uses directly from the UI, instead of only through a package.

Impact Analysis Improvements

Impact analysis tables can resolve several impacts at once instead of one by one. A new table also tracks impacts caused by embedded-text changes in the object you're viewing, and every table can be filtered by the owner.

Onboarding & Guided Tours

This release begins a complete rebuild of onboarding: a first-login introduction, a running checklist of first steps, contextual hints on individual interface elements, and guided tours through specific workflows, with each piece configurable by administrators. The new experience ships disabled by default while remaining rough edges are worked out, with a full rollout planned for a later release.

First-Login Walkthrough

The first time someone logs in, a short interactive walkthrough greets them. It visualizes the account's connected apps as a live graph, introduces the AI layer running underneath Dawiso, and lets the new user try asking it something real. The final screen asks what brought them to Dawiso: **Find data**, **Understand data**, **Manage data**, or **Govern data**, each opening a guided tour suited to that choice. **Explore on my own** skips the whole sequence.

Guided Tours

A tour walks through a feature step by step, calling out one screen element at a time. Certain steps require actually completing the action shown rather than skipping ahead with a click. Every available tour lives in the **Tours & guides** section of the **Help** menu, each showing its own status (**Completed**, **In progress**, **Ended early**, or **Not started**), and any of them can be replayed from scratch whenever you like. Completing a tour that leads into another offers a **Start tour** button to launch that follow-up immediately.

Administrators can switch guided tours off for the whole environment. See Guided tours for the full reference.

Onboarding Checklist

Out of the box, a checklist in the top navigation bar tracks progress through ten early tasks (connecting a data source, assigning owners, setting up classifications, and inviting colleagues among them), each one checking itself off the instant you take the matching action anywhere in the app. A task with a linked tour offers a **Show me how** button that starts it directly.

Two panel actions are available: **Minimize**, which keeps the checklist available in a collapsed state, and **Dismiss**, which closes it entirely. A dismissed checklist can be reopened later from the help menu, or restored immediately with the **Undo** option on the confirmation toast. Administrators can turn off parts of onboarding for the whole instance, and chain a guided tour to a specific task. See Onboarding for the full reference.

Onboarding Hints

An onboarding hint marks a single interface element with a small pulsing dot the first time a user encounters it. Clicking it pops open a short explanation of what that element does, with a **Got it** button to dismiss it and, when a tour is linked, a **Show me how** button to jump straight into it.

Hints never block the page, and only one shows per browser tab at a time. Each one is marked seen the moment it appears and never shown to that user again, on any device. The whole feature is disabled by default. Turning it on still requires an installed package to supply the actual hint text and targets. See Onboarding for the full reference.

Packages

This release introduces several updates to packages and how you can work with them, including a UI-driven way to build custom applications without hand-writing package JSON.

Attribute Validation

A package can restrict an attribute type's values to a regular-expression pattern, using a `regex_validation` feature with `regex_validation_error_translation_key` for the message shown when a value fails it. This applies to single-line text, multi-line text, and rich text alike.

A mismatched value is rejected whether it comes from:

- A normal edit or bulk edit
- The `Set Attribute` automation action
- An Excel import
- Object creation

An empty value passes unless the attribute is also marked mandatory. For a rich-text pattern that's a plain length limit, the editor shows a built-in "Too long: max N characters" message with a character counter, instead of requiring the custom error-message feature.

Both features must be defined together. Installing a package with only one of the pair fails at install time, and the pattern itself must compile under the .NET engine's ECMAScript-compatible mode or the same failure applies. See Attribute Types & Features for the full reference.

Components as Widgets

A template component can also serve as a home screen widget thanks to the new `widget` property.

This is the package-authoring mechanism behind homepage and dashboard widgets. The user-facing side of building a homepage or dashboard from widgets is covered under Customizable Layouts Using Widgets. See components for the full schema reference.

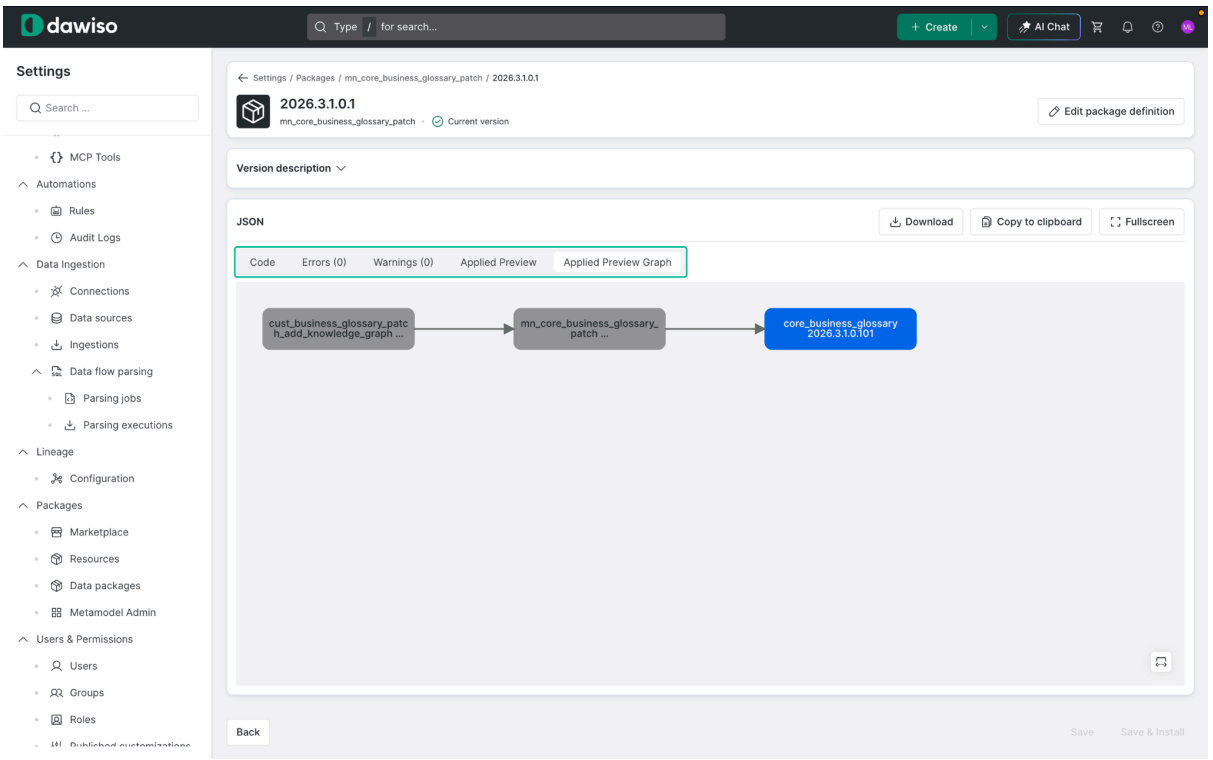
Metadata Sync

A package's metadata syncs automatically from a central file, instead of needing a manual update wherever the package is installed.

Package Editor

A package version opens in a full editor for reviewing or changing its JSON definition, name, or description, surfacing its number, key, and current-install status up top:

- **Tabs**: The raw **Code**, any validation **Errors** or **Warnings**, and an **Applied Preview** of the fully resolved definition. For patch packages, this becomes a navigable **Applied Preview Graph** instead of raw JSON.
- **One draft at a time**: Directly changing an already-installed or past version and saving spins up a fresh draft from those edits, and locks every other version to read-only until you either finish or abandon it.
- **Separate save and install actions**: **Save** stores a draft, and **Save & Install** does both, so work in progress can sit without being deployed. You can also save with `Ctrl+S` (or `Cmd+S`).



A package version's Applied Preview Graph tab, showing a patch chain resolving to the base package

Every patch links to the package underneath it in the **Applied Preview Graph**, so a multi-patch chain reads top to bottom as a path ending at the base package.

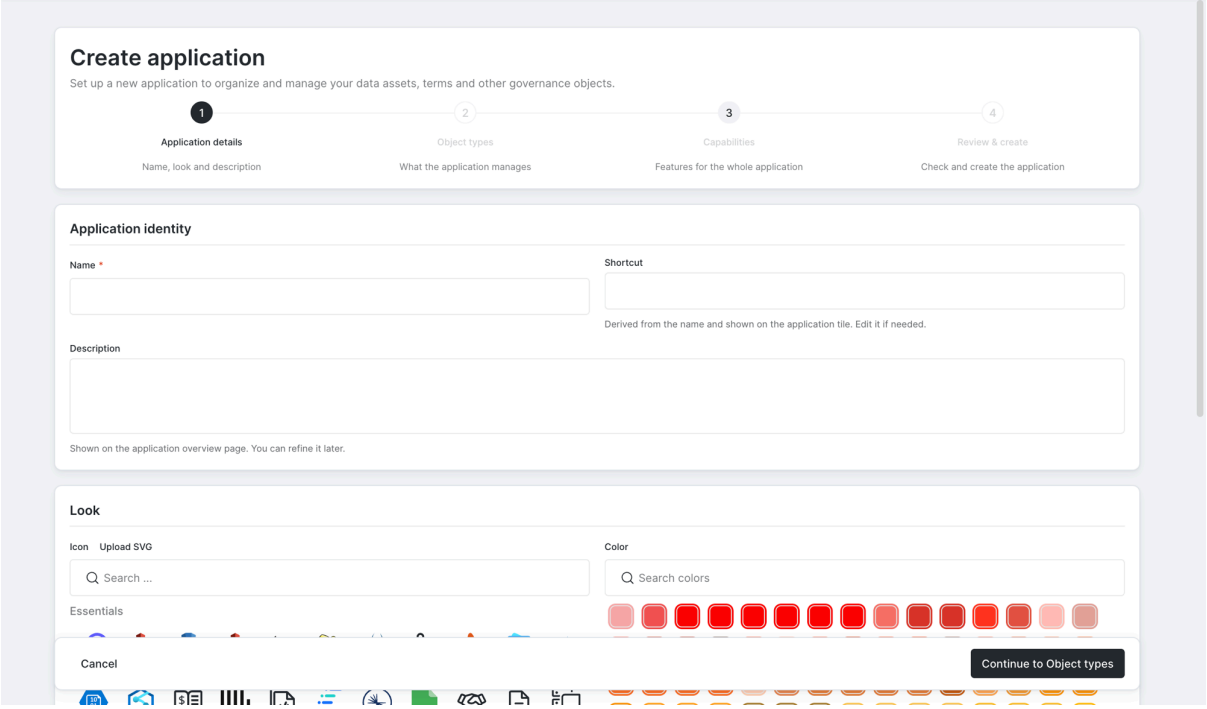
Patch packages can target a component nested inside a panel individually, but not one inside a splitter or tab group, since those sections carry no identifying key a patch could reference.

Administrators can also discard a draft package version they no longer need, instead of being stuck with it until the next save. Discard is disabled while a task is still running for the draft, or once the version is installed or versioned.

See Working with packages for the full walkthrough, or Patch Packages: Advanced Guide for the patch-specific reference.

Build Custom Applications Without Package JSON

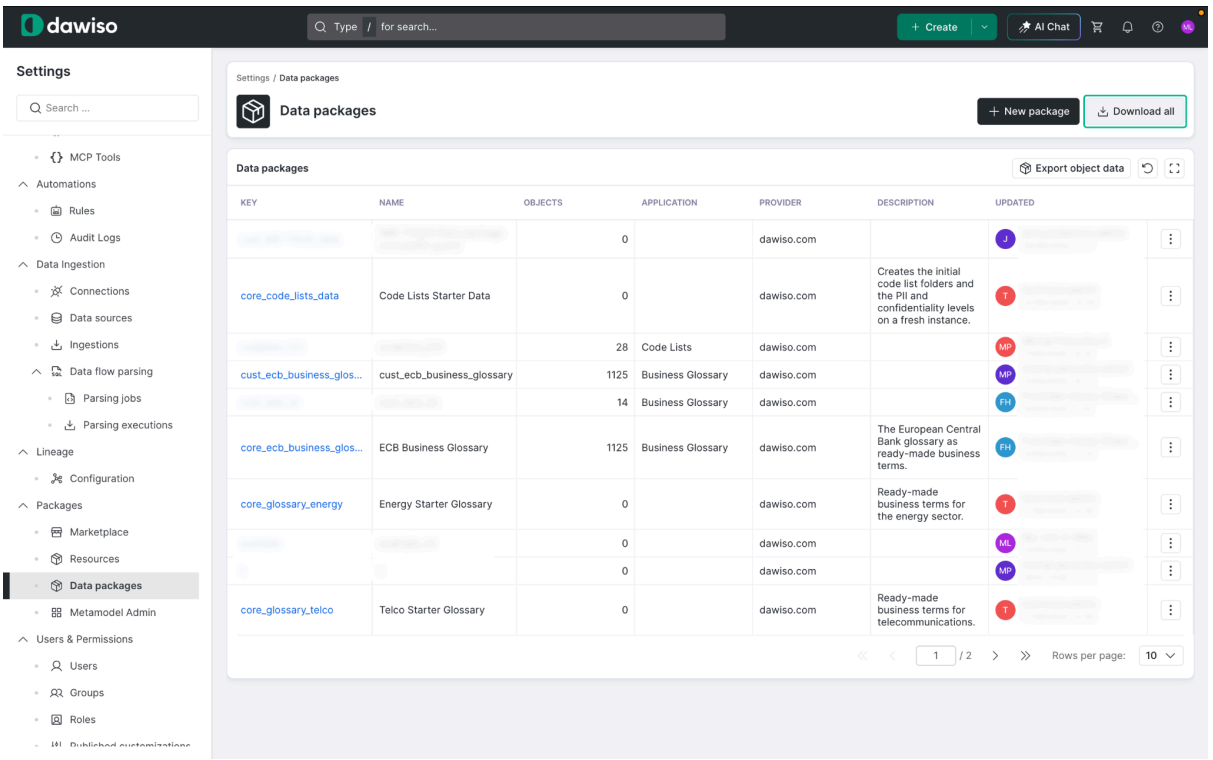
Building a custom application no longer requires hand-writing package JSON. Create a new application together with its object types in one save. Edit application to update one created this way. Add new object types to an existing application, and define new custom attributes or reuse a matching standard one. Configuring any of these produces a JSON package, installed and versioned like any other package.



The Create application page building an application and its object types in one save

Packages Updates

Downloading every package at once doesn't block on an immediate file download. Dawiso assembles the archive out of view and pings you with a notification once it's ready, linking both the file itself and the export task's status page. Leaving the page doesn't interrupt the export.



A packages list with a Download all button in the top-right corner

Packages: Marketplace Improvements

Enabled by Default

New cloud instances ship with the Marketplace enabled by default, so a fresh environment starts with useful defaults already in place rather than a blank slate. Packages flagged for automatic installation set themselves up on their own soon after the environment spins up, with nobody needing to open the Marketplace and click install. See Marketplace for the full reference.

Prerequisite Resolution

Installing a package from the Marketplace that depends on others doesn't stop to ask for each one separately. Dawiso works out the full chain of missing prerequisites and installs them together in one step, rather than making an admin resolve and install each dependency by hand before retrying the original package.

Marketplace Checks for Updates Right After Deployment

Marketplace checks for package updates as soon as the platform restarts, in addition to its nightly schedule, so a new package becomes available right after an upgrade instead of waiting for the next scheduled check.

Reserved `core` Prefix

The Marketplace enforces the `core` package-key prefix as a hard restriction rather than just a best practice: any package key starting with `core`, regardless of case, is blocked at creation, rename, or install time. That prefix is reserved for Dawiso's own default packages. See Working with packages for the full reference.

Packages: New `data` Type

A `data` package type moves real content, not configuration, from one Dawiso environment to another.

Automatic Installation

The `autoInstall` property is valid only on regular and `data` packages, not `patch` or `automation`, and a package only self-installs once every package it `dependsOn` is already installed or is also auto-installing in the same pass. See package for the full schema reference.

Export Object Data

Export object data, an admin action in **Settings**, turns a chosen application's objects in one space directly into package content, no hand-written JSON required.

- Running the export again on the same application reuses the same object keys, so it only picks up what's new or changed instead of creating duplicates every time.
- Each exported scope can also be set to only seed its content when the destination is still untouched, so a starter dataset never silently overwrites work someone already did there.

- After install, the package detail page reports whether the data **seeded**, was **skipped**, or was **partially seeded**, so an admin can tell at a glance whether the package did what it was supposed to.

See Export object data for the full walkthrough.

New Schema

`data` packages introduce four new schema pieces:

- A new package `type`: `data`.
- A new `data` asset: an array of scopes, each declaring a `space` (create one) or `spaceRef` (use an existing one), a `dataInstall` condition, and the `objects`, `objectImages`, and `objectRelations` that scope installs.
- A new `objects` asset: creates real catalog objects.
- A new `objectImages` asset: attaches a binary image to a package-created object.
- A new `objectRelations` asset: declares relations between objects the same package creates.

Product Tours and Catalog Objects

Previously, something like a guided tour only worked where it was originally built. Moving it elsewhere meant rebuilding every step from scratch, since its internal pointers were tied to that specific environment.

A `data` package fixes this by matching each piece up by name rather than by internal ID, so the tour lands correctly no matter which environment installs it. Running the same install again later folds updates into the tour that's already there instead of creating a second copy.

Two kinds of content ship this way today: product tours, and whole sets of catalog objects too. See Data packages for the tour-export walkthrough.

Performance

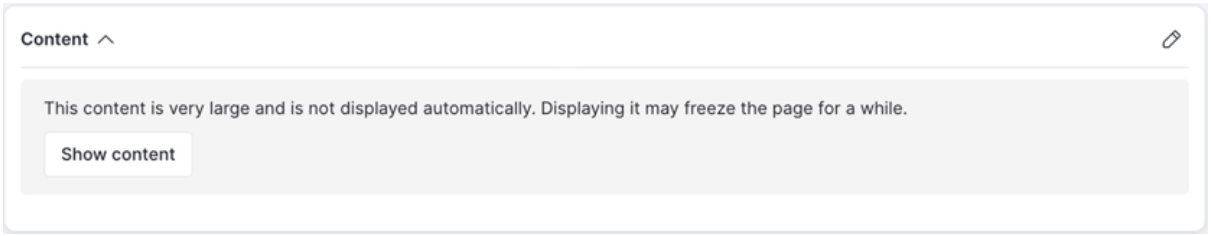
Several backend and diagram fixes make the platform more responsive under load, especially with many objects, deeply embedded content, or large rich text.

Performance Optimizations

- **Faster app start**: Opening an object's detail page no longer downloads a large, mostly-unused bundle of extra code in the background, and the app also starts faster after login.
- **Faster change-log filtering**: A targeted index replaces a slow full scan, so filtering a large change history stays fast even with hundreds of thousands of records.
- **Faster cloning and bulk writes**: Cloning large object trees and importing large Excel files in update mode no longer lock the database or block other users' concurrent work.
- **Faster package installs**: Installing a package no longer locks the whole app for other users while it runs. Opening an object or loading a template during an install used to hang until the install finished. It now responds right away.
- **Faster saves for embedded text and automation**: Saving an attribute embedded in many other objects no longer locks the database or blocks other users' concurrent saves on large saves and automation runs. Objects that embed the changed attribute show the new text a few seconds later instead of instantly. This applies to manual saves, the `setAttribute` automation action, and clearing a value.
- **Faster templates**: Object-type templates come from a per-version cache instead of being rebuilt every time, cutting a multi-second wait down to a fraction of a second.
- **Fewer redundant redraws**: While someone else is actively editing an object you have open, incoming updates are batched together instead of forcing a full page redraw for every single one.
- **Smoother large diagrams**: Selecting a node in a large knowledge graph is near-instant. The same fix applies to data lineage, ER, and conceptual model diagrams.

Large Rich Text Loads Safely

Rich text over 250,000 characters no longer renders the moment you open an object, and the same limit covers the Activities changelog's version comparison view. A **Show content** button takes over instead, so opening a large document or comparison doesn't freeze the page. See WYSIWYG HTML editor for the full reference.



A Content panel showing a message that the content is too large to display automatically, with a Show content button

Operational Monitoring

New health endpoints for Elasticsearch, the database, and the WebSocket hub, plus a Prometheus metrics endpoint covering tasks, ingestion, user sync, and the scheduler, so Zabbix or Grafana can watch Dawiso alongside the rest of your infrastructure.

Security & Permissions

Administrators can restrict hierarchy changes to admin accounts only, tightening who's allowed to restructure the catalog.

A user-relation picker, like the one used to assign an owner or steward, can also be scoped to a specific pool of people. Instead of offering every user or group, it can offer only those who already hold a different relation on the same object, such as limiting reviewer candidates to people already assigned as a contributor. See codetable-label-user for the full schema reference.

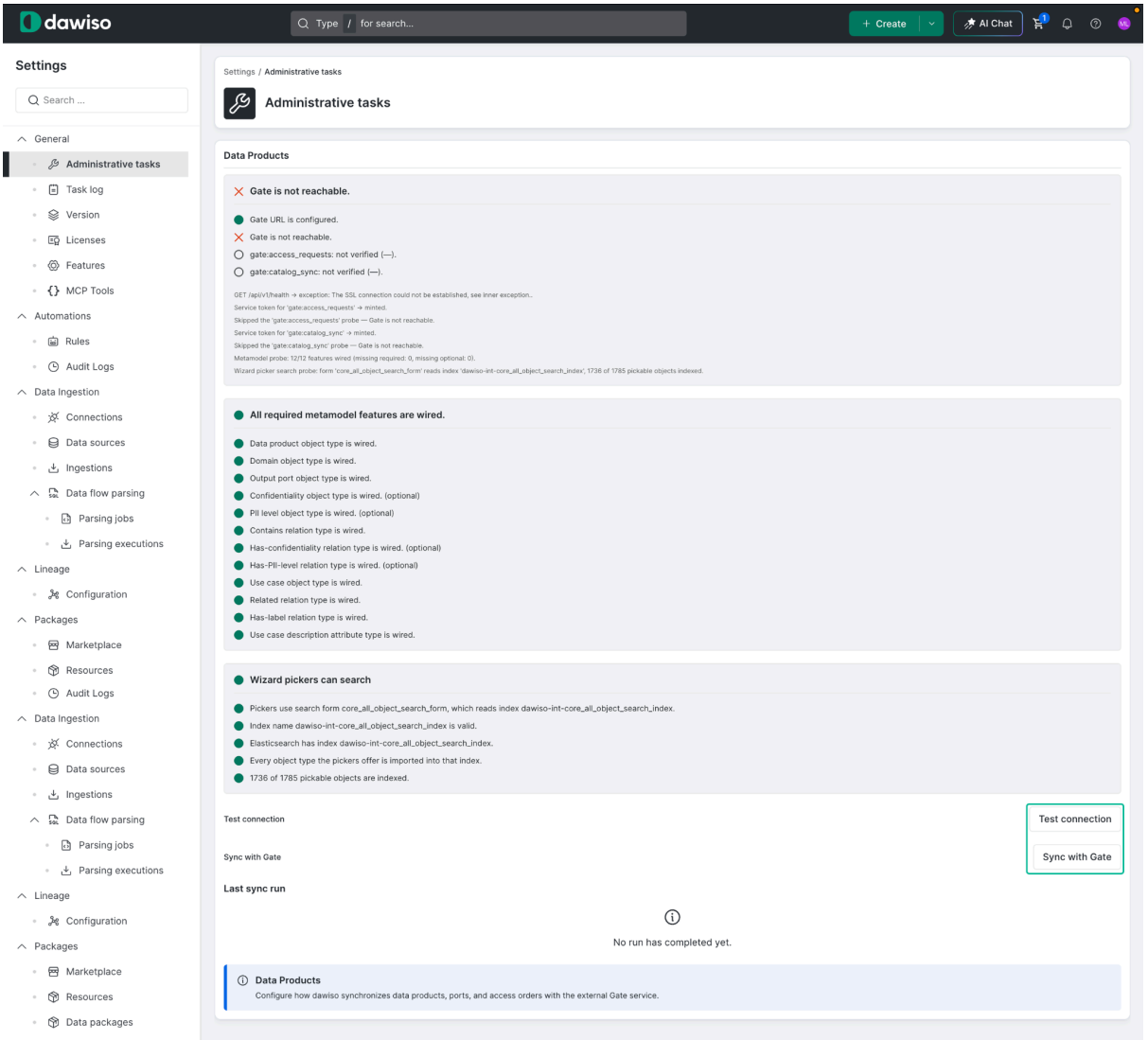
Unified Platform

Dawiso increasingly works as a platform of connected apps rather than one application. This adds Gate, a new access-management app for data products, and a shared identity service connecting Dawiso and Gate together.

Gate

Gate is a new external app that oversees data products and the output ports they expose, with its own screen for platform mapping, tiers, classification templates, and a nightly sync. Full configuration details for that screen are being documented separately.

Within Dawiso, a **Data Products** panel under **Settings** lets an admin confirm the Gate integration is healthy: **Test connection** confirms Dawiso can reach Gate, and **Sync with Gate** forces an immediate sync instead of waiting for the regular event-driven one. A **Last sync run** section reports how the most recent sync went, including what delivered, what failed, and what came through without a classification, each with a recommended fix. See General settings for the full reference.



Data Products panel with Test connection and Sync with Gate buttons

Deleting a data product or domain in Dawiso retires it in Gate, so Gate no longer offers it for access requests. An object restored in Dawiso after deletion stays retired. For details, see Data products and Gate integration status.

Identity Service

A shared identity service sits underneath Dawiso and Gate, so navigation between the two authenticates through one place instead of two separate logins. Dawiso and Gate each connect to it through a single OAuth2 client, and neither side stores a password the other relies on. It's also the service issuing the short-lived tokens the Gate integration status panel checks for.

UX/UI Enhancements

Tooltips shed their sticky behavior, a new in-app satisfaction survey rolled out, table filters are easier to read, and additional improvements landed in navigation, spaces, the **Fullness score**, and self-service actions on object pages.

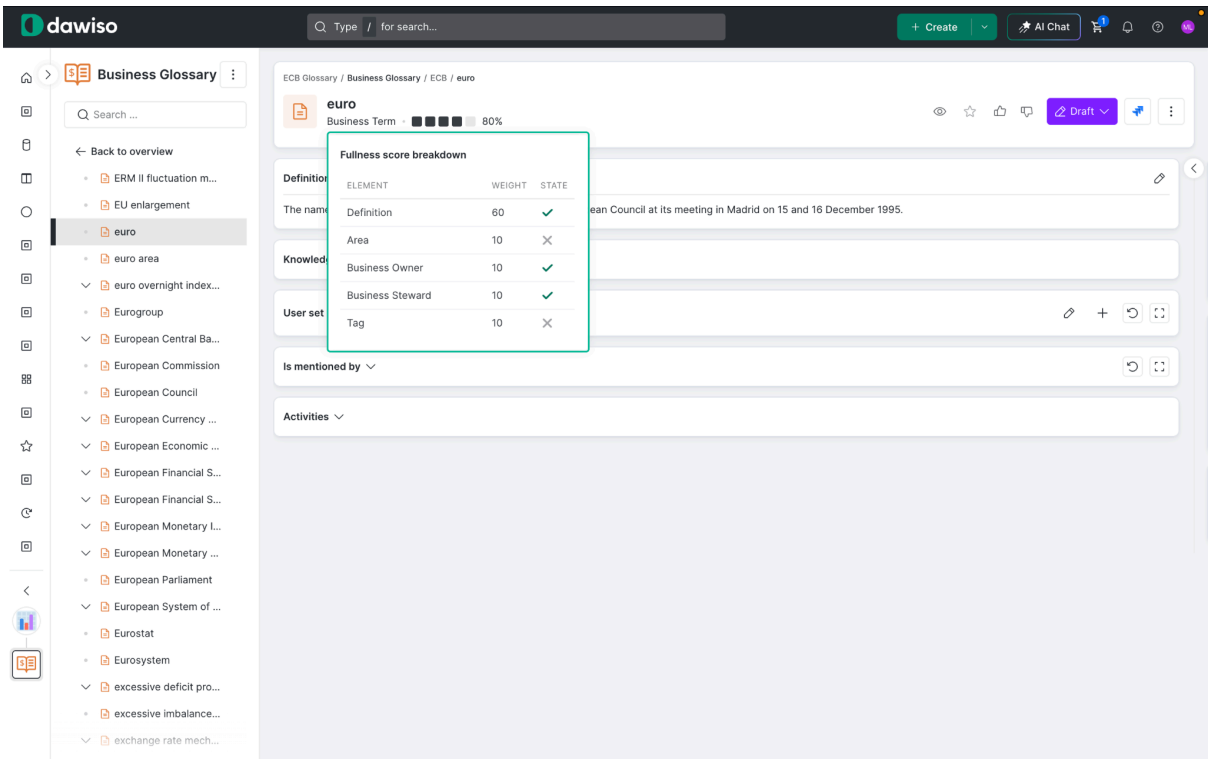
Table Filters Are Easier to Read

A table column's filter list widens to fit its longest value, instead of truncating every item to one line at a fixed width. Values that still don't fit wrap to a second line before truncating.

Fullness Score Detailed Overview

Hovering over a Fullness score opens a per-item checklist instead of a single number. It lists each attribute, relation, and user relation the score counts, how much each is worth, and whether it's already been filled in or still blank.

That's enough to spot what's missing without switching over to the separate weight table. The popup is view-only and doesn't total anything itself, so the percentage next to the object stays the one number on screen. See Business Glossary for the full reference.



Fullness score breakdown popup listing each element with its weight and state

Navigation

The left-side object hierarchy auto-expands to three levels by default when you open it.

NPS Survey

A short 0–10 satisfaction survey with an open question shows up in the corner of the screen. Server-side logic decides when to trigger it, based on real usage rather than immediately after login, and administrators can turn it off.

Spaces Icons

Creating a space offers a wider selection of icons to choose from.

Tooltips Improvement

Tooltips across the app are not sticky by default, so moving the cursor away closes them right away. Tooltips over content someone might need to copy, such as names, emails, paths, IDs, or error messages, stay open as before.

User Relation Actions

Object pages can offer self-service buttons for claiming or dropping a user relation, such as acknowledging ownership, instead of always requiring an administrator to assign it.

- A button's own label can change once it's clicked.
- Its visibility can be restricted to users who already hold a different, specified relation on the object.
- The same visibility restriction also applies to section-title form actions, so a form-opening button can stay hidden from anyone without the right relation.
- A user-picker field in forms can allow selecting more than one person at once, instead of only one.

WYSIWYG HTML Editor

The text editor picked up formatting tools, safer attachment handling, richer embeds, and an AI writer that now preserves your formatting.

Formatting Tools

- The toolbar stays pinned in view while you scroll its text, instead of scrolling away with it.
- A **Format Painter** copies formatting from one part of your text to another.
- A **Page break** button inserts a break that carries through to the Word export.
- Images can be pasted directly from the clipboard, in addition to the existing **Insert Image** button and drag-and-drop.
- Links no longer inherit unwanted formatting from the surrounding text.

Attachment Behavior

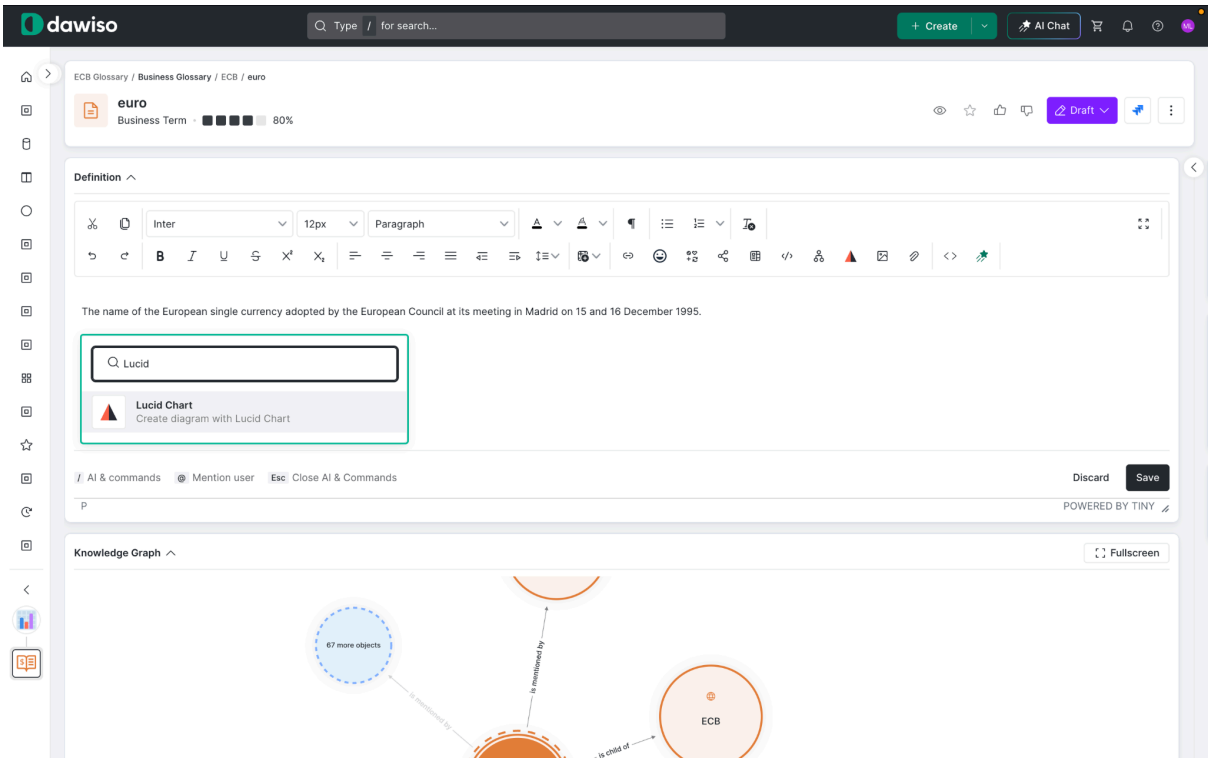
Attachments removed before being saved no longer linger in the object's Attachments list afterward, and removing an attachment from an object's text after saving hides it from the

Attachments section and blocks its download there, without deleting the file. Attachments uploaded directly in the **Attachments** section are unaffected.

Upgrading to this release also runs a one-time background task that applies this retroactively to editor-uploaded attachments already removed from text before the upgrade, so some of them disappear from the **Attachments** section shortly after upgrading.

Diagram Embeds

The editor's diagram options grew beyond Draw.io. Typing / also offers a Lucidchart embed. Pasting a Lucidchart link turns it into a live, interactive diagram inside the description, rather than a static image. The Markdown editor gained the same embedded-content capability rich text already had.



The / slash menu in the rich text editor, showing the Lucid Chart embed option

Help Me Write

The AI writer keeps your text's tables, headings, and other formatting instead of flattening its reply to plain text. Replacing a table cell by cell keeps each reply aligned to its own cell, and a status message shows while the reply generates.

See WYSIWYG HTML editor for the full reference.

Fixes

Filters That Silently Returned Too Few Results

- The object name filter also matches objects without numbering. Previously it matched only the numbering-prefixed name, so unnumbered objects never showed up in search results such as the Comments dashboard.
- A Yes/No attribute filter matches the value regardless of whether it's stored as `true`/`false` or `1`/`0`, instead of returning zero rows for the format it didn't expect.

Permission Recalculation After Identity Service User Changes

On instances using identity service for login, reactivating a user, creating a new user, or changing a user's role correctly recalculates their space permissions right away. Previously, a user in one of these situations could be missing access to public spaces until an unrelated permission recalculation happened to run.

Word and PDF Exports Keep Their Spacing

A space that was the entire content of a formatting tag, such as bold, italic, underline, superscript, or subscript text, no longer disappears from Word and PDF exports, including the export that compares two versions of an object. Existing documents export correctly without resaving.

Smaller Fixes

- A chart component's configured object-type filter is applied. Previously it showed objects of every type regardless of the filter.
- The object type filter also appears while searching the default hierarchy view, matching every other hierarchy type.
- The application overview page stacks its columns at narrow window widths, instead of squeezing a table panel down to about 160 pixels wide.