Arveo Flex Client Documentation

Table of Contents

1. Introduction . . . 2
2. User workflows . . . 3
   2.1. Access and permissions . . . 3
   2.2. Choose a scenario and language . . . 3
   2.3. Work in a Simple Document scenario . . . 5
   2.4. Work in a Record scenario . . . 6
      2.4.1. Share documents between records (m:n) . . . 7
   2.5. Find and inspect documents . . . 9
   2.6. Create, edit and delete . . . 10
   2.7. Choose content and a viewer . . . 12
      2.7.1. Read a PDF . . . 13
      2.7.2. View, edit and save PDF annotations . . . 14
      2.7.3. Inspect an image . . . 15
      2.7.4. Read a TIFF . . . 16
      2.7.5. Read text content . . . 17
      2.7.6. Downloads, unavailable content and historical viewing . . . 17
   2.8. Versions and comments . . . 18
   2.9. Recover from errors and unfinished work . . . 18
   2.10. Document workspace settings . . . 19
3. Litigation holds . . . 20
4. Retention dates and protected lifecycle actions . . . 21
5. Administration . . . 22
   5.1. Enter administration . . . 22
   5.2. Choose a scenario structure . . . 23
      5.2.1. Create a Simple Document scenario . . . 24
      5.2.2. Create a one-to-many Record scenario . . . 25
      5.2.3. Create and configure an m:n Record scenario in V2 . . . 26
   5.3. Configure a Simple Document scenario . . . 27
   5.4. Configure a Record scenario . . . 29
   5.5. General settings and attributes . . . 30
   5.6. Content, versions and record deletion . . . 31
   5.7. Configure views . . . 32
      5.7.1. Understand the view matrices . . . 34
      5.7.2. Configure lookup-backed attributes . . . 34
   5.8. Organize scenarios into categories . . . 36
      5.8.1. Choose shared dashboard presentation . . . 37
      5.8.2. Import and export categories only . . . 37
   5.9. Save, duplicate, order, import and export . . . 38

5.9.1. Save and discard scopes . . . 38
5.9.2. Configuration transfer and legacy files . . . 38
5.10. Categories and dashboard presentation . . . 39
5.11. Manage languages and translations . . . 40
5.12. Validation and concurrent editing . . . 41
5.13. Import reference . . . 41
5.13.1. Configuration import fields . . . 41
5.14. Configure scenario permissions . . . 46
5.15. Browse the permission catalog . . . 47
5.15.1. Independent document permissions and upgrades . . . 49
5.16. Configuration QA behavior . . . 49
5.17. Recycle-bin configuration (in development) . . . 50
5.18. Lifecycle policy defaults and repository constraints . . . 50
5.19. Recycle Bin view configuration . . . 52
6. Configure litigation holds . . . 53
6.1. Understanding type settings without a separate reference . . . 53
7. Configure retention management . . . 55

Version 0.1.4; source 3ebf362f0877f1fad8aa6d291b94f15a0dc2ac39.

Chapter 1. Introduction

This guide explains how to use Arveo Flex Client and administer its scenarios, categories, dashboard presentation and translations. Start with User workflows for everyday tasks; Administration covers configuration and import/export for administrators.

Chapter 2. User workflows

2.1. Access and permissions

Open Arveo Shell using the address supplied by your administrator and sign in with your own account. Open **Flex Client** from the Shell dashboard. The application runs inside Shell; a bookmark must use the Shell entry point. If authentication expires, sign in again through Shell. Before repeating a failed save, reload and check whether the change was already persisted.

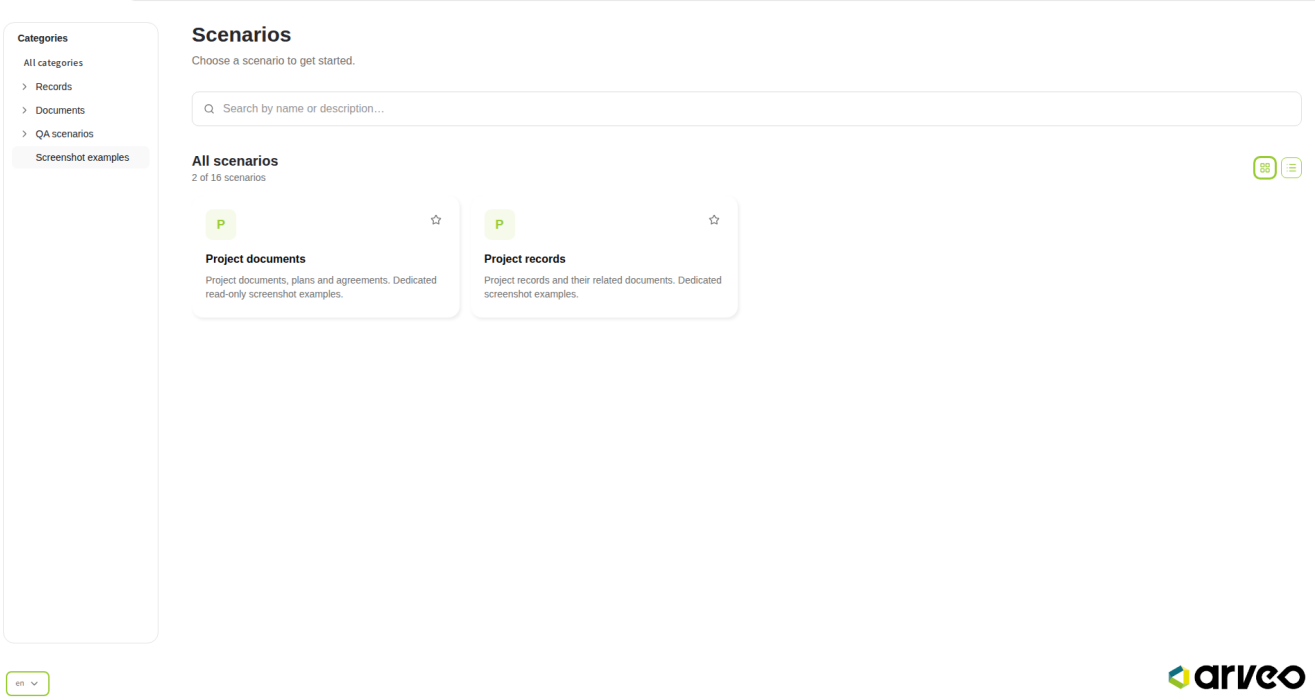
A normal user sees only visible scenarios with an explicit scenario-access grant. Direct links require the same grant. Access alone does not permit reading records, documents or content: actions have independent grants and must also be enabled in configuration. Administrators can configure all scenarios but receive no automatic scenario grants. Repository authorization still applies independently. Ask your administrator when access is denied; do not repeatedly submit an action.

Columns, details and create/update fields may have independent visibility and editability rules. Hidden fields do not appear; visible restricted fields are read-only. A form cannot submit if a required restricted field cannot be satisfied. Navigation refreshes permissions. If permissions change, protected views disappear and further unauthorized submissions are blocked; unsaved input is retained only in memory. An ECR denial preserves input and does not remove the Flex grant.

2.2. Choose a scenario and language

Select English or Deutsch with the dashboard language selector. Scenario names, descriptions, category labels and field labels use the configured translations, falling back to the default text when a translation is missing. Open a scenario from a card or list row. To read a long description in full, hover over a scenario card or list row for one second. Keyboard focus on the scenario title opens the same enlarged preview. Move into the preview to scroll or select its text; press Escape to close it. Use the breadcrumb to return to the dashboard or to the parent record. Hidden scenarios are not available through ordinary navigation.

Use the dashboard search and card/list view controls to find scenarios, and mark favorites for quick access. When your administrator uses tree presentation, choose a category in the sidebar to narrow navigation. Flat presentation does not use that category hierarchy. Scroll within the scenario card or list area to reach more scenarios. The dashboard heading, search, view controls, pagination and language footer remain visible. Favorites and long category trees have their own scroll areas. Optional scenario icons and categories help navigation; neither changes document membership or grants repository access.



- **Categories:** when category navigation is shown, choose a category to include its scenarios and those in all its child categories. **All categories** also includes unassigned scenarios; there is no separate Unassigned category in the dashboard. Your administrator can choose a flat catalog instead, hiding category navigation without removing scenarios or their assignments.
- **Search:** search by scenario name or description in the current language. Search narrows the main catalog within the selected category, not repository documents. Clear the search and choose All categories when a scenario is unexpectedly missing. Results retain configured scenario order; use pagination to see more matches.
- **Favorites:** use the favorite control to add or remove a personal shortcut. The favorites area shows your visible favorite scenarios independently of the main catalog's search and category selection. A favorite does not grant repository access or make a hidden scenario visible.
- **Cards view / List view:** choose how to display the catalog. This choice and your favorites are saved privately for your account. Search, selected category and page are temporary; they are not saved dashboard preferences. The administrator's tree/flat choice is shared and is separate from your cards/list choice.

Scenario icons are chosen by the administrator. When no supported icon is configured, the dashboard shows the scenario's initial instead.

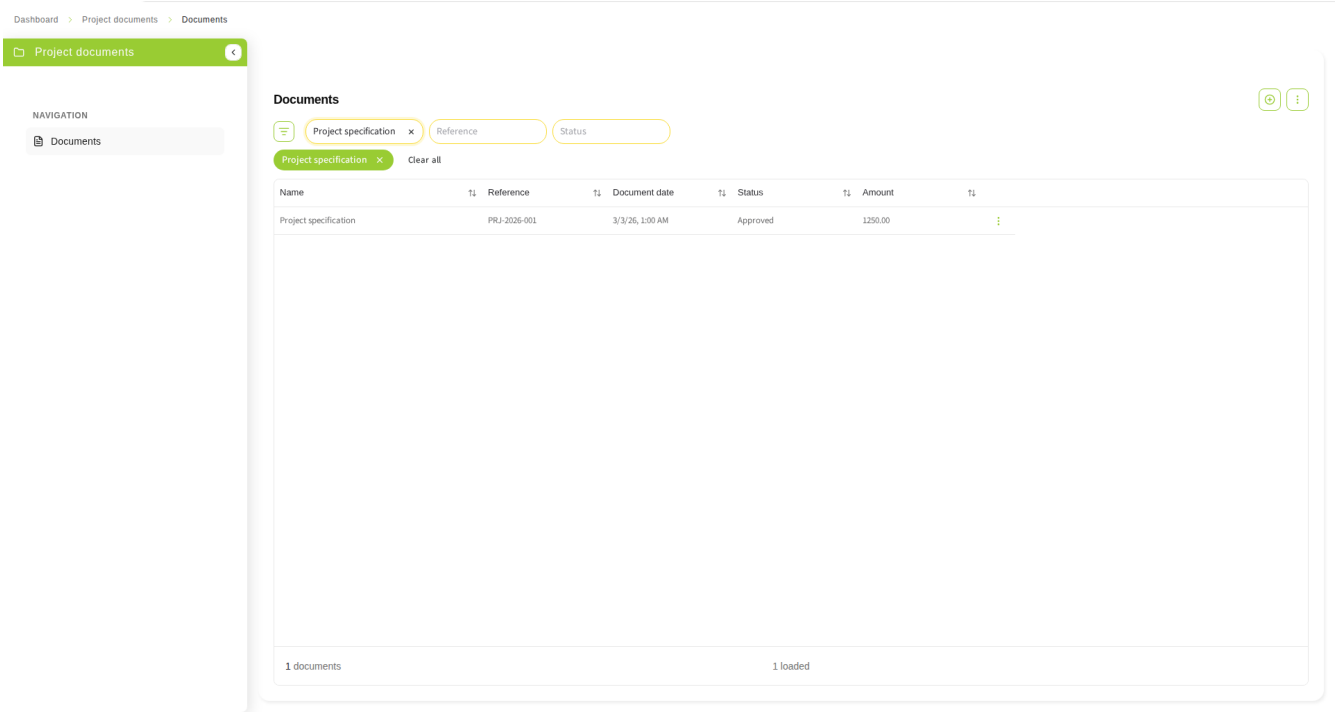
If dashboard preferences cannot be restored, use **Retry loading preferences**. Changes remain pending until restoration succeeds, so they do not replace preferences that could not be loaded. If saving fails, use **Retry saving dashboard preferences**. A warning means the latest choices are not confirmed as saved; it does not prevent opening a scenario.

A one-stage scenario opens a document list. A two-stage scenario first opens a record list; selecting a record opens its documents. The selected record scopes child searches and creation. Confirm the parent before modifying documents.

2.3. Work in a Simple Document scenario

A Simple Document scenario opens one list directly. There is no parent-selection step. The list contains the permitted items of the configured type; its displayed title and fields are chosen by your administrator.

1. Open the scenario card from the dashboard.
2. Enter search values to narrow the list. Clear a filter to broaden the results again; use the offered sorting and column settings to arrange the list. More results load as you move through the list.
3. Select an item to open its workspace alongside the list. Details shows metadata; Viewer is offered for document content; History appears when configured.
4. Use the create action to add an item if it is available. Complete required writable fields and select/upload content as requested. Save and verify the new item in the list.
5. Use an item’s edit or delete action where offered. Check the confirmation and persisted result. Closing the workspace returns attention to the list and does not delete the item.



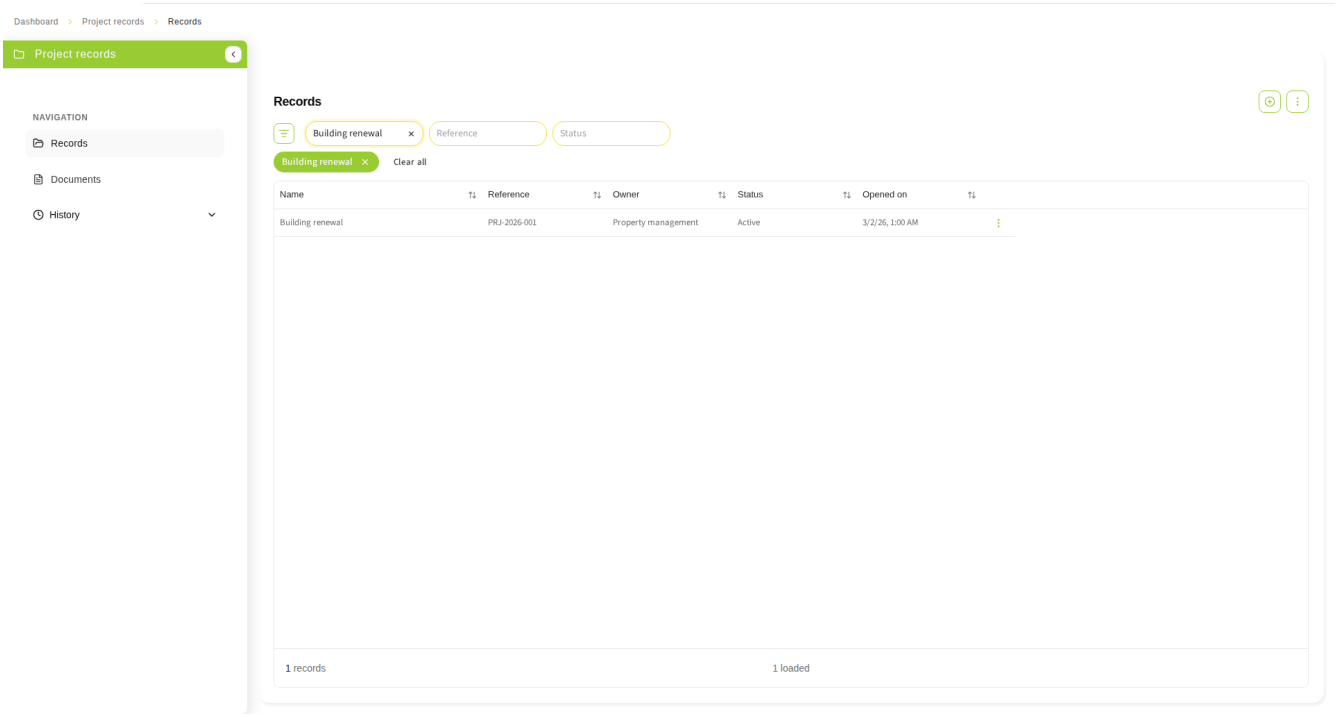
Example: to inspect an invoice, search for its number, select the matching row, check Details, then open Viewer. If a filter hides the saved item, clear that filter before assuming the save failed.

The sidebar shows **Documents** in a Simple Document scenario. When your administrator enables Recycle Bin and grants access, **Recycle Bin** opens trashed documents directly. Record scenarios also provide **Records** and nested Records/Documents destinations in Recycle Bin. Breadcrumbs above the workspace show the scenario and current destination and let you return to an ancestor. Table headings distinguish **Documents**, **Records**, **Trashed Documents** and **Trashed Records**; these labels follow the selected language. When a search includes trashed items, a muted row tint and a narrow status column identify them. The header reserves the same space as each row, so business values stay aligned whether an item is active or trashed.

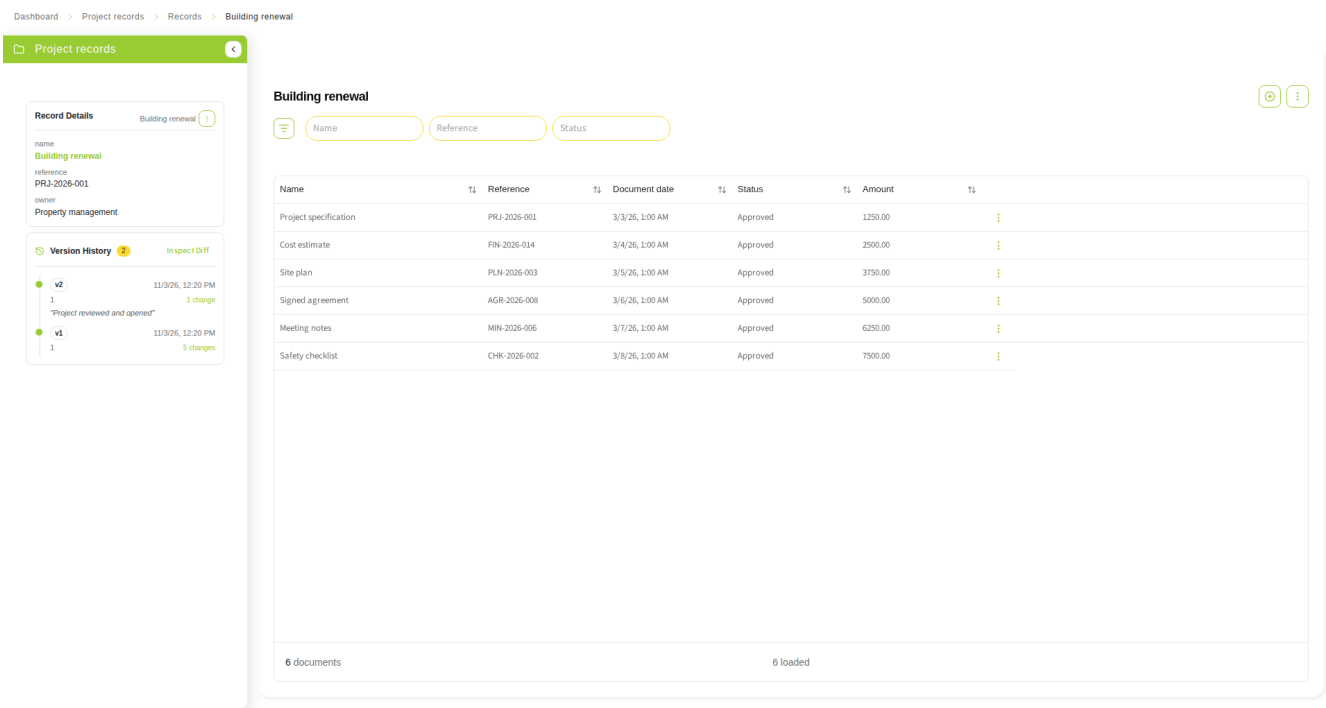
2.4. Work in a Record scenario

A Record scenario first opens the record list. Selecting a record then opens a different list containing its related documents. Search and sorting apply to the stage currently on screen.

In an m:n scenario, the same document can appear under several records with the same identity. The selected record's document filters, counts and pages are evaluated within its membership; the separate flat document view is not evidence that a document belongs to a particular record. Records without links display an empty list. See Share documents between records (m:n) for sharing, creation and removal instructions. Sharing does not copy the document.



1. Search for the parent record, for example a case number or folder name. Check identifying values before selecting it.
2. Select the record to open its documents. Check the breadcrumb and configured parent summary to confirm the current context. The breadcrumb uses the highlighted summary field when it is populated and visible to you; otherwise it uses the permitted record label or identifier. An empty child list can mean that the record has no matching documents or that your search/access excludes them.
3. Search within the selected record's documents. Clearing the child search broadens results within that record; it does not turn the list into a search across all records.
4. Select a document to open Details, Viewer and configured History, just as in a Simple Document scenario.
5. To create a child document, first choose the correct parent and use the child list's create action. The relationship to the current parent is controlled by the scenario; it is not an arbitrary editable grouping field.
6. Close the document workspace when finished. Use the breadcrumb to return to the record list, then select a different record. Closing a document workspace alone does not change its parent context.



Record labels show only currently permitted metadata, falling back to the record ID. Recent-record history uses record IDs so saved descriptive labels cannot reveal fields that are now hidden.

Parent record actions and child document actions are separate. The record summary card's three-dot menu offers Edit, Delete and Trash for active records, or Restore and permanent Delete for trashed records, according to scenario settings and permissions. It uses the same confirmation workflows as the record list. Use the record list's actions to create, edit, inspect history or delete a parent where offered. Use child actions for an individual document. Creating a record does not automatically create its documents. Changing parent metadata does not edit all children.

For a **1:n with foreign key** scenario, before the deletion confirmation opens, the client counts the related documents using record-scoped document read permission. Cascade deletion requires record-scoped document delete permission; unlinking requires record-scoped document edit permission and permission to update the relationship field. All-document workspace grants do not substitute for these permissions. Check the count and whether those documents will be deleted or unlinked. If the count cannot be determined, the client reports an error and does not open the confirmation; retry after resolving the loading or access problem. Record deletion is blocked for m:n scenarios, regardless of the cascade setting.

WARNING

In a one-to-many scenario, deleting a parent can delete all related documents when cascade deletion is enabled. Otherwise children are unlinked: their parent relationship is cleared, while their metadata and stored content are retained. They no longer appear under the deleted record. A mandatory foreign-key relationship requires cascade deletion. Read the confirmation and ask your administrator when the affected scope is unclear.

2.4.1. Share documents between records (m:n)

An m:n membership associates an existing document with a record. It is not a copy or a move. Editing shared metadata or content changes the same document wherever it is linked.

1. Open the document from a selected record's document list or the scenario's flat document view.
2. Open **Details**, then **Edit document**. The picker requires enabled document editing and the document-edit permission for the active workspace. Record-scoped grants and all-document grants are independent.
3. Search in **Add document to record** and check the identifying values of the target record. The picker searches authorized records using administrator-configured fields, independently of authored table columns and personal column preferences; record-read and field visibility permissions still restrict those values. An empty result is not proof that a record does not exist or is accessible to you.
4. Choose one record. Already-linked records are marked and cannot be selected again. Selection is staged: searching, choosing, clearing an unsaved choice or closing the drawer does not add or remove a membership.
5. Choose **Save**. Verify the document in the target record's list and in its previous record's list. Existing memberships are preserved; repeating a confirmed addition does not create another membership. A membership-only save does not create a document version.

Example: a document under case A can also be added to case B without copying its content. An edit from either case is visible from the other. Adding it to B does not remove it from A.

To create a document, first open the intended record's document list and use its create action. The new document is initially linked to that record. Creation requires the configured creation setting and both document-create and document-edit permissions, but can remain enabled when editing existing documents is disabled. Retrying a failed initial link without further metadata changes also works in this create-only configuration. Adding an additional record requires enabled document editing. Creation is disabled in the m:n flat document view because no current record is selected. Repository authorization still controls every write.

To remove a membership:

1. Open the record whose membership you intend to remove and confirm the breadcrumb.
2. Use the document row's **Remove from this record** action, which requires document-edit permission independently of the document editing switch.
3. Check the confirmation. Cancel changes nothing; confirm removes only this record-document membership.
4. Check the refreshed list. The document and its other record memberships are preserved. Removing its final membership leaves it unassigned rather than deleting it.

IMPORTANT	**Delete document everywhere** is a separate disabled action. Record deletion is also blocked in m:n scenarios with cascade both ON and OFF. A verified repository-wide relationship cleanup contract is required before these workflows can be enabled. Do not reset data or apply one-to-many deletion instructions to bypass this safeguard.

If a membership read fails, use the editor's **Retry** action and wait for a successful load before adding a record; the pending choice is preserved. A failed search also preserves your choice. A partial-save error keeps the drawer open: completed document changes and the confirmed identity

remain saved, while the pending addition is retained for retry. If creation succeeded but its initial link failed, retry completes the link using the same identity. If creation could not be confirmed, check the repository before creating again. A failed or uncertain removal is not reported as success; check the refreshed membership before retrying.

2.5. Find and inspect documents

Use the search fields offered by the scenario, choose a supported operator and enter a value of the displayed type. For example, search for a known document name, then clear the search to return to the complete permitted list. Sort using the available table headings; use paging to move through results. Configuration determines available columns, order and searchable fields.

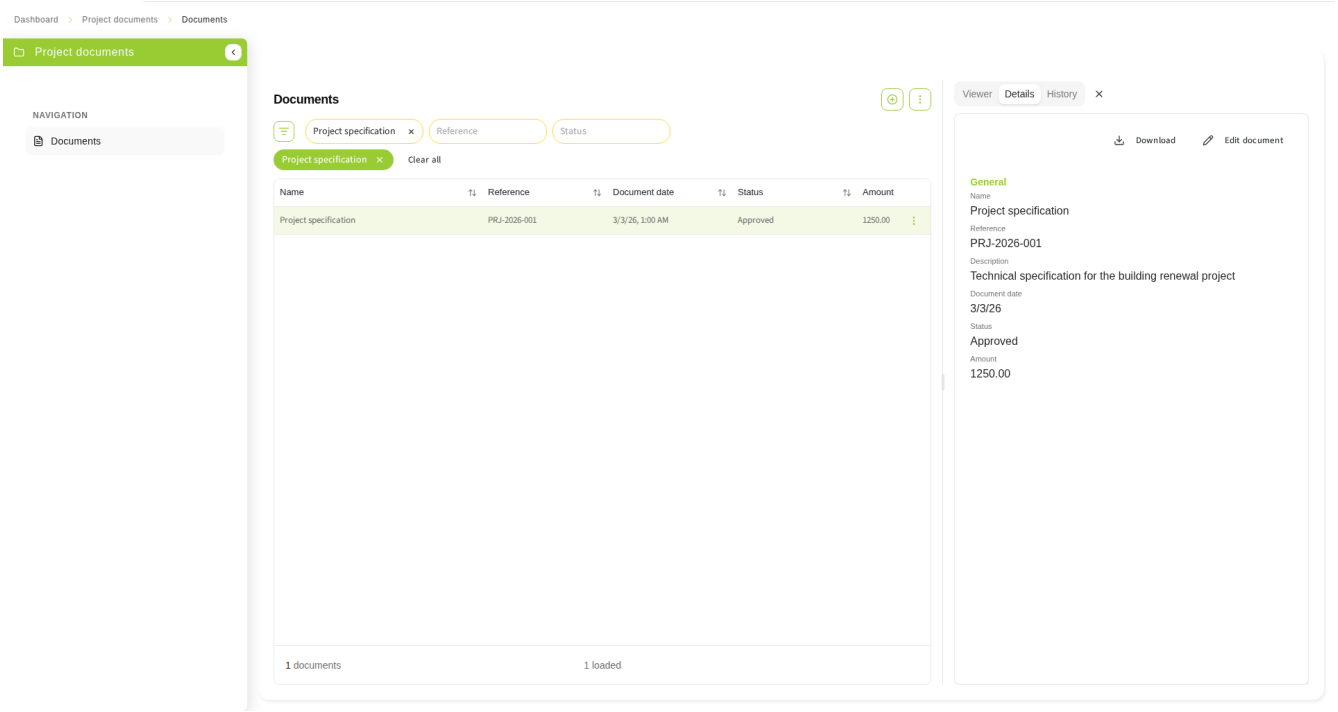
When a field uses an administrator-configured lookup, select a permitted repository value from its dropdown instead of entering arbitrary text. Labels can combine several columns with ` — `; the filter uses the stored key, never the displayed label. Depending on its settings, the dropdown allows one or several selections and optional case-insensitive search. Multiple stored keys match with OR within that field; other active fields still narrow the query. When search is enabled, start typing and continue in the same input as the dropdown opens; opening it does not move focus to **Retry**. Typing searches the options without applying a document filter until a value is selected. Empty lookup search filters show the field name inside the input, matching the other search fields, rather than as a separate label above it. Multiple-selection dropdowns show checkboxes beside the matches and stay open when a value is selected or deselected. Click a row or use the arrow keys and Enter to toggle a value; press Escape or click outside to close the dropdown. While a selected key is being validated, the dropdown stays open and further checkbox changes are temporarily unavailable; continue selecting once validation finishes. Checkboxes reflect confirmed selections only. Changing visible matches preserves selections on other pages. Scroll through the matches to browse more records. Single- and multiple-selection dropdowns automatically load the next page when you reach the final 10% of the scroll range, without a **Load more records** button. If loading fails, use **Retry** to continue. Active lookup filters show the latest selected match in that input without extra search captions, repeated labels or selection-removal buttons. The green filter chips use the same configured display columns as the dropdown, including for restored selections; unresolved values fall back to their stored key. Use a chip’s × button to remove only that selected key: other selections and active fields remain applied, and the dropdown reflects the remaining selections. Removing the last key clears that lookup filter; **Clear all** removes all filters. Filtering still uses exact stored keys, not chip labels, even when several selections have the same label.

Select lookup values individually, including across pages; lookup-filter **Select all** is deferred to AFC-134 and is not available. Press **Escape** to close the dropdown and cancel its search. The search text clears, and reopening starts with unfiltered values. Committed selections and active filter chips remain. Replace the input text to start another query, or erase it to browse unfiltered values; editing the search text alone does not remove active filter chips.

Restored lookup selections resolve separately from dropdown search and paging. A value outside the visible page is not cleared. Unavailable or ambiguous saved selections and retry actions remain visible beside the lookup input when the dropdown is closed; the stored keys remain in the active chips. If the administrator changes a mapping or disables its filter, incompatible saved filter preferences are discarded before querying. If its source type disappears or is no longer a supported document/container type, the lookup control is omitted until the configuration can be repaired.

Lookup browsing and saved-value resolution also require access to a configured source scenario and permission to read its records or documents. All configured label, stored-value and dropdown search columns must be visible in that source's Details and permitted for you. If access changes or a required column is hidden, the lookup cannot use another scenario to bypass that restriction in the matching active source. Keep existing selections and ask an administrator to check the source scenario and field visibility; an unavailable lookup is not proof that a saved value was deleted.

Select a row to open the document workspace. Choose **Details** to inspect metadata. Read-only fields cannot be changed and retain their stored values when other fields are saved. Dates, enumerations and references use their corresponding controls rather than arbitrary text. Not every configured field has a control in every view. There are no column-header filter controls or table CSV/ZIP export actions in this version.



2.6. Create, edit and delete

1. Choose the create action when enabled. In a two-stage scenario, first choose the parent record.
2. Complete all required writable fields. Select valid enumeration/reference values. Supply content when required by the chosen document type.
3. Submit the form. Resolve validation errors next to their fields; preserve the draft until the outcome is clear.
4. Confirm that the saved document appears in the correct list and reopen its details to check persisted values.

To edit, open the existing item and use its edit action. Change writable fields and save. Metadata-only edits leave unchanged content intact. When replacing original content, check filename and MIME information before saving. A failed save is not proof that nothing was stored: reload the item before retrying.

Configured metadata lookups show a single-selection dropdown. Choose the displayed item; its

stored key, not its label, is saved. A saved value outside the current page or dropdown search is resolved independently. Optional editable fields offer **Clear selection**; required fields still need a value, and read-only or permission-restricted fields cannot be changed through the dropdown.

Wait while selections load. Empty results do not mean that an existing value should be erased. If loading fails, keep the draft and use **Retry**. A confirmed unavailable selection remains visible: depending on the administrator's policy, it is preserved or an editable field requires replacement before saving. An ambiguous/invalid selection is not resolved by choosing the first matching row. A temporary service failure is shown separately and does not silently clear data or turn into a removal.

In a strict one-to-many scenario, an administrator may enable choosing another parent. Creation starts with the selected record's referenced key, which need not be its ID. If the parent field is editable and you have the required grant, select another parent and save. After a successful move, the original record's document list refreshes and the moved document disappears from that list. Creation under another parent likewise does not add a row to the original parent's list. A failed save retains your draft parent selection for retry; it does not change the persisted relationship. When the setting is disabled or the field is read-only, keep the existing parent. Many-to-many scenarios continue using their separate membership controls. Moving a held document also requires any additional permission configured for editing under litigation; enabling parent selection does not bypass that protection.

Trash and permanent deletion are different actions. Where the scenario enables the recycle bin, use **Trash** to retain an item for restoration. Selected documents can be trashed or restored together. For a record action, confirmation shows related items and, when configured, asks whether to include children. Restoring included children restores all trashed children, even those trashed earlier. Restoring a single document may first require restoring its parent; declining that parent restoration cancels the document action. Native repository cascades are automatic. A success notification confirms completed trash and restore actions. Partial or unconfirmed operations do not show a success notification; review the results and refresh the list before retrying.

Permanent deletion still requires its own confirmation and cannot be undone. For the existing active-record deletion workflow, one-to-many cascade deletion removes children; disabling it unlinks children, and mandatory foreign keys require cascade behavior. In m:n scenarios, use **Remove from this record** for membership removal. The recycle-bin feature remains in development; see the administration chapter's development notice before relying on it. Cancel if scope is unclear.

Deleting an active record follows the same cascade/unlink policy with Recycle Bin on or off. The count includes all linked documents. If any child is trashed, restore it before choosing an unlink policy. Deleting children permanently needs the corresponding active-delete or trash-purge permissions. Individual document deletion from a record also works with Recycle Bin on or off and remains limited to that record. If an item changes between active and trashed during permanent deletion, execution stops; refresh and confirm the current state before trying again.

Permanently deleting a trashed record always deletes its trashed children first; active linked children must be moved to the recycle bin beforehand. Restore, Delete and Trash are separate actions. Restoring a parent through a native cascade can restore other trashed siblings too, including documents trashed independently. Read the total impact before confirming.

For bin deletion, enter the displayed name exactly, or DELETE for bulk/unnamed items. Restore needs no deletion phrase. Cancelling makes no changes. A failed or interrupted action may have completed some steps: refresh the list and inspect the items before starting again. Repository retention restrictions still apply.

2.7. Choose content and a viewer

Open Viewer in the document workspace. The client chooses the viewer from the selected content's format, not from the scenario name. The same scenario can contain PDFs, images and text documents.

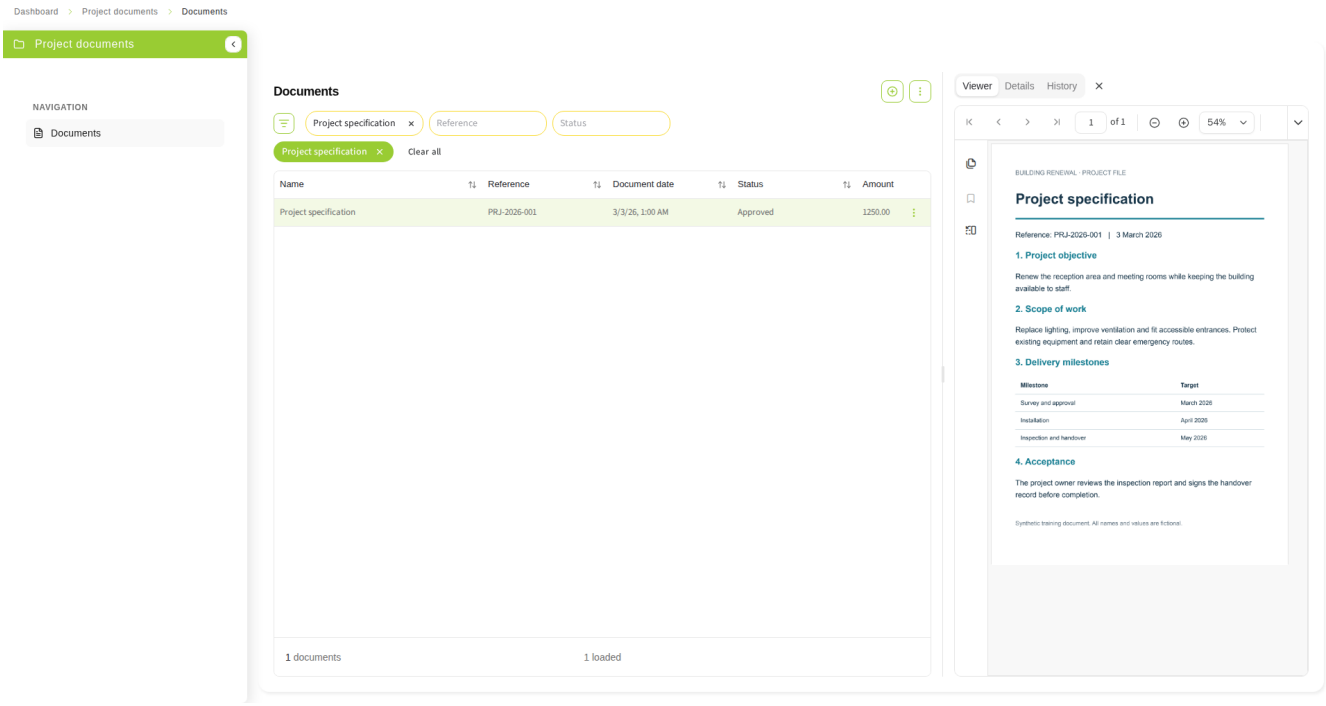
An **original** is a stored content element of the document. A **rendition** is another representation associated with an original, for example a PDF generated from another format. Renditions may be unavailable until repository processing finishes. They are not additional documents or historical versions.

When several configured elements are available, use the content selector above the viewer. Entries are grouped under their originals. The default element is chosen by the administrator and can be an original or a rendition. If only one element is available, there may be no selector. Check the selected entry before viewing or downloading.

Viewer	Content	Main features
PDF	PDF documents	Page navigation, magnification, pan, selection/search, print, download and configured annotations
Image	PNG, JPEG, GIF, BMP, SVG, WebP, AVIF	Zoom, fit/original-size display, rotation, fullscreen and download
TIFF	TIFF content, including multipage files	Page navigation/thumbnails, zoom, rotation and download
Text	Content identified as `text/*`	Line numbers, wrapping, text zoom, copy and download
Fallback	Other formats	Download when enabled; no dedicated inline viewer

There is no dedicated native Office, audio or video viewer in this client. A supported rendition may make such content viewable, for example a PDF rendition. HTML identified as text is shown as text rather than as an interactive webpage. Viewer display controls do not change stored content.

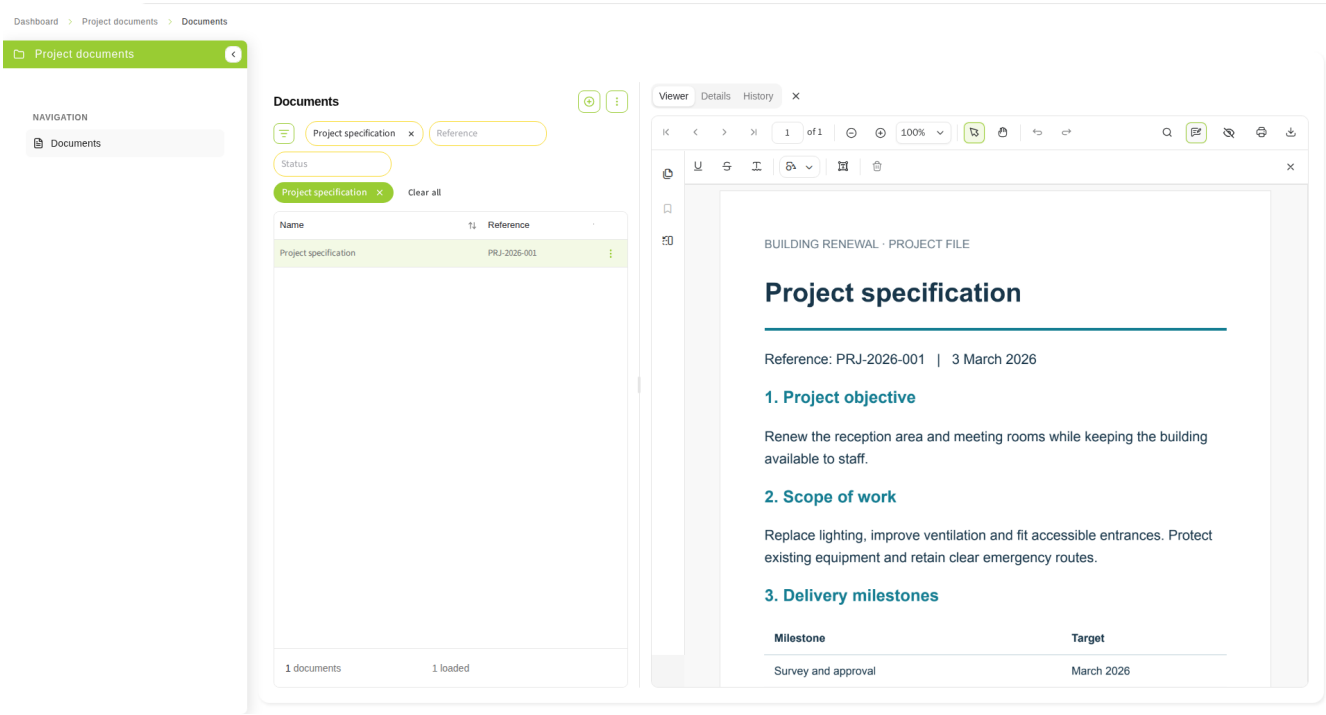
2.7.1. Read a PDF



- **Page navigation:** use previous/next and the page-number control to move through the document. Check the displayed page count; document pages are distinct from pages of search results.
- **Magnification:** use zoom controls or the offered fit modes to make text readable. A narrow workspace may hide toolbar items in an overflow menu; widen the workspace when needed.
- **Pan and selection:** choose Pan to move around an enlarged page, or Selection to select text. These modes affect mouse interaction, not repository metadata.
- **Search:** use the viewer's search tool to find text inside this PDF. This is separate from scenario search. Image-only scans may contain no searchable/selectable text; the client does not perform OCR on demand.
- **Print:** use the PDF print action when available. Browser printing and device selection occur outside the scenario configuration workflow.
- **Download:** use the download action when enabled for the selected content. Check the filename and browser download destination.

2.7.2. View, edit and save PDF annotations

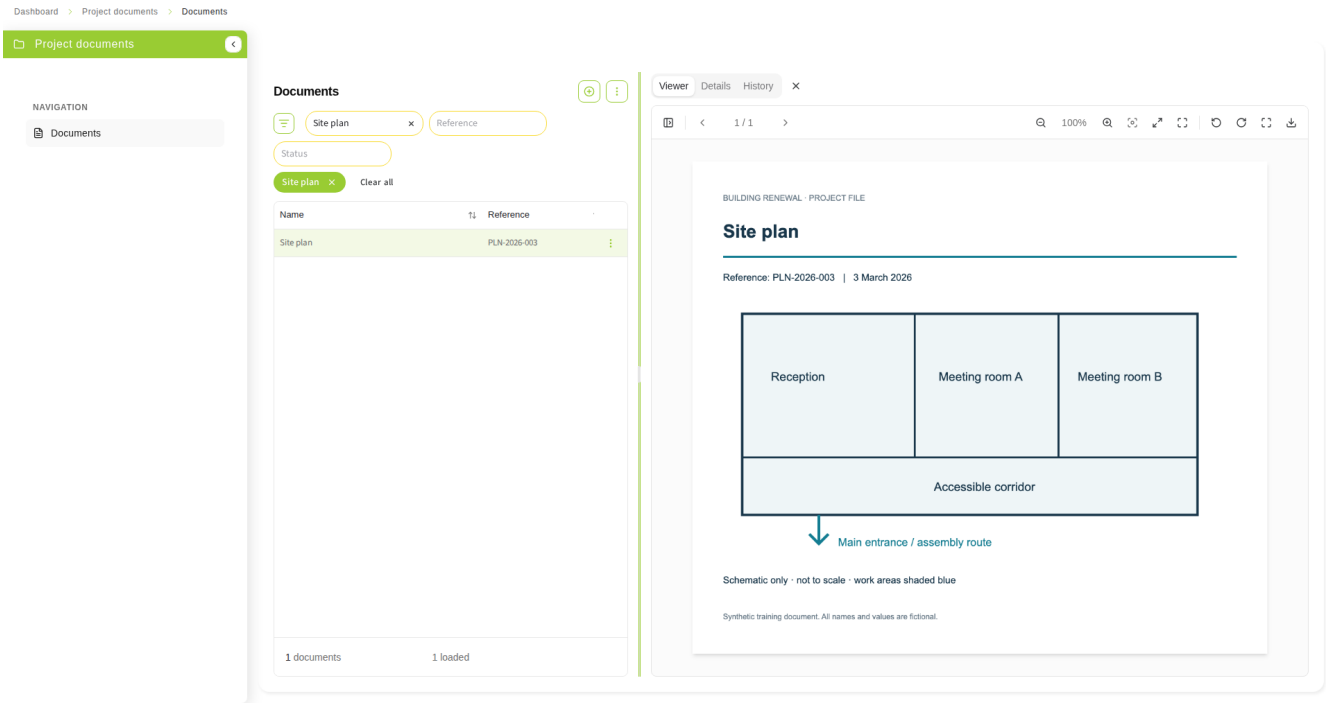
Annotations require enabled configuration and repository permissions. The viewer can show saved annotations without allowing you to edit them. Historical versions are read-only. Annotation settings belong to an original and also apply to its renditions.



1. Use Show/Hide annotations to control their visibility. Hiding them is a display choice, not deletion. Editing tools are available only when annotations are visible and editable.
2. Enter annotation editing through the toolbar. Text-markup tools include highlight, underline, strikethrough and squiggly underline. They require selectable text. Free-text and shape tools can be used for visible notes and markings.
3. Select an annotation to change the offered fill/stroke color, thickness or opacity, move/resize it where supported, or delete it. Undo/redo operates on editing changes; it is not repository version restoration.
4. Choose Save annotations to persist changes. Until saving succeeds, the draft is not proof of a stored annotation. Reopen the content to check the result when necessary.
5. When changing content or leaving the document with unsaved annotation changes, use the offered Save and continue, Discard and continue, or Cancel choice. Cancel keeps the current document and draft open.

If annotations cannot load, use Retry and wait for a successful load before editing. A save failure requires resolving the error and retrying; do not discard a needed draft just to dismiss the message. Permission restrictions or loading errors can disable editing even while the PDF itself is visible.

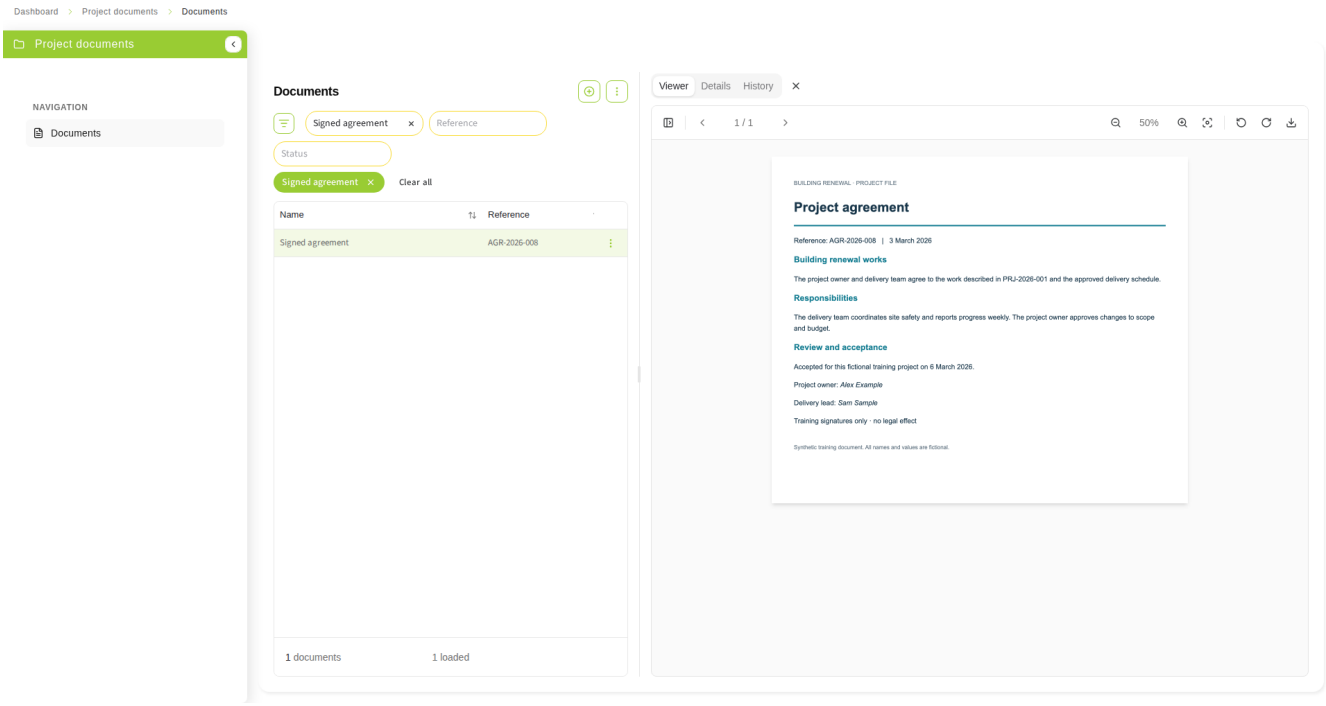
2.7.3. Inspect an image



Use zoom in/out to change magnification and reset to return to the base zoom. Fit adapts the image to the available space; original-size display uses its natural dimensions. Rotate clockwise or counterclockwise to read sideways scans; these rotations affect the display only. Use fullscreen for more space and exit fullscreen to return to the workspace. Download obtains the selected stored content when enabled, rather than a screenshot of your rotated view.

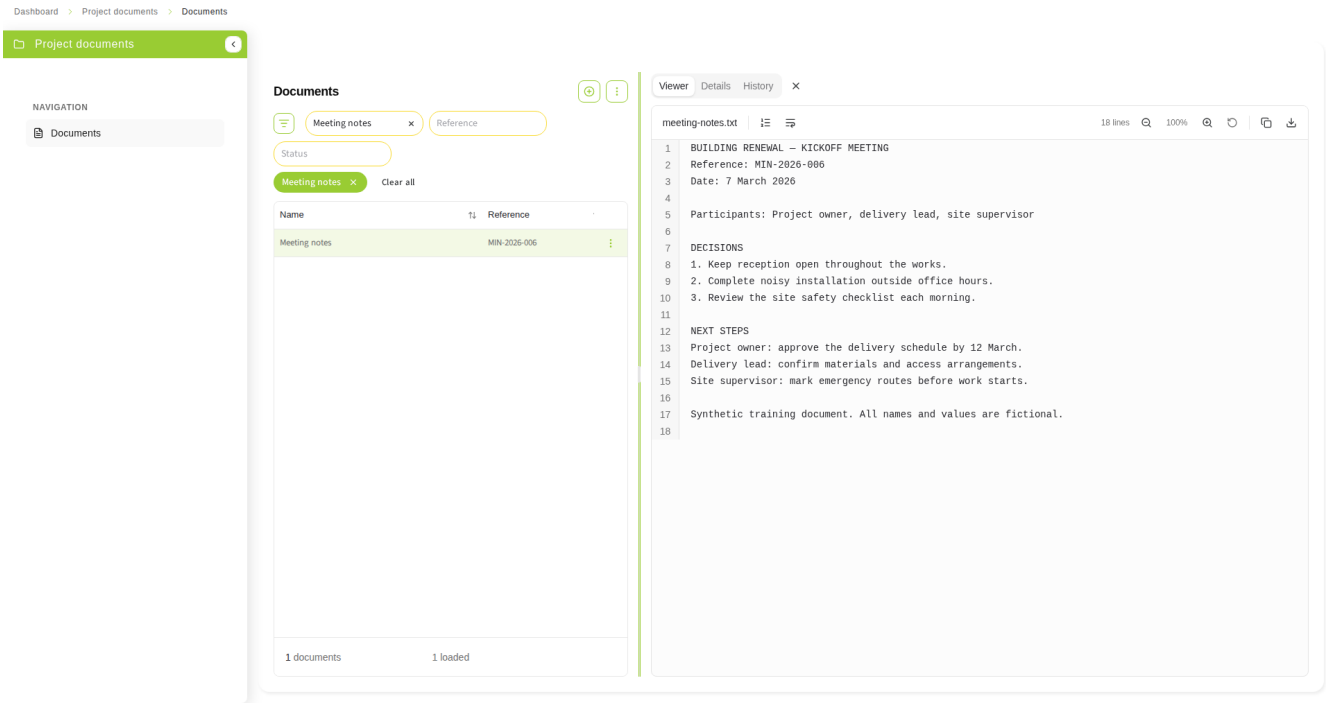
The viewer component can navigate multiple supplied images, but this client supplies the selected content as one image. Use the document content selector to switch between stored originals/renditions; do not expect an image gallery of all documents. Image formats do not expose the PDF annotation toolbar.

2.7.4. Read a TIFF



TIFF files can contain several pages. Use previous/next or the page-number control to change the active page; toggle the thumbnail sidebar to identify pages visually. A single-page TIFF has no additional page to navigate to. Zoom in/out and reset adjust the active page's display; rotation helps read sideways pages. Download, when enabled, retrieves the TIFF content, not merely the rendered page. PDF text search and annotation editing are not TIFF controls.

2.7.5. Read text content



Toggle line numbers when referring to a particular line. Enable wrapping to keep long lines within the viewer; disable it to preserve long-line layout and scroll horizontally. Text zoom changes displayed font size, and reset restores the base size. Copy text copies the content to the clipboard when the browser permits it; download saves the selected content when enabled. These actions do not edit the stored text. This viewer is a reader, not a source editor or a rendered HTML preview.

2.7.6. Downloads, unavailable content and historical viewing

Viewing and downloading are separate capabilities. Each original and rendition has its own availability and download settings; repository permissions still apply. A hidden download control does not mean the document has no content. An unsupported format offers a download fallback only when download is enabled.

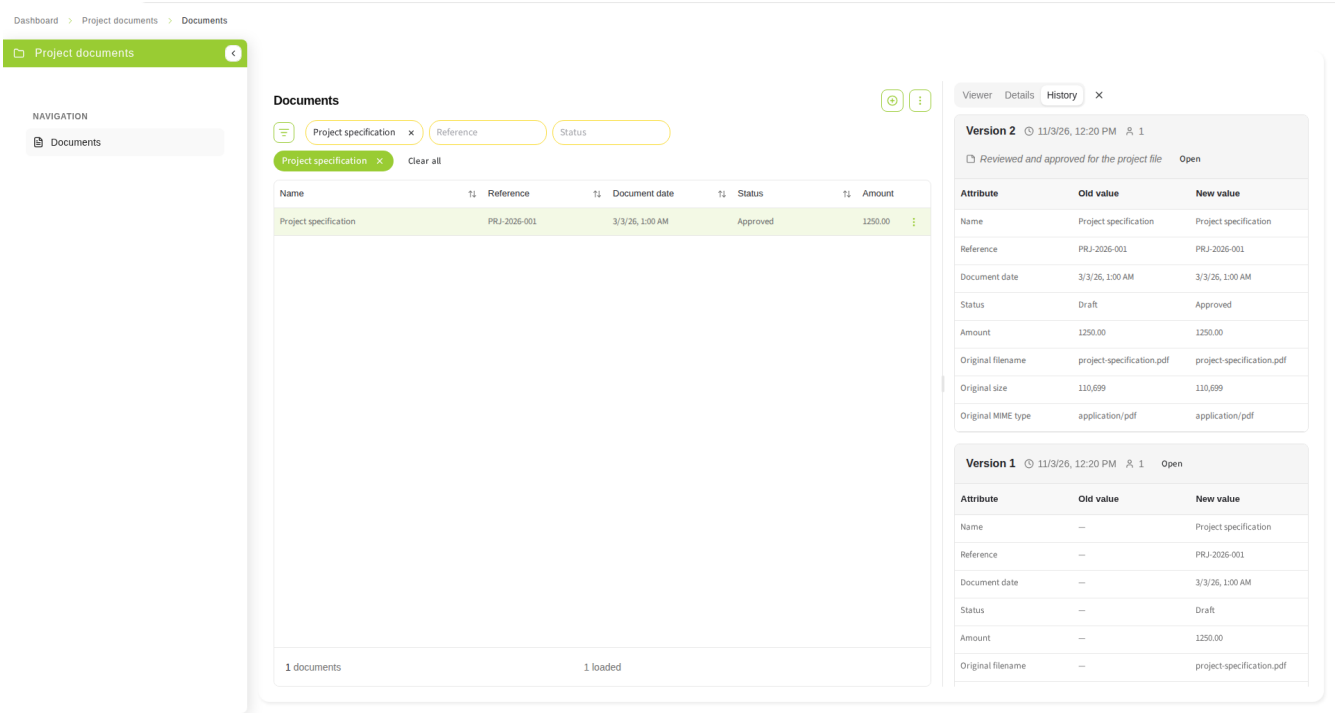
Loading errors can result from inaccessible content, invalid format metadata, missing renditions or viewer initialization failures. Check the content selector, try an available original, and reload through Shell if needed. An empty preview alone does not establish an authorization failure. Report the scenario, document and selected element to your administrator without including credentials.

Content and annotations may be unavailable while a document is in the recycle bin, including content and annotations of historical versions. Retention does not guarantee preview or download access while trashed; a failed read does not prove that content was deleted. If access is needed, use the authorized Restore workflow, review its related-item impact before confirming, then reopen the content after restoration. Ask your administrator if restoration is unavailable. See Recycle-bin configuration (in development) for the current limitation.

From History, open a historical version to inspect its content and version fields in a separate drawer. Historical content uses its own content selector. Close that drawer to return to the current version. Viewing an old version does not restore it; annotations of historical content cannot be edited.

2.8. Versions and comments

When history is enabled, open **History**. Select a historical version to inspect that version rather than the current document. Return to the current version before editing. Comments appear only where enabled and permitted; cancel an unfinished comment to discard it, or submit it and reopen history to check the saved result.



Historical selection is an inspection action; it does not restore or overwrite the current version. Record and document history are independently configured.

2.9. Recover from errors and unfinished work

- Validation failure: correct the indicated fields and submit again.
- List/loading failure: retry or reload through Shell; verify the selected scenario and parent record.
- Permission denial: contact the administrator with the action and item involved, without sending credentials.
- Session expiry: authenticate through Shell again, then check persisted state before repeating a write.
- Navigation with unsaved work: save explicitly or intentionally discard your draft. Configuration and PDF annotation editing provide their documented save/discard/cancel choices. Do not assume that every other form draft has the same protection.

Recent records and saved views depend on scenario context and repository access. For removed or inaccessible records, return to the scenario and search again.

When a configuration administrator has no accessible visible scenarios, the dashboard explains that scenario access requires an explicit mapping and offers a configuration shortcut. Administration remains available. Your administrator can browse the permission catalog and

configure access; no permissions are granted automatically.

2.10. Document workspace settings

In Record scenarios, **Documents** lists all scenario documents and a selected record lists only its linked documents. Administrators can configure these workspaces independently, so their columns, filters, details and permitted actions may differ. Each workspace remembers its own enabled table and search preferences. Content and version capabilities share the scenario’s document-type settings.

Chapter 3. Litigation holds

Eligible records and documents display a litigation-hold indicator in a fixed table column, like the recycle-bin indicator. A success notification confirms setting or removing a hold, and the table refreshes automatically. Set hold is available only for items without a hold; for a multiple selection, every selected item must be without a hold. Remove hold is available only for held items; for a multiple selection, every selected item must be held. A hold protects the entity across its versions, including after its retention date expires. Setting or removing a hold requires an enabled scenario feature and the permissions configured by your administrator; the repository may still reject an operation.

Use the row menu, selected-row actions, record card or document details to set or remove a hold. The confirmation shows affected entities and options for connected records or documents. Setting a hold protects against permanent deletion; removing it leaves other retention rules in effect. Review the confirmation before proceeding. Optional inclusion covers visible related documents only. Automatic inclusion is required by scenario configuration. Native repository inheritance is mandatory and can affect documents you cannot see. You cannot override an inherited child hold independently. When removing a hold from such a document, include the connected record in the confirmation. This removes the hold from the record and all documents inheriting it, including documents you cannot view. The record removal permission is required. Without including the record, the removal cannot be confirmed.

The administrator can configure different child-inclusion rules for setting and removing holds. Removing a record hold does not necessarily remove independent document holds. In strict record scenarios, setting a document hold can optionally or automatically include its parent and then visible siblings. Shared documents retain an entity-wide hold without automatic propagation through many-to-many relationships.

If the confirmed selection changes, review the refreshed confirmation and confirm again. After an uncertain failure, refresh and inspect current protection before retrying.

The Litigation navigation entry lists held records and documents when enabled and authorized. Viewing this list does not grant permission to change holds. Held items in the recycle bin require the usual recycle-bin visibility permissions.

Your administrator can require an additional permission to edit held items. This includes metadata, relationships, content, versions and annotations; ordinary edit permissions still apply.

Chapter 4. Retention dates and protected lifecycle actions

Where your administrator enables it, tables and document details show whether retention is unsupported, unset, active or expired. In document details, the retention label and state appear inside the metadata card; hold and retention actions share its action row with Download. The table settings allow you to move the retention column among the business columns; protection indicators remain fixed. Litigation hold is shown separately. Expiration does not itself grant permission to delete. Use **Set retention date** from the entry menu, record card, details or selected-row actions. Choose a day in the date picker, then choose **Review scope**. The current local time is captured when you select the day and retained through confirmation. If that time is skipped or repeated on the selected day due to daylight saving time, select the day again at another time. Review the summary of affected and unchanged entries, the associated records and documents to include, and whether retention can be extended. Existing longer dates remain unchanged. Configured automatic inclusion is mandatory; optional inclusion requires your choice. Configured inclusion happens only during this action, not when creating or linking documents. For native inheritance, explicitly include the source record. The record and all inheriting documents may change, including documents you cannot see. You need the applicable source-record permission. Choose **Set retention date** only after reviewing this effect. Cancellation changes nothing. After an error, your input remains; review again before retrying. An uncertain response must be reconciled by re-reading state. Lookup-backed metadata and parent-record reassignment follow the same editing protection as ordinary fields. A configured parent selector does not grant permission to edit a held document or a document in the recycle bin.

A litigation hold always blocks moving an entry to the recycle bin and permanently deleting or purging it. Active retention requires a separate administrator-configured override in addition to normal action and repository permissions. Retention management permission does not grant this override. Neither deletion nor trash clears a hold or retention date. Extending retention remains possible during a hold with the retention permission.

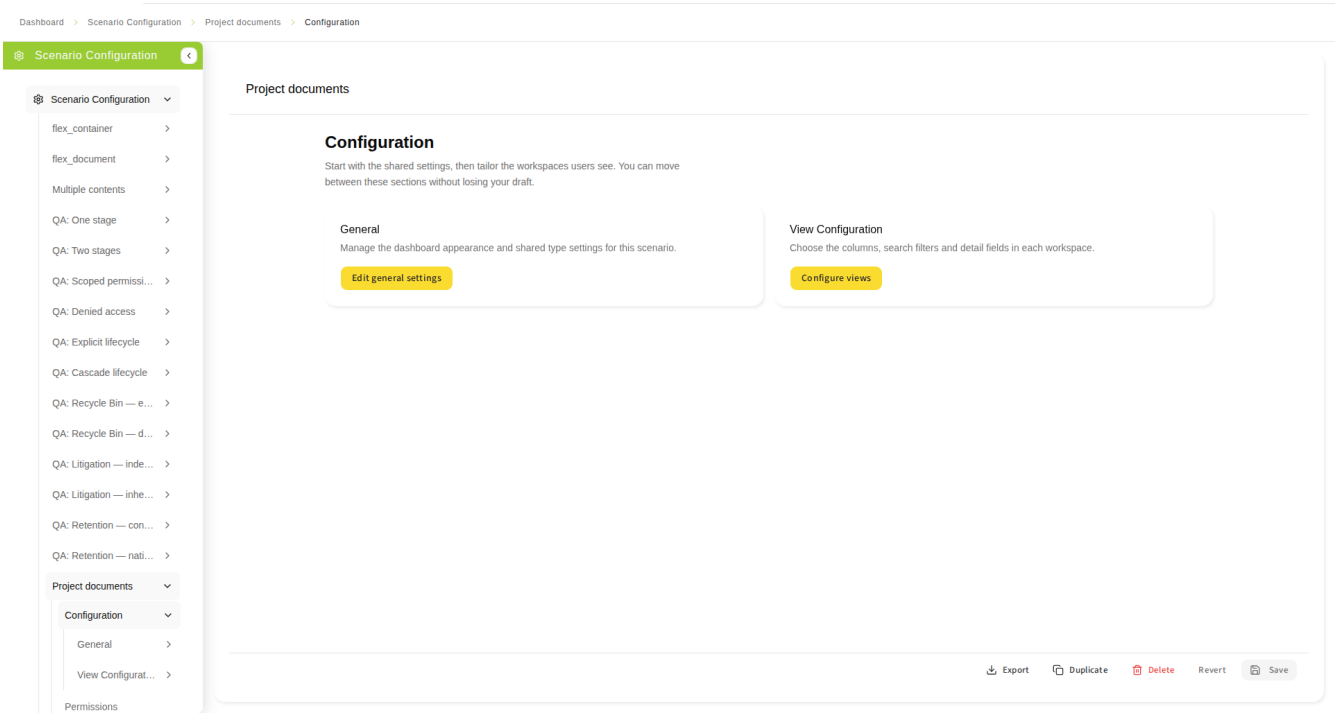
Chapter 5. Administration

5.1. Enter administration

Sign in through Arveo Shell with a configuration-administrator account. Open Flex Client, then **Open settings**. Normal users cannot manage configuration. Repository access and configuration-administrator permissions are distinct; enabled scenario capabilities never override repository authorization.

Use the administration navigation to switch between **Scenario configuration**, **Permission catalog**, **Categories** and **Localization**. Scenario settings, category hierarchy/assignments and dashboard presentation share one configuration draft. Moving between these workspaces retains unsaved changes; it does not save them. Enter through the Shell client and use this navigation rather than bookmarking a nested administration URL.

Each administration page starts with a short explanation. Configuration cards describe their purpose and provide a direct action. Shared type settings distinguish global attribute availability from view-specific visibility; permission pages distinguish configuration defaults from individual rules. Expand **How are these rules applied?** for the rule explanation. The global Permission catalog shows discovery status separately from identity search and usage inspection.



Choose **Scenario Configuration** in the navigation to open the scenario list. Search by scenario name, title or type, filter by stage kind, or use Create, Import and Export all. Select a scenario to open its overview, which shows its description, identity, structure, types and visibility, plus Export, Duplicate and Delete. Use the **Configuration** and **Permissions** navigation cards or sidebar children to open the corresponding editor. The breadcrumb shows Dashboard, Scenario Configuration, the scenario title and the current section; ancestor links navigate back. Clicking the Scenario Configuration row opens the list and toggles its submenu. Clicking a scenario row opens its overview and toggles its children, closing other scenario branches. The chevron is part of the row. Clicking anywhere on a scenario card opens its overview. Cancelling an unsaved-change prompt

preserves the current page and menu expansion. The Configuration sidebar contains **General** and **View Configuration**. Each grouping opens a landing page with links to its children. General contains Overview (naming, icon, dashboard and record management) and TypeDefinitions (record and document attributes, versions and content). One-stage scenarios omit record settings. View Configuration follows runtime navigation: Records contains Record search and Documents within a record; Documents configures direct document search. Select a screen area using its button or the example image to open its settings without deeper navigation. Breadcrumb ancestors return to their landing pages. The JSON preview is read-only; edit using structured controls or validated import. Export a backup before significant changes. Unsaved edits are retained between a scenario's overview, Configuration and Permissions pages. Moving to another scenario or the scenario list asks you to save, revert or cancel when the current scenario is dirty. Import returns to the list, where Save all persists the shared draft; successful duplication opens the copy's overview, and deleting the open scenario returns to the list.

The separate **Categories** workspace manages the category tree, scenario assignments and dashboard presentation. In scenario General settings, choose an optional semantic icon; it changes presentation only, not repository data or permissions.

5.2. Choose a scenario structure

A scenario is a configured entry point into repository data. Choose its structure before configuring its labels and views. The document and record types must already exist in the repository; this wizard does not create types or relationships.

For a Record scenario, the wizard offers **1:n with foreign key** or **m:n with shared documents**. The latter uses a dedicated container-to-document relation, not a copied document or a document foreign key. Only compatible current-version-only relations with a unique endpoint-pair index and no required writable custom relationship attributes are selectable. If no compatible type is offered, ask the repository administrator to check those definitions.

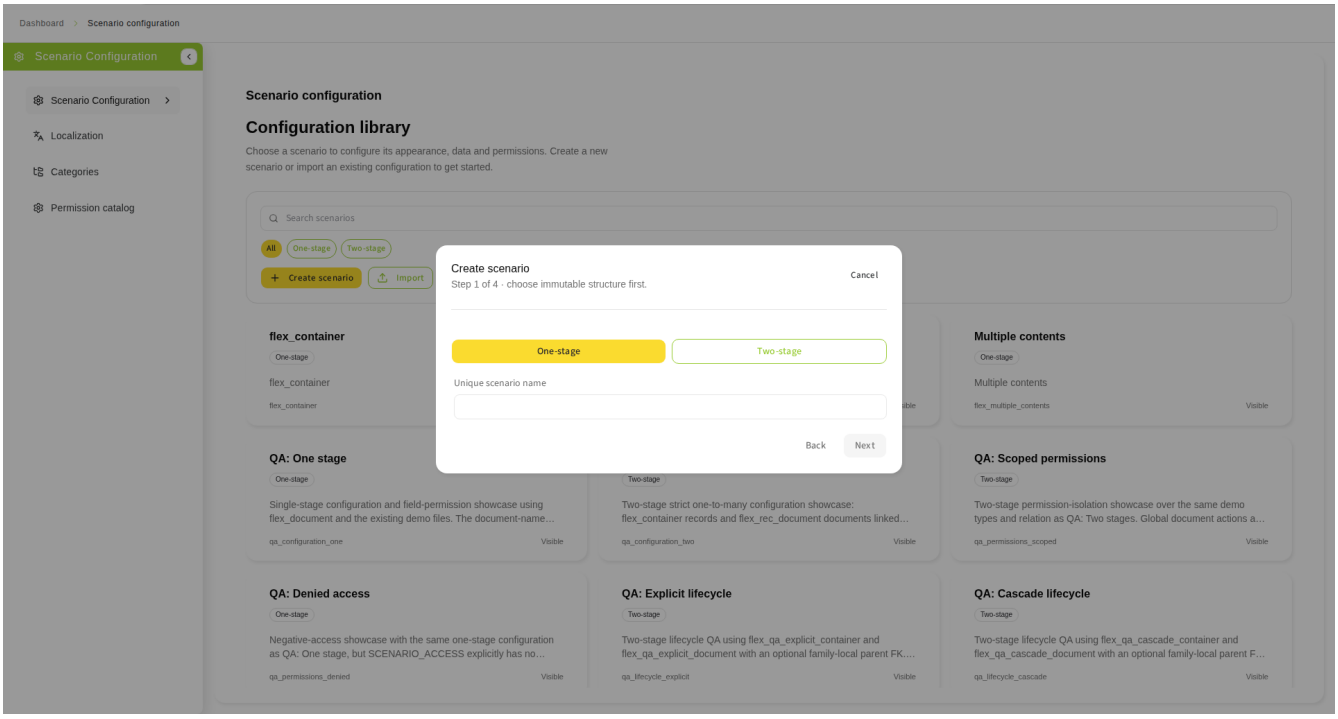
IMPORTANT	m:n authoring, record-scoped browsing, document-edit linking, creation from a selected record and current-record removal are available in configuration version 2. Enable document editing to expose the picker; additions/removal use the existing document-edit grant. Global document deletion and record cascade workflows remain unavailable pending a verified complete repository-wide cleanup contract. Do not treat authoring as a completed destructive lifecycle workflow.

Choice	Simple Document (one-stage)	Record (two-stage)
Opening the scenario	One list of items of the selected document type	A parent record list, followed by documents belonging to the selected record
Type selection	One document type	Child document type plus a relationship that determines the parent record type

Choice	Simple Document (one-stage)	Record (two-stage)
Configuration	Document attributes, views, content and history	Separate record and document attributes, views and history; document content; stored record-deletion policy
Example	A searchable collection of invoices	Case records, each containing its own correspondence

The wizard identifies these choices as **One-stage** and **Two-stage**. This guide calls them Simple Document and Record scenarios. A record is the parent organizing the documents, not another content rendition.

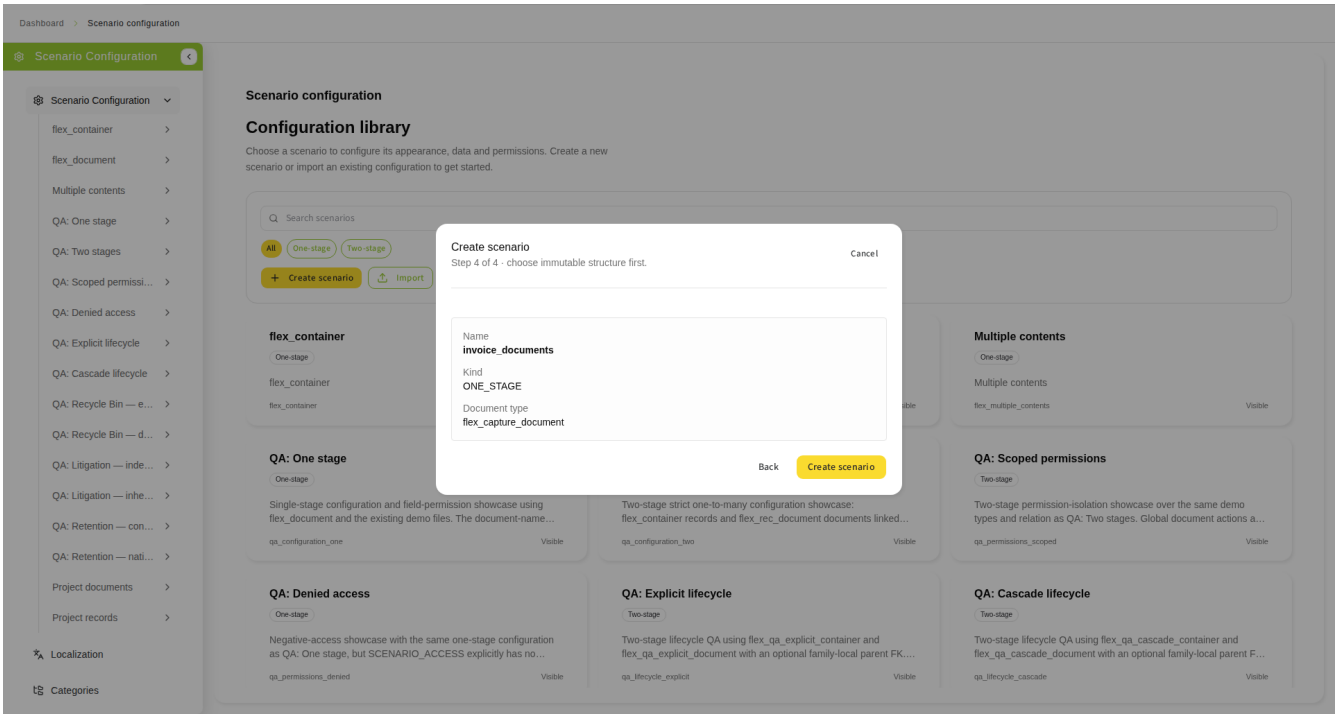
After a new scenario is successfully saved, leaving administration requires no further save unless other configuration or localization edits remain unsaved.



5.2.1. Create a Simple Document scenario

1. **Step 1 - identity:** choose One-stage and enter a unique scenario name. Use a stable descriptive identifier, for example `invoice_documents`. Leading and trailing whitespace is removed. The name must contain 1 to 200 characters; a duplicate name is rejected when creating the scenario. The name is its identity, while the displayed title can be configured later.
2. **Step 2 - document type:** select the repository type containing the items you want to expose. For example, the demo uses `flex_document`. A type choice controls which attributes and content elements are available later. It does not convert existing objects to that type.
3. **Step 3 - defaults explanation:** read the explanation of the proposed document configuration. This step is informational, not the attribute editor. The editor becomes available after creation.
4. **Step 4 - review:** check the name, kind and document type. Choose Create only when these structural choices are correct. Successful creation immediately stores the scenario and opens its

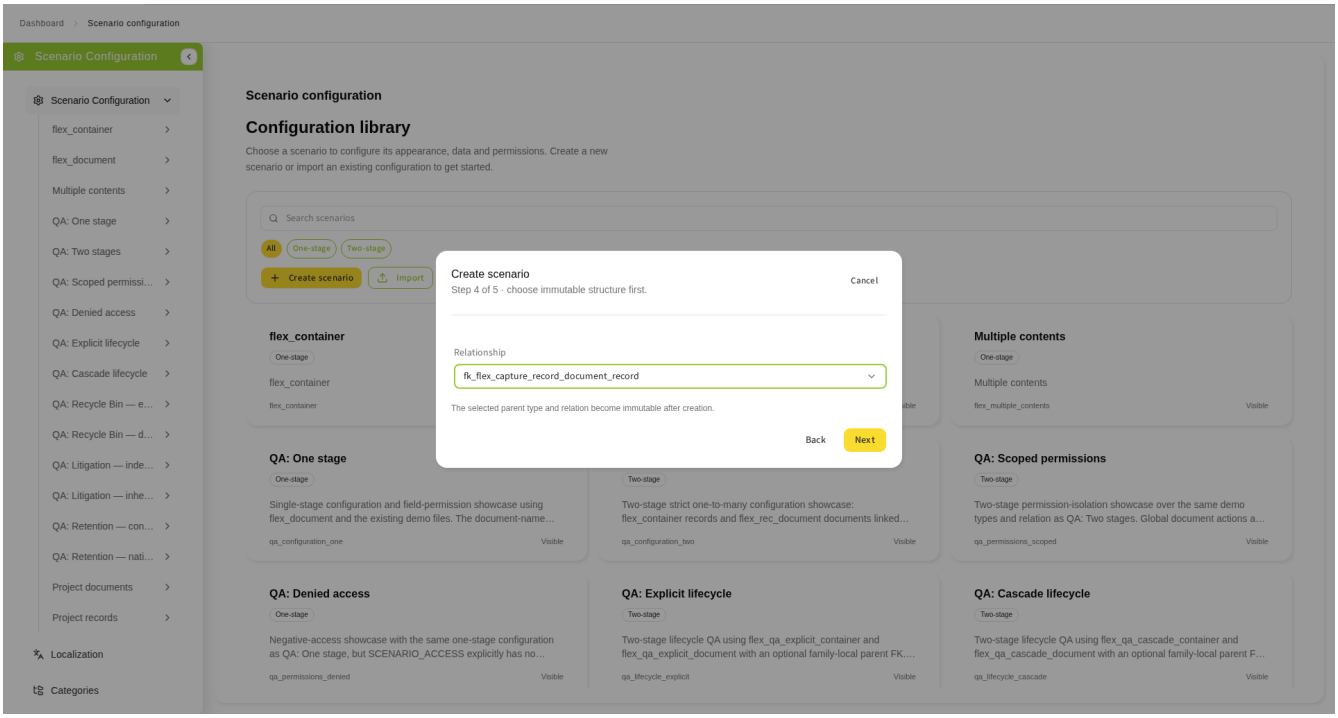
overview; choose Configuration to edit it; later changes require Save.

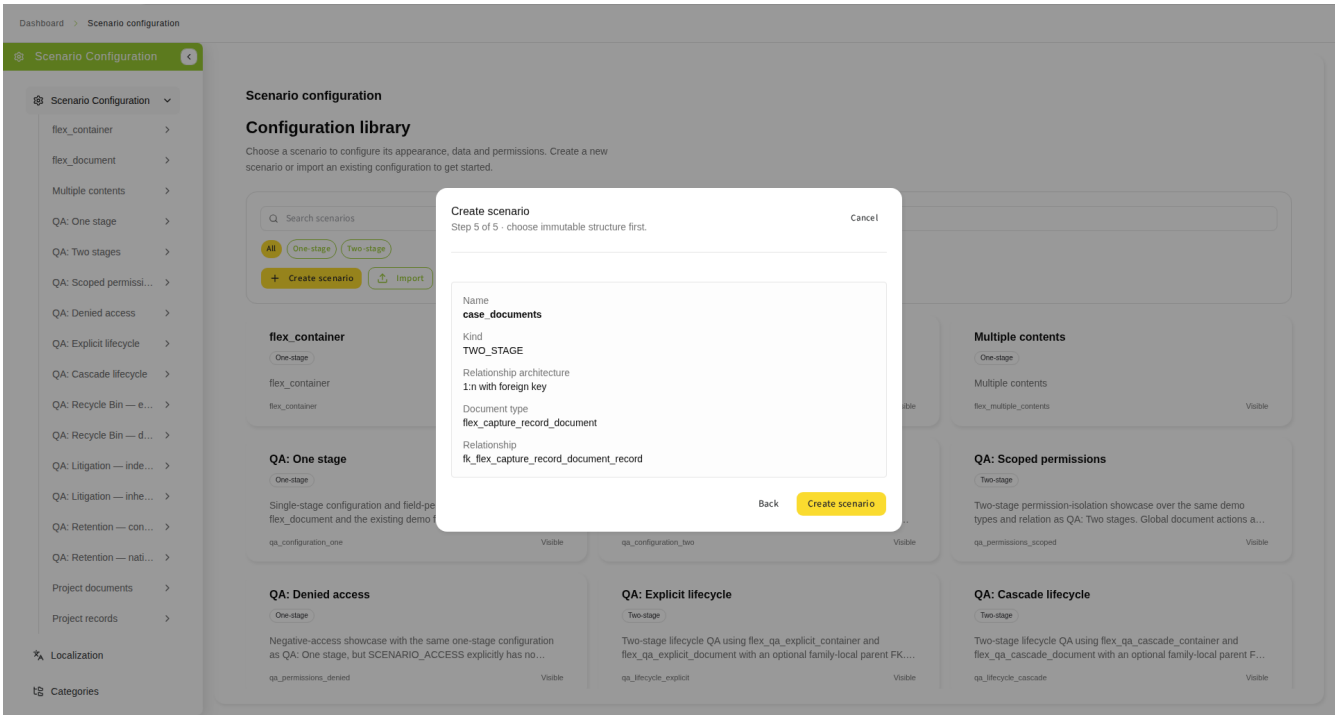


Next is disabled while a required choice is missing. Back returns to an earlier step; after changing a type, check the downstream choices again. Cancel closes the wizard without creating a scenario. A failed Create requires resolving the reported error; check the sidebar before repeating a request after a connection failure.

5.2.2. Create a one-to-many Record scenario

1. **Step 1 - identity:** choose Two-stage and enter a unique stable name, for example `case_documents`.
2. **Step 2 - architecture:** select **1:n with foreign key** (Strict one-to-many) when each document belongs to one parent through a repository foreign key. For shared documents, follow Create and configure an m:n Record scenario in V2 instead.
3. **Step 3 - child document type:** select an eligible document type. The choices are filtered to types with a supported child-to-parent relationship. In the demo, `flex_rec_document` is the child type. An empty list means that an eligible repository relationship is needed; choosing an unrelated type is not a workaround.
4. **Step 4 - relationship:** choose the foreign key linking the child to its parent. Each option shows the relationship name, child attribute and target type/attribute. The target determines the record type; there is no separate arbitrary parent-type choice. A single eligible relationship is selected automatically. When there are several, choose the one that represents the intended grouping.
5. **Step 5 - review:** verify name, kind, architecture, child type and relationship. For example, `fk_flex_rec_document_container` links the demo documents to `flex_container`. Create persists the initial scenario; configure its two stages next.





Going Back can clear dependent architecture, type or relationship selections. Recheck them before advancing. The wizard does not move documents, create parent records or repair repository relationships.

IMPORTANT	Name, kind, types and relationship structure are immutable after creation. To change the structure, create a new scenario. Duplicating also requires a new unique name.

5.2.3. Create and configure an m:n Record scenario in V2

1. Choose Two-stage and enter a unique scenario name.
2. Choose **m:n with shared documents** rather than the foreign-key architecture.

3. Select an eligible child document type and its compatible container-to-document relation. The relation determines the record type; the wizard does not create repository types or memberships.
4. Review the immutable structure and choose Create. A safe discovered text attribute supplies initial picker defaults; if none exists, creation leaves an unsaved draft requiring administrator selection rather than inventing an entity field.
5. Configure the record and document stages, including the independent **Record selection** settings below, then save and reload.

Open General → Overview and expand **Record selection**. Enable the record attributes shown in options and order them with the shared ordering control. Configure searchable record attributes separately, with **equals**, **contains** or **starts with** where compatible with the live type. Text predicates are intended to match case-insensitively and be combined with OR. Numeric/scalar attributes support equality only; unsupported types are rejected.

At least one display column and one compatible search column are required. The picker settings are independent of authored record-table columns, sorting and personal column preferences. Record-read permission and record table-field visibility rules still restrict its identifying and searchable values. Removed or disabled references remain visible as invalid settings for explicit repair; they are not silently dropped. Save, export and reload to verify the authored order and search rules. Use V2 files for these fields: legacy V1/unversioned imports cannot contain them.

Enable document editing for membership additions, and document creation for creation from a selected record. Additions require the existing document-edit grant; creation requires both document-create and document-edit grants, independently of the switch for editing existing documents. A create-only configuration permits initial-link retry without further metadata changes. Current-record removal checks record-scoped document-edit permission independently of the editing switch. Record-scoped create/edit grants are separate from the all-document workspace grants. Repository authorization still applies. Browsing, linking, selected-record creation and unlinking are available; enabling a setting does not grant access or enable global deletion.

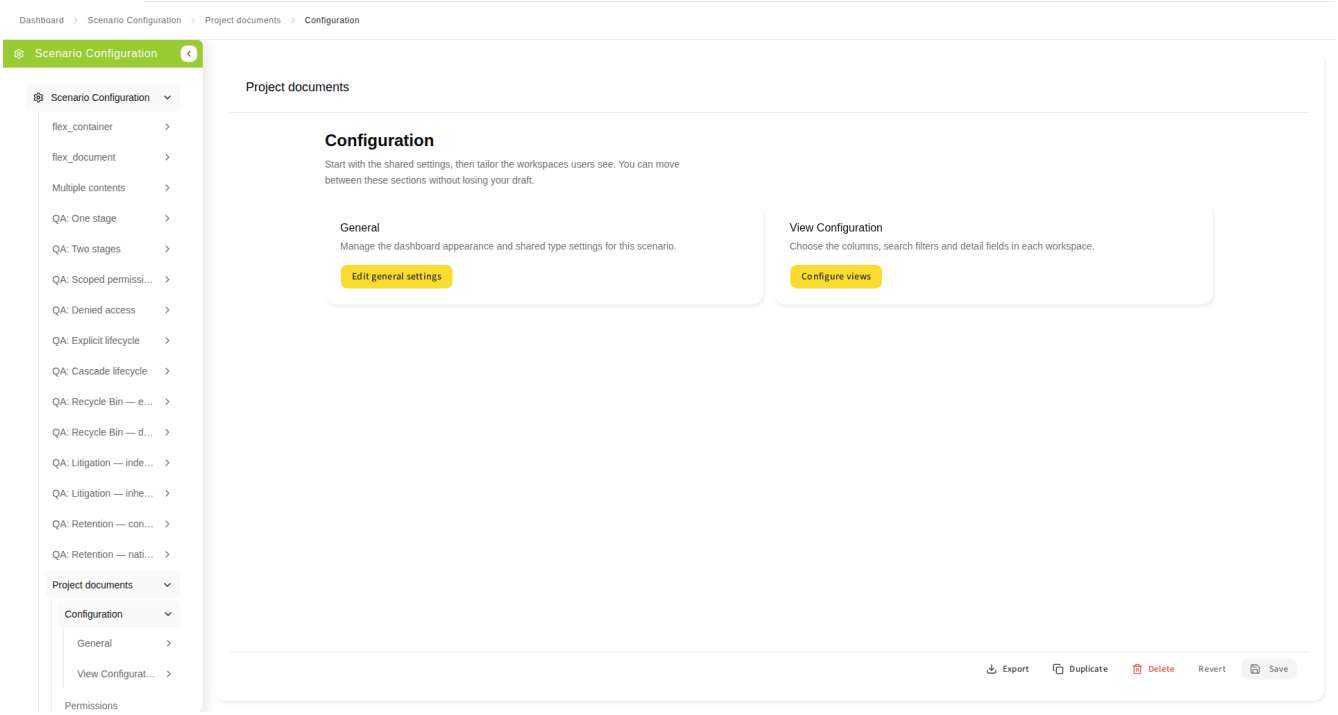
The stored m:n cascade setting records a policy: enabled would delete associated documents globally, including their memberships under other records; disabled would preserve documents and remove memberships before deleting the record. Neither setting enables record deletion in the current runtime. Global document deletion and record deletion remain blocked until complete repository-wide relationship cleanup is verified; a scenario's configured relation alone cannot establish that guarantee.

Verify with authorized test data: open records A and B, add an existing document from A to B through the document editor and check both lists. A membership-only save must not create a document version. Remove it from A using **Remove from this record** and confirm that the document remains under B. Do not test the recorded cascade policy as an available m:n deletion workflow. See Share documents between records (m:n) for user steps and partial-save recovery.

5.3. Configure a Simple Document scenario

Select the new scenario in the sidebar and work through its document settings. Start with a small set of useful fields, save, then test it as a normal user before making it visible to the wider audience.

1. Set the displayed title, description and dashboard visibility. A useful title such as Invoice documents is easier to recognize than a technical type name. Hiding a scenario removes ordinary navigation; it does not revoke repository access.
2. Open General → Document type, versions and content. Enable globally available attributes and give them clear labels. Review mandatory, primary-key, read-only and unique badges before enabling creation or editing.
3. Open View Configuration → Documents. Configure table persistence in Table, search persistence in Search, and creation/editing switches in Details. Persistence controls remembered view state; it does not store repository documents. Repository permissions still decide whether an action is allowed.
4. Configure Content and document Versions as described below. Choose a default available content element and decide which originals and renditions users can view or download.
5. Configure Document views: the table for finding items, search fields for narrowing results, detail/form fields for inspecting and editing them, and version fields where history is enabled.
6. Save and reload. Open the scenario as a user, search for a known item, inspect its metadata and content, and check that the enabled actions and required fields are usable.

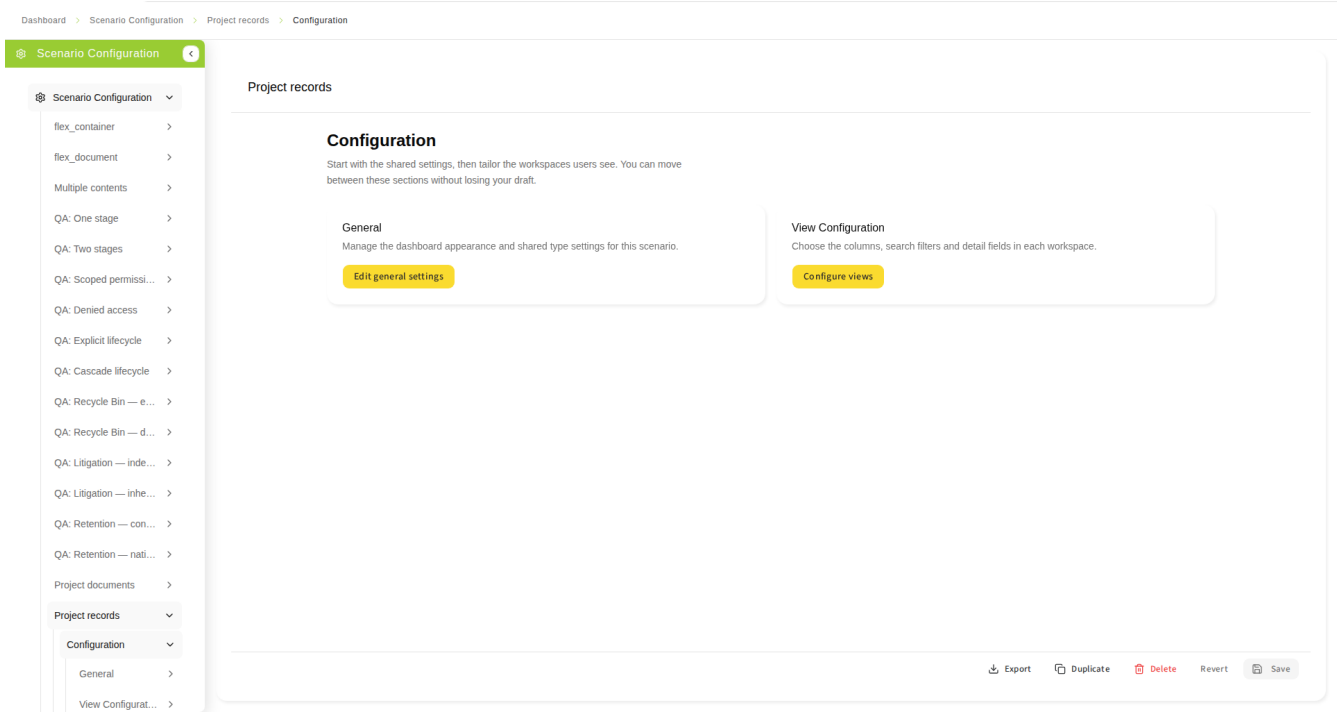


Example: for invoices, expose invoice number and date in the list, add corresponding search fields, and organize writable metadata in a meaningful form section. Keep repository-maintained fields read-only. Enable download only for the content elements users need.

Configuration navigation stops at four editor pages: **General** → **Overview** for naming, icon and dashboard visibility; **General** → **TypeDefinitions** for document and record attributes, attribute lookups, content and versions; and **View Configuration** → **Records** and **View Configuration** → **Documents** for grouped view settings.

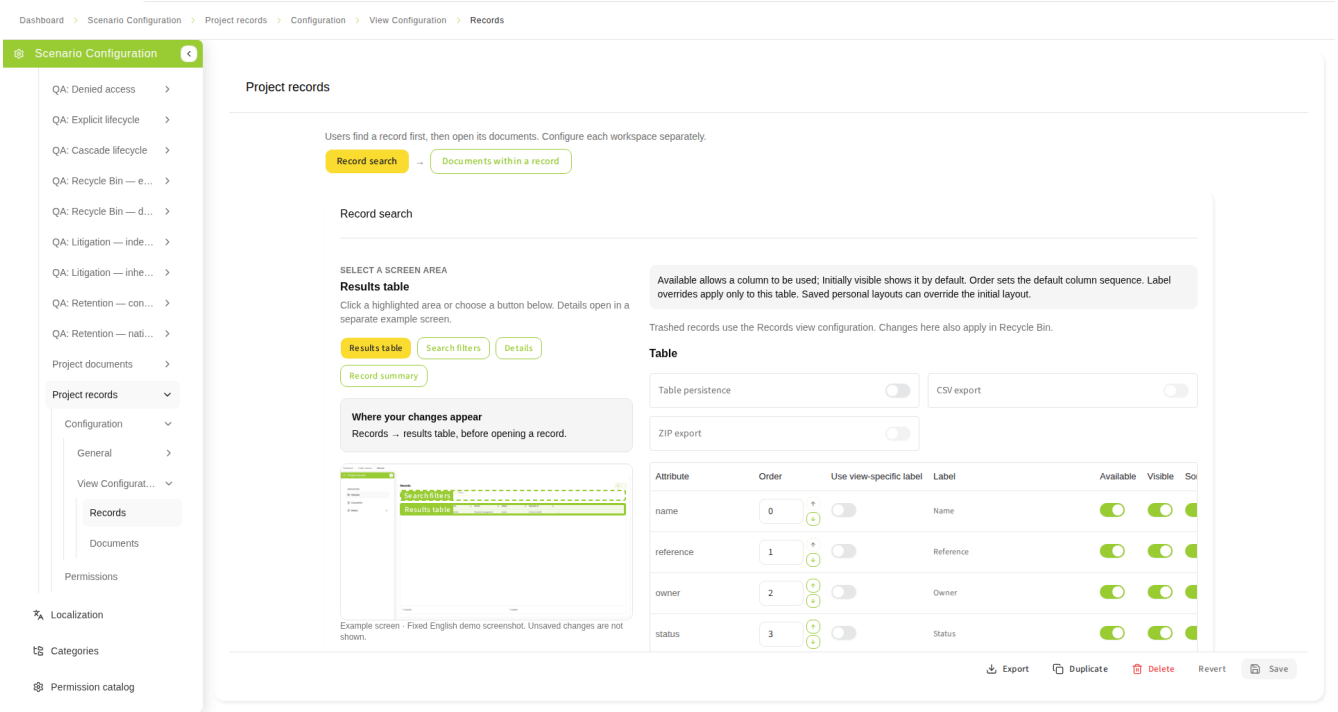
5.4. Configure a Record scenario

A Record scenario has shared record/document type settings and three independently configured runtime workspaces: Record Search View, Record Documents View, and Documents (all documents). View Configuration groups the record workspaces on the Records page and the all-document workspace on the Documents page. Documents inside a selected record can have different table columns, search filters, detail fields, action permissions and saved preferences from all documents.



1. Review the immutable child type and relationship. Confirm that the derived parent type represents the intended record, such as a case or folder.
2. Open General → TypeDefinitions and use the Record sections to configure shared record attributes and history. Include fields that help identify the parent, such as its name or reference number. Parent create/edit settings do not enable child creation or editing.
3. Under View Configuration → Records, expand the Record search sections and configure record table columns, search filters and details. On that same Records page, configure the Record summary card and expand the Documents in record sections for child-document table, search and details. Summary-card field visibility permissions are independent of record-detail visibility. Set summary order, labels and highlighting where offered. Include enough information to distinguish similarly named records.
4. Configure shared child attributes, versions and content under the Document sections in General → TypeDefinitions. Configure View Configuration → Documents separately from the Documents in record sections on Records. Both use child attributes; only Record Documents View is scoped by the selected relationship and record.
5. Configure document originals, renditions and annotation behavior. Parent records without content do not gain a document viewer merely because document content is enabled.
6. Enable Record versions/comments and Document versions/comments independently. Choose version fields appropriate to each type.

7. Review the stored record-deletion policy before exposing deletion; the configuration editor does not offer a Record management card or Delete child documents toggle. In a one-to-many scenario, cascade deletes child documents first; without cascade, the client clears their foreign-key relationship before deleting the parent. A mandatory foreign key forces cascade behavior. In m:n scenarios, configure the separate record selector; record deletion remains blocked with either cascade setting.
8. Save and test both stages: find a record, open its child list, check the summary, inspect a child and return to the parent list. Repeat with another record to verify that documents are scoped correctly.



Example: put case number and case title in the Record table and summary, and document name and document date in Document views. A case title label does not rename a child document field. Explain deletion policy to users before allowing parent deletion.

5.5. General settings and attributes

Overview settings are always expanded. Use General → Overview to set the localized title, description and dashboard visibility. Use General → TypeDefinitions to edit shared settings. The Record card appears first with Attributes, Versions and Attribute lookups; the Document card contains Attributes, Content, Versions and Attribute lookups. One-stage scenarios show only Document. Visibility controls navigation, not security. Each document workspace and the record search workspace have their own creation and editing switches in their Details editor. Disable these actions when they should not be exposed; repository authorization still applies. Deletion availability also depends on permissions, and the stored record-management policy determines the treatment of children. It does not override the m:n deletion safeguard. Existing cascade-deletion and parent-selection settings are retained when editing and saving other settings, but their toggles are not exposed in the configuration editor.

Choose a scenario icon from the icon picker in General → Overview. The icon is used in dashboard cards, favorites and list rows. Choose the default option to restore the scenario initial. Imported

icon identifiers that are valid but unavailable in this client are retained and also display the initial; do not supply SVG markup or an image URL as an icon identifier.

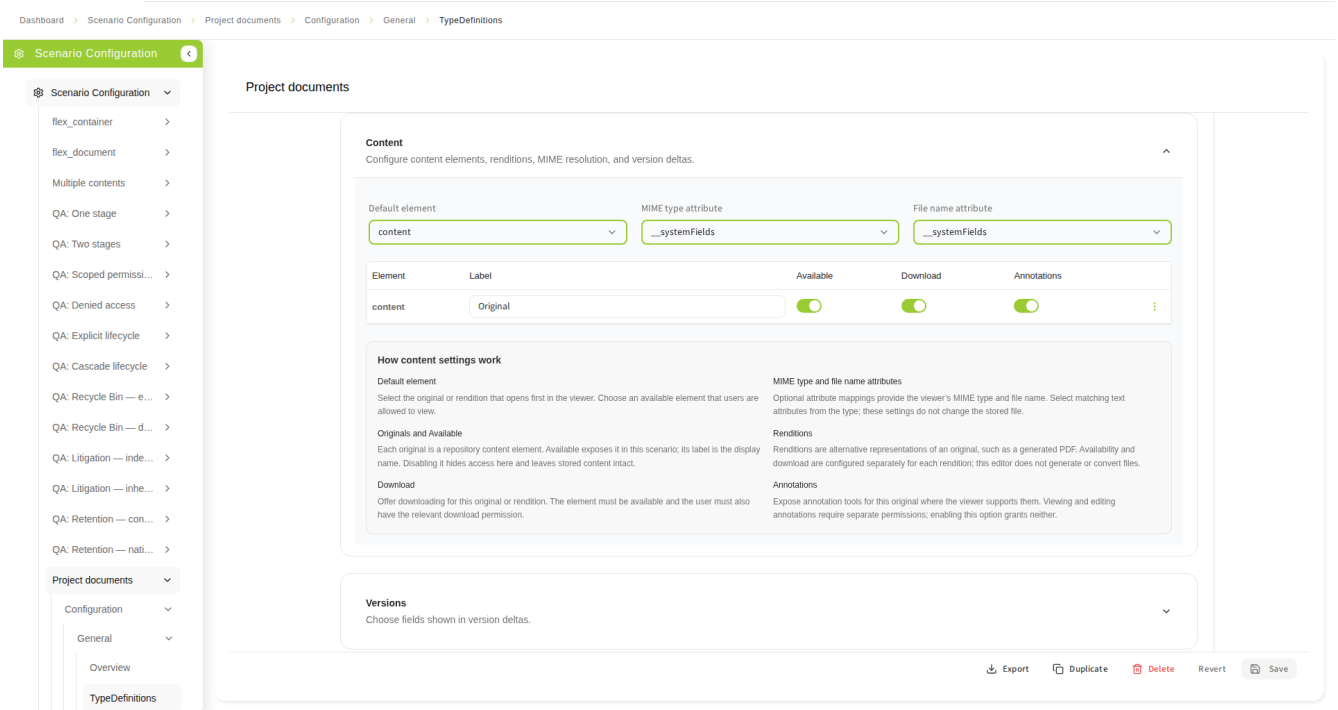
The attribute matrix displays type information and mandatory, primary-key, read-only and unique metadata. These badges describe the repository contract. Enable available fields and preserve required writable fields when creation or editing is enabled. Do not try to make a repository read-only attribute editable.

After adding an attribute to a repository type definition, restart the client to load the updated definition. Existing configurations discover the field in their matrices while preserving authored settings. If you have unsaved administration edits, save or discard them before schema reconciliation completes. Review the new field's availability and view settings, then save the scenario to publish it to runtime views. Discovery alone does not change the published configuration.

Localized text includes a default value and optional values per language. Use the localization drawer to supply translations. A field/view label override applies only where enabled; otherwise the normal configured label applies. Preview each supported language after saving.

5.6. Content, versions and record deletion

Open **General** → **TypeDefinitions** and expand the Document **Content** or **Versions** section. Configure original and rendition labels, availability and independent download switches. Choose an available default content element or rendition. Annotation settings belong to the original and are shared by its renditions; default visibility must be disabled when annotations are disabled.



Optional filename and MIME attributes must reference writable string fields. If omitted, repository content metadata supplies the values. The primary upload populates configured filename/MIME fields; other content elements retain their own metadata. Editing the filename before submission also changes the primary content's system name.

Enable version history and comments separately for document and record stages. These options control the client surfaces, not underlying permission grants.

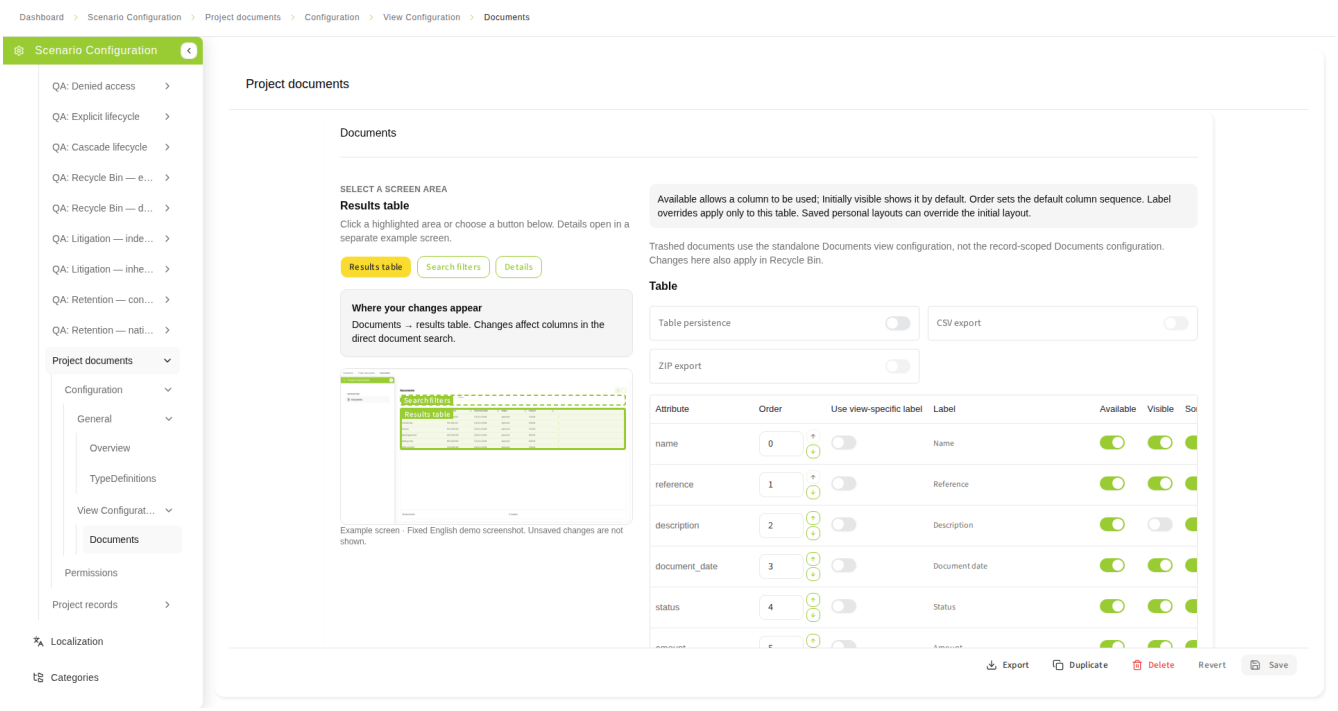
For **1:n with foreign key** scenarios, review the stored `recordManagement.cascadeDeleteEnabled` policy in a configuration export. With cascade enabled, deleting a parent deletes its documents first. Without it, the foreign key is cleared on the children; their metadata and stored content are retained. A mandatory foreign key forces cascade deletion. The deletion confirmation shows the related-document count and the configured delete or unlink consequence. If the count cannot be loaded, confirmation is blocked. Test the configured policy with a disposable record and explain this consequence to users before exposing record deletion.

For **m:n with shared documents**, the cascade setting is stored policy only: record deletion and global document deletion are unavailable. Membership removal preserves the document, even when its final membership is removed. This behavior is independent of the stored cascade setting.

5.7. Configure views

Open **View Configuration** and choose **Records** or **Documents**. In Records, choose **Record search** or **Documents within a record**. Select **Results table**, **Search filters**, **Details** or the record-owned **Record summary** using the labelled buttons or highlighted screenshot areas. The selected settings open immediately; these selections do not add deeper navigation. Configure globally available attributes first, then their supported use in search, table, detail and create/edit views. Adjust order and labels with the provided controls. Only actual runtime consumers are customer features: a configuration field alone does not establish an available UI action.

The example screen stays visible beside the settings on wide screens and above them on smaller screens. **Where your changes appear** explains the runtime location, and the settings guidance explains availability, initial visibility, order and label overrides. Record summary settings belong to the record even though the summary appears beside its documents. Example screenshots are fixed English demo images; they do not reflect unsaved changes. Documents examples match the scenario layout: one-stage examples have no scenario side navigation, while two-stage examples retain it. Documents within a record use their own examples and settings. Use the area buttons if an image cannot load. Switching areas retains your draft; the footer reports unsaved changes until you choose Save.



- Search: enable fields and arrange their order. Ordinary controls and operators follow each field type; lookup dropdowns have their own search settings described below.
- Table: configure available columns, ordering and visibility and sorting. Verify the result in the runtime list.
- Detail/forms: configure field ordering, sections and read-only behavior where supported; required writable fields must remain accessible for enabled mutations.
- Two-stage summary: configure parent summary fields and localized labels, then verify the selected-record context. New defaults contain up to four fields; an authored summary may include more. Existing available fields and their settings are preserved when the administration draft is initialized or saved.

5.7.1. Understand the view matrices

Setting	Meaning and practical use
Global attribute Available	Makes an attribute eligible for configured views. Enable it here before using it in a view. Turning it off can remove dependent uses; inspect the editor's confirmation and validation.
View Available	Includes the field in this particular table, search, detail, version or summary view. It does not grant repository permission.
Order	Moves a field to the requested position in that view. Orders in another stage or matrix remain independent.
Label override	Uses a view-specific label when enabled. Otherwise the stage attribute label is used. Supply translations through the provided translation action.
Table Visible	Controls initial visibility among available columns. Users may have remembered column choices when persistence is enabled.
Table Sortable	Allows sorting by the column where supported by the repository/runtime. Enable it for useful sortable fields, not every displayed value automatically.
Detail Section	Groups fields under a localized section heading, for example General or Classification.
Detail Read-only	Prevents editing that field through the configured form. It cannot make a repository read-only attribute writable.
Version fields	Determines which configured metadata fields are shown for history. Enabling history and selecting its fields are separate choices.
Summary fields	Displays parent context for Record scenarios, with order, labels and highlighting where offered. It does not change child attributes.

Do not hide required writable inputs from enabled creation/editing workflows. Primary keys, system-maintained fields and the relationship managed by a Record scenario may be locked. A field can be visible in a table and absent from search, or searchable and absent from the table; these are independent choices.

Column-header filtering and CSV/ZIP table export are not available in this version. Do not configure a workflow around those actions.

5.7.2. Configure lookup-backed attributes

Open **General** → **TypeDefinitions** and expand **Attribute lookups** inside the Document or Record card. Select an attribute and enable a shared lookup mapping. Lookup sections expand like Attributes, Content and Versions; they are not separate nested cards. The header lists attributes with enabled mappings as **Active lookup filters**, or shows **No lookup filters set up** when none exist. This summary remains visible when collapsed and updates with draft edits; it does not depend on dropdown search being enabled. Disabled mappings and imported filter/form settings without a shared mapping are not listed as active. The source must be an accessible repository document or container type represented by a configured scenario of the same type and object kind.

A container source needs the record stage of a Record scenario; a document source needs a document stage. Users need access to that source scenario and permission to read its records or documents; configuring a lookup does not grant these permissions.

Make every configured display, stored-value and dropdown search column visible in the source stage's Details and permitted for the user. This includes `id` when explicitly chosen for any of those purposes; retaining an item's identity internally does not make `id` an unrestricted lookup column. The active scenario is preferred when it is an eligible source; otherwise sources follow configuration order. If a matching active source hides a required column, the lookup does not switch to another scenario to bypass the restriction. Missing source access or hidden required columns make the lookup unavailable rather than exposing restricted fields. Check this setup with the intended normal-user permissions, not only as an administrator.

For a Record scenario, the document lookup source, stored column and labels are shared by the all-document and record-scoped document workspaces. Their filter and form dropdown settings are independent. Open the record view configuration and select its document workspace to edit those scoped settings; changing them does not change the all-document dropdowns. Saving validates all configured workspaces, including retained settings on a workspace you are not currently editing.

When reopening the editor or switching scenarios, the attribute selector initially shows the first attribute with a saved lookup, if one exists. Select another attribute to inspect or configure its settings; an ordinary attribute correctly shows lookup disabled. An explicit attribute choice remains selected while editing the current stage.

1. Choose the source type and a **unique stored-value column** compatible with the target attribute's scalar type. For example, store a unique customer code while displaying the customer's name and city. A composite unique key is not sufficient for a single stored column.
2. Select and order the display columns. Their formatted values are joined with ` — `; users see this label, but queries and saves use only the stored value.
3. Choose what happens when a saved value is no longer available: **Preserve** keeps it, while **Require replacement** requires a valid selection before an editable form can save. A temporary lookup failure is not proof that the value was removed.
4. Configure filter and form dropdown search independently. Enable search and select columns with Equals, Contains or Starts with. Only display columns have an **Order** control; search-column order does not affect matches or result sorting. Text matches are case-insensitive and matches across columns combine with OR. Non-text scalar columns support Equals only. A query that cannot represent a column's type skips that column, so a name search can still match text columns alongside numeric Equals columns. If no column can accept the query, no matches are returned. Disabling search leaves browsing available.
5. For the filter, choose single or multiple selection. Users select individual values and can retain selections across dropdown pages. Forms always select one value.
6. Save the scenario, reload it, and verify the permitted workflow as a normal user. Full version-2 exports retain the mapping and both sets of dropdown settings; existing configurations without these optional fields remain valid.

Each dropdown search attribute may appear only once, even with a different operator. Imported column objects can pass JSON Schema's whole-object uniqueness check yet fail semantic validation

for repeating an attribute; remove the duplicate before saving. Parent selection and lookup mappings do not override litigation edit protection, retention lifecycle restrictions or recycle-bin read-only rules. Grant ordinary edit permissions and, for protected held entries, the separate configured permission to edit under litigation. Retention management and lifecycle override grants remain independent.

Lookup-filter **Select all** is not available; it is deferred to AFC-134. Imports reject the retired `selectAllEnabled` property even when it is `false`; remove only that property from older branch-created exports. Before deploying this unreleased contract update, an administrator must also clean any saved configuration containing it, otherwise startup validation fails. There is no automatic cleanup or data reset; preserve other settings and saved filter selections.

Changing a source does not silently rewrite the stored, display or search columns. Removed, inaccessible or incompatible references remain in the draft with validation errors so that you can repair them explicitly. A mapping on a foreign-key field must match the repository's declared relationship; an ordinary lookup does not create a relationship.

Imported filter/form lookup settings without a shared attribute mapping remain marked for repair. Enabling the shared mapping preserves those authored surface settings; complete the source and column mapping before saving. Scenario, category and localization configuration saves all apply the same lookup and parent-selection validation, so stale settings must be repaired even when the current edit is elsewhere. Saving unrelated settings preserves untouched scenario configuration, including its existing export flags; validation still applies to retained lookup mappings.

For a strict one-to-many foreign-key scenario, the optional `recordManagement.parentSelectionEnabled` setting is disabled by default. The configuration editor does not offer an Allow choosing another parent record toggle; existing or validated imported settings remain supported. When enabled in the stored configuration, users can choose another parent during document creation or editing. The picker derives its source and stored column from the selected relation; a non-ID referenced key is supported. Existing create/edit grants and relation-field read-only rules still apply. This setting does not alter many-to-many membership workflows or introduce new permissions.

Parent selection uses the dropdown search settings and relation-field read-only rule of the current document workspace. A read-only relation in the record-scoped form remains locked even when the all-document form allows reassignment, and conversely.

5.8. Organize scenarios into categories

Open **Categories**. Categories organize dashboard navigation; they do not create repository folders, move documents or grant permissions. Each category has a stable ID and a required default label, with optional translations. Renaming or translating a label does not change the ID or its assignments.

1. Enter a nonblank label and choose **Add root category**. To create a nested category, select its parent and choose **Add child category**.
2. Select a category to edit its default label or translations. Do not leave the required default label empty or whitespace-only.
3. Use the parent selector to move a category and its children under another category, or back to

the root. A category cannot become its own ancestor. Moving to another parent appends it to that parent's children.

4. Use **Move up** and **Move down** to order categories among siblings. Category order controls navigation order, not the order of scenarios in the catalog.
5. Under **Scenario assignments**, choose one category for each scenario or leave it **Unassigned**. Hidden scenarios can also be assigned. A parent dashboard filter includes scenarios assigned to its descendants; unassigned scenarios appear under All categories only.
6. Choose **Save all** to persist the shared configuration draft, including any retained scenario or dashboard-presentation edits. Verify the saved dashboard as a normal user.

Deleting a category requires confirmation and removes its entire subtree from the draft. Scenarios assigned anywhere in that subtree become unassigned; the scenarios and repository data are not deleted. Save all to persist the removal. Back up the full configuration when assignments must be recoverable; a category-only export does not contain them.

5.8.1. Choose shared dashboard presentation

In Categories, choose the tree/category presentation or the flat catalog and then **Save all**. Tree presentation shows category navigation. Flat presentation shows all visible scenarios in configured order without a category filter. Both retain search, favorites, cards/list selection and pagination.

Choose **Save all** after changing this setting. Category-only and presentation-only saves preserve untouched scenario settings; switching Flat catalog to Categories / tree does not require editing a scenario first. This is an administrator-owned setting for everyone, not a private user preference. Switching to flat retains the category tree, translations and scenario assignments; switching back to tree restores their navigation structure. It also preserves users' favorites and cards/list preferences.

5.8.2. Import and export categories only

Export categories downloads `categories.json`: a JSON array containing the current draft's complete category tree, including unsaved edits, stable IDs, translated labels and child order. It contains no scenarios, category assignments, scenario icons or dashboard presentation. For example:

```
[
  {
    "id": "finance",
    "label": { "default": "Finance", "values": { "de": "Finanzen" } },
    "children": [
      { "id": "invoices", "label": { "default": "Invoices" } }
    ]
  }
]
```

1. Choose **Import categories** in Categories and select a category-array file, not a full configuration envelope.

2. Review validation errors before retrying. IDs must be globally unique and labels valid. New roots and children are appended in imported order. An existing ID must retain the same label/translations, parent and sibling position; a conflicting definition aborts the whole import. Use the editor, not import, to rename or move an existing category.
3. After successful validation, review the merged draft and choose **Save all**. Import does not immediately persist anything and does not change scenario assignments or dashboard presentation.

An empty or identical category import makes no changes; it does not delete existing categories. Invalid files, conflicting definitions or failed validation leave the draft unchanged. If the draft changes while validation is running, repeat the import against the current draft rather than assuming the earlier result was applied.

5.9. Save, duplicate, order, import and export

5.9.1. Save and discard scopes

Save editor changes and verify the result after reloading. Save success and failure messages appear as toast notifications, consistently across the administration workspaces. Field validation remains beside the affected controls. Failed saves retain your changes; a success notification confirms that the write completed. The scenario editor's **Save current** saves the selected scenario. When category or dashboard-presentation changes are pending, the action becomes **Save all** and saves the complete shared configuration draft, even when the selected scenario itself is unchanged. **Revert current** discards only the selected scenario's edits; it does not undo category, presentation or other scenario edits.

Save all in Categories also persists retained scenario edits. Confirmed configuration reload discards the whole shared configuration draft and restores saved data, rather than only the visible workspace. Export needed drafts before discarding them.

Leaving administration with unsaved configuration or localization changes opens **Unsaved administration changes**. Choose Save, Discard or Cancel deliberately. A failed save retains the unfinished draft. While a save is in progress, wait for it to finish before navigating or starting another write. The browser's close/reload warning is not a save action.

Duplicate a scenario when you need a similar configuration with a new identity. Change order using the sidebar ordering controls and save where requested. Export one scenario or the full configuration through the provided action. Store the export securely; it may contain business-specific labels and configuration.

5.9.2. Configuration transfer and legacy files

Configuration exports use a version-2 envelope with `schemaVersion`, `categories`, `scenarios` and `scenarioPresentation`. A single-scenario export uses the same envelope, including the full category tree and presentation but only the selected scenario. It is not a bare scenario object or a category-only array. Full configuration export is the appropriate backup for scenarios, their assignments and the shared dashboard layout; neither export includes private dashboard preferences.

To import configuration, use the JSON import action in Scenario configuration. Imports merge by scenario name, keeping unrelated scenarios and categories. Matching names update compatible scenario settings; they do not permit changing immutable structure or conflicting category assignments. Category conflicts abort the import. The complete merged draft must pass schema and live repository validation before it is applied. Review it and save explicitly; import success is not a successful save.

An explicit `scenarioPresentation` in a supported versioned import selects tree or flat. If it is omitted from a supported versioned import, the current draft choice is retained. This merge behavior differs from a new v2 configuration, where omission defaults to tree.

Legacy files containing only `{ "scenarios": […] }` or `{ "schemaVersion": 1, "scenarios": […] }` are also accepted by this import action. They are converted to v2 with flat presentation before validation, upgrading document-view structure while preserving authored settings. Existing unrelated scenarios and categories remain; an existing scenario's conflicting category assignment still prevents import. Unknown fields/versions and mixed legacy/versioned envelopes are rejected. An empty scenario array does not clear the existing configuration. Subsequent exports always use v2.

Normal saves and new exports use configuration version 2. Full exports preserve categories, scenario category assignments, optional icons and dashboard presentation. Supported legacy V1/unversioned payloads are normalized to flat V2 with an empty category list before merging; this does not clear the draft's existing categories or change existing relationship models. The backend upgrades valid stored V1 configuration on startup rather than requiring a reset. A merge retains the current dashboard presentation when the imported V2 file omits that setting.

Delete only the intended scenario after reviewing confirmation. Configuration removal does not itself describe a repository data-deletion workflow. Reset replaces configuration with seeded defaults; back up first and use reset only in an explicitly disposable or otherwise authorized environment.

5.10. Categories and dashboard presentation

Open **Categories** from the settings sidebar. Create root or child categories, configure their localized labels, and arrange their navigation order. Stable category identities are distinct from repository record identities; moving or assigning a scenario does not link documents to records.

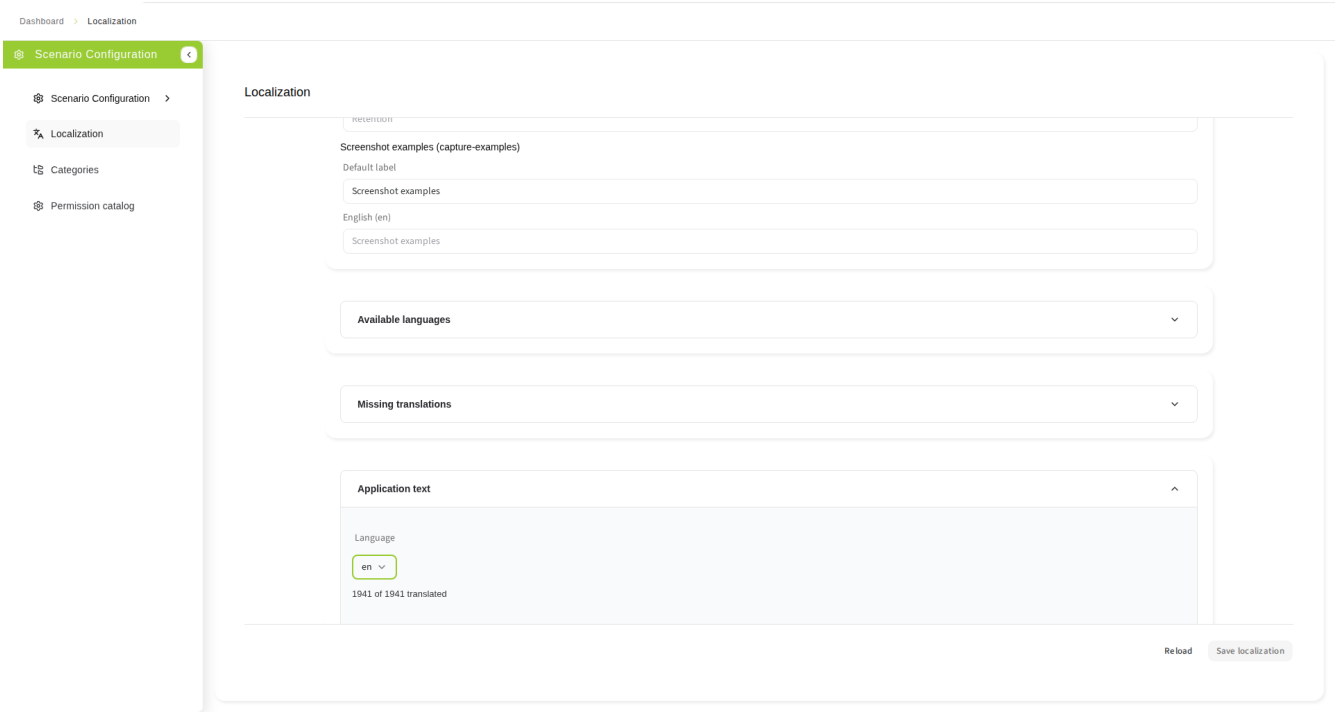
Category and scenario translation keys use canonical language tags, matching language settings. Examples include `de`, `de-DE`, `zh-Hans`, `en-001` and `sr-Latn-RS`. Incorrect casing, underscores and malformed or repeated subtags are rejected by configuration validation.

Assign scenarios to categories and choose **Tree** or **Flat** presentation. Tree presentation uses the authored category hierarchy; flat presentation keeps scenario navigation without that hierarchy. Save changes explicitly and check the dashboard as a normal user. Category and scenario changes share the configuration draft: preserve unsaved edits when switching workspaces, and resolve stale-save errors before retrying.

5.11. Manage languages and translations

On first system startup, English and German UI catalogs are installed automatically from the packaged application; no translation-provider setup or manual import is required. The shipped scenario and category labels also include both languages. Later startups add newly shipped UI keys and update untouched shipped defaults while preserving administrator overrides and saved scenario/category configuration. Customer-authored text requires its own translations.

Open **Localization** from the administration navigation. Choose a catalog language, search by translation key or text, edit the translation and save. Enable a language for users only when it is suitable for your audience. Select the language on the dashboard and check both static UI text and scenario labels.



Editing a label's default-language text also updates an inherited translation for that same language. Translations in other languages remain unchanged.

Missing text uses the default language. Scenario-owned labels and global UI translations are separate. An unsaved translated draft is not persisted by changing the runtime language. Translation generation, when offered, needs a configured available provider; review generated text before accepting overrides. This guide does not assume that a generation provider is installed.

Category labels belong to the shared configuration, not the global UI catalog. In Localization, edit the required raw default label separately from explicit translations for each language. Changing the catalog's default language does not move or overwrite a category's stored translation. Empty or whitespace-only category defaults block saving and must be corrected at the field. Label edits do not change IDs, hierarchy or scenario assignments.

Localization saves only resources with pending edits. If the shared configuration and catalog both have changes, configuration is saved first, followed by the catalog. A partial-success message means configuration was saved but catalog changes remain unsaved; correct the error and retry the remaining work. Saving configuration may include scenario, category or presentation edits

retained from the other administration workspaces.

Reload is unavailable during a save. When shared configuration edits exist, its confirmation covers both localization and configuration: confirming discards retained scenario, category and presentation edits as well as localization changes. Cancel to keep those drafts.

5.12. Validation and concurrent editing

Invalid references, duplicate scenario names or category IDs, blank category defaults, dangling category assignments, unavailable default content, inconsistent annotation switches, missing required fields and unsupported relationships prevent a save. Correct the indicated setting rather than removing validation.

A stale save is rejected when another administrator has changed the scenario or shared configuration being saved. Your draft is retained. Preserve a copy of the intended edits, explicitly reload the saved configuration and reapply necessary changes. Remember that a shared reload can discard edits from other administration workspaces. This version does not provide a conflict-specific merge interface. For authorization or session errors, sign in again through Shell and check current state before repeating writes.

5.13. Import reference

5.13.1. Configuration import fields

Field	Type / values	Required
`envelope.schemaVersion`	2	yes
`envelope.scenarioPresentation`	tree, flat	no
`envelope.categories`	array	yes
`envelope.scenarios`	array	yes
`localizedText.default`	string	yes
`localizedText.values`	object	no
`category.id`	identifier	yes
`category.label`	localizedText	yes
`category.children`	array	no
`labelOverride.enabled`	boolean	yes
`labelOverride.value`	localizedText	no
`attributeLookup.sourceType`	identifier	yes
`attributeLookup.storedValueColumn`	identifier	yes
`attributeLookup.displayColumns`	array	yes
`attributeLookup.unavailableValuePolicy`	preserve, requireReplacement	yes

Field	Type / values	Required
`repositoryLookupSearchColumn.attribute`	identifier	yes
`repositoryLookupSearchColumn.operator`	equals, contains, startsWith	yes
`lookupSearch.enabled`	boolean	yes
`lookupSearch.columns`	array	yes
`searchFilterLookup.selectionMode`	single, multiple	yes
`searchFilterLookup.search`	lookupSearch	yes
`detailFieldLookup.search`	lookupSearch	yes
`attribute.attribute`	identifier	yes
`attribute.label`	localizedText	yes
`attribute.available`	boolean	yes
`attribute.lookup`	attributeLookup	no
`tableColumn.attribute`	identifier	yes
`tableColumn.available`	boolean	yes
`tableColumn.initiallyVisible`	boolean	yes
`tableColumn.order`	integer	yes
`tableColumn.initialWidth`	number	no
`tableColumn.sortable`	boolean	yes
`tableColumn.headerFilterEnabled`	boolean	yes
`tableColumn.labelOverride`	labelOverride	no
`searchFilter.attribute`	identifier	yes
`searchFilter.available`	boolean	yes
`searchFilter.order`	integer	yes
`searchFilter.labelOverride`	labelOverride	no
`searchFilter.lookup`	searchFilterLookup	no
`detailField.attribute`	identifier	yes
`detailField.available`	boolean	yes
`detailField.order`	integer	yes
`detailField.readOnly`	boolean	yes
`detailField.section`	localizedText	yes
`detailField.labelOverride`	labelOverride	no
`detailField.lookup`	detailFieldLookup	no

Field	Type / values	Required
versionField.attribute	identifier	yes
versionField.available	boolean	yes
versionField.order	integer	yes
versionField.labelOverride	labelOverride	no
summaryField.attribute	identifier	yes
summaryField.available	boolean	yes
summaryField.order	integer	yes
summaryField.highlight	boolean	yes
summaryField.labelOverride	labelOverride	no
table.persistenceEnabled	boolean	yes
table.columns	array	yes
table.csvDownloadEnabled	boolean	yes
table.zipDownloadEnabled	boolean	yes
search.persistenceEnabled	boolean	yes
search.filters	array	yes
search.includeTrashedEnabled	boolean	no
detail.fields	array	yes
detail.editEnabled	boolean	yes
detail.createEnabled	boolean	yes
versions.enabled	boolean	yes
versions.commentsEnabled	boolean	yes
versions.fields	array	yes
stage.attributes	array	yes
stage.table	table	yes
stage.search	search	yes
stage.detail	detail	yes
stage.versions	versions	yes
stage.summaryCard	object	no
rendition.element	identifier	yes
rendition.label	localizedText	yes
rendition.available	boolean	yes
rendition.downloadEnabled	boolean	yes
original.element	identifier	yes

Field	Type / values	Required
original.label	localizedText	yes
original.available	boolean	yes
original.downloadEnabled	boolean	yes
original.annotations	object	yes
original.renditions	array	yes
content.defaultElement	identifier	yes
content.originals	array	yes
content.mimeTypeAttribute	identifier	no
content.fileNameAttribute	identifier	no
recordManagement.cascadeDeleteEnabled	boolean	yes
recordManagement.parentSelectionEnabled	boolean	no
recordManagement.recordSelection	recordSelection	no
recordManagement.cascadeTrashEnabled	boolean	no
recordManagement.cascadeRestoreEnabled	boolean	no
recordManagement.trashChildrenMode	automatic, ask	no
recordManagement.restoreChildrenMode	automatic, ask	no
recordManagement.restoreParentMode	automatic, ask	no
scenario.name	identifier	yes
scenario.categoryId	identifier	no
scenario.icon	string	no
scenario.title	localizedText	yes
scenario.description	localizedText	yes
scenario.visible	boolean	yes
scenario.kind	ONE_STAGE, TWO_STAGE	yes
scenario.twoStageType	STRICT_ONE_TO_MANY, MANY_TO_MANY	no
scenario.documentType	identifier	yes
scenario.recordType	identifier	no
scenario.relation	identifier	no
scenario.document	stage	yes
scenario.record	stage	no

Field	Type / values	Required
`scenario.content`	content	no
`scenario.recordManagement`	recordManagement	no
`scenario.permissions`	scenarioPermissions	no
`scenario.recordDocuments`	documentView	no
`scenario.recycleBin`	recycleBin	no
`scenario.litigation`	litigation	no
`scenario.retention`	retention	no
`permissionRule.allow`	array	yes
`permissionRule.deny`	array	yes
`fieldPermission.stage`	document, record, recordDocument	yes
`fieldPermission.surface`	table, detail, create, update, summaryCard	yes
`fieldPermission.attribute`	identifier	yes
`fieldPermission.visibility`	permissionSetting	no
`fieldPermission.editability`	permissionSetting	no
`scenarioPermissions.actions`	object	yes
`scenarioPermissions.fields`	array	yes
`scenarioPermissions.defaults`	permissionRule	no
`scenarioPermissions.newTargetMode`	defaults	no
`permissionSetting.allow`	array	no
`permissionSetting.deny`	array	no
`permissionSetting.mode`	none, defaults	no
`documentView.table`	table	yes
`documentView.search`	search	yes
`documentView.detail`	detail	yes
`recordSelection.displayColumns`	array	yes
`recordSelection.search`	object	yes
`recycleBin.enabled`	boolean	yes
`recycleBin.viewEnabled`	boolean	yes
`recycleBin.deleteOnlyTrashed`	boolean	yes
`litigation.enabled`	boolean	no
`litigation.viewEnabled`	boolean	no
`litigation.preventUnprivilegedEdit`	boolean	no

Field	Type / values	Required
`litigation.setChildrenMode`	disabled, ask, automatic	no
`litigation.removeChildrenMode`	disabled, ask, automatic	no
`litigation.setParentMode`	disabled, ask, automatic	no
`litigation.setSiblingsMode`	disabled, ask, automatic	no
`retention.enabled`	boolean	no
`retention.tableVisible`	boolean	no
`retention.detailVisible`	boolean	no
`retention.protectedLifecycleEnabled`	boolean	no
`retention.childrenMode`	disabled, ask, automatic	no
`retention.parentMode`	disabled, ask, automatic	no
`retention.siblingsMode`	disabled, ask, automatic	no

NOTE Conditional requirements and live repository validation also apply. Export an existing scenario as a starting point; this inventory does not describe every semantic constraint.

5.14. Configure scenario permissions

Select a scenario and use the **Configuration** and **Permissions** navigation destinations. Both edit the same scenario draft. Permission drawers stage their changes separately until you choose Save; the page Save/Save all then persists the scenario draft. The Permissions overview groups scenario access, actions and field controls. Search by feature, attribute or criterion; filter explicit rules, configuration defaults, entries without grants or disabled targets. Grouped tables show **Name**, **Allowed** and **Denied** with the complete configured identity labels and a three-dot menu at the end of each row. Empty sections are hidden. These lists describe configuration, not a user's effective access.

Edit permissions only in the scenario's **Permissions** section. Configuration feature switches and view matrices contain no permission controls. Each permission row's three-dot menu contains only **Edit Permission**, which opens the shared right-hand drawer. Browse one catalog, initially filtered to **All**, and select an identity; its dialog offers green **Allow**, red **Deny**, and **Cancel**. The wide drawer keeps the Allow and Deny lists prominent above the catalog, with a Remove button for every entry in either list. Any allow criterion may grant; any matching deny wins. Client roles retain their client identity, groups retain full paths, and claim values retain their type. Disabled features retain editable rules.

Configure the scenario-wide **Configuration defaults** rule above the permission tables. Edit its Allowed and Denied criteria using the same staged drawer. Configuration status is read-only text in the table. Choose **No grants**, **Configuration defaults**, or **Explicit permission rule** inside the editor. Rows using defaults show their resolved criteria, labelled as inherited, and follow changes to the shared rule. No grants always denies. Explicit rules remain independent. New scenarios start with

every row using defaults and a shared Allowed criterion for realm role `flex-admin`, with no Denied criteria. Administrators can edit this initial rule; existing scenarios retain their saved settings. Previously omitted field rules show *Existing configuration behavior*; explicitly choose Configuration defaults to opt them into the shared rule. Without a default rule, inherited actions grant nothing and inherited fields retain their configuration behavior. An explicitly empty default rule denies.

For fields, table visibility, detail visibility, create visibility/editability and update visibility/editability are independent. Rules can only restrict existing schema/configuration availability and writability. Removing the final criterion from an explicit rule leaves an empty denying rule.

At the bottom of Permissions, **Reset to factory defaults** asks for confirmation, restores the shared Allowed criterion for realm role `flex-admin`, clears Denied criteria and all explicit row rules, and sets every row to Configuration defaults. Other scenario settings are preserved. Reset remains an unsaved draft change until page Save/Save all.

Select multiple rows across sections to add identities to all selected Allowed or Denied lists. Section selection affects only its visible rows; selected rows remain selected when filtering. Clear the selection explicitly or switch scenarios. Bulk additions merge into existing rules without duplicates or removals. Save in the drawer applies additions to the scenario draft; Cancel discards them.

Choose **Validate Permissions** for advisory checks of the current draft. Hints identify missing catalog identities, duplicates, identities present in both lists, obsolete field references, unavailable grants and missing access/viewing prerequisites. For example, annotation grants cannot enable content whose availability is disabled. Findings refresh after draft changes and link to the affected configuration. Checks do not simulate user memberships, block saving or automatically change rules.

A stale catalog warning retains existing selections. Deployment operators provision read-only discovery credentials and register selectable audiences and typed claim values. Administrators do not select criteria based only on their own identity. Imports reject unsupported new criteria, including deny rules; correct the complete mapping before saving.

Schema v2 exports include the scenario's action rules and field overrides, with no discovery credentials, cached catalogs or current-user grants. Supported v2, v2 and v2 imports upgrade without generating grants. Existing scenarios without mappings deny access after migration: configure initial access and action mappings before rollout. Configuration administration does not automatically grant business access.

5.15. Browse the permission catalog

Open **Permission catalog** in administration navigation to browse all discovered realm roles, client roles and groups, independently of your own memberships. Search by role, client identifier or full group path and filter by category. Audiences and custom claims are registered by operators and are marked separately from Keycloak entries.

The page shows the last successful discovery and warns when its snapshot is stale. **Reload catalog** reloads the server snapshot; discovery retries occur at most every five minutes. If discovery has never succeeded, ask your operator to configure the dedicated read-only Keycloak discovery account. Existing choices and scenario mappings survive an outage. The inventory itself remains

read-only. Each entry shows its number of configured occurrences, including unsaved changes and hidden scenarios. Identities referenced by scenarios but absent from the catalog are also shown as **Missing from catalog**. Absence from an unavailable or stale snapshot is unconfirmed.

Sort the catalogue by Source, Identity, Usage count or Status using the column headers. Open **Usage details** from an entry's trailing three-dot menu to see each scenario, action or field/control and its Allowed/Denied disposition. Remove an individual occurrence to update the shared administration draft. Removing the final identity preserves the explicit empty rule; it does not restore field inheritance. Choose **Save** to persist the shared draft, or **Discard changes** to discard all unsaved administration changes, including changes made on other configuration pages. Navigation to a usage retains the draft. **Validate Permissions** checks all scenarios on this page.

To grant or deny access, open the scenario's **Permissions** page and select catalog entries in the target's drawer. Configure **SCENARIO_ACCESS** and the required action rules, then save the scenario. Configuration administration alone grants no scenario access. You can use administration even when the dashboard shows no scenarios. Navigation to the catalog retains unsaved configuration drafts.

The drawer header and Save/Cancel footer stay visible while the body scrolls. **Save** applies only its working copy to the scenario draft. **Cancel**, Escape and drawer dismissal discard drawer-only changes. Canceling the identity-choice dialog leaves the drawer draft intact. Save the scenario page to persist applied edits.

Select a permission name to open **Affected UI**. Choose a location to see its effect and, where useful, a highlighted screenshot. Use **Zoom in** for a larger, readable image. One rule can apply to several locations; changing the selected location does not create or edit a rule. **Edit Permission** opens the existing editor. The preview is independent of bulk-selection checkboxes and remains open when filtering the list. On smaller screens it appears above the tables. **Close preview** closes only the explanation.

The screenshots are static English examples. They do not show your exact configured fields, unsaved changes or a user's effective access. Each field is identified by its actual target label; the image highlights a representative region. Read the conditions for related permissions and configuration requirements. Historical content shares content-view/download rules, while historical annotations cannot be edited. Unimplemented actions and effects without a useful screenshot are explained in text.

Select the **Configuration defaults** heading to inspect targets explicitly using the shared rule. An omitted field rule retains existing configuration behavior and is not an explicit defaults selection.

The Permissions tables contain independent targets for table visibility, detail visibility, create visibility/editability and update visibility/editability. Select targets across sections and choose bulk Allow or Deny to collect additions in the drawer. Save merges criteria, skips duplicates and retains unrelated controls. Bulk operations add only; remove existing criteria or return to defaults by editing an individual permission.

The scenario permission overview shows **Configuration status**. Search applies together with the selected filter. **Configuration defaults** includes explicit defaults selections and field controls retaining inherited configuration behavior; the visual defaults explanation lists only explicit

consumers of the shared rule. Explicit empty rules deny access.

Choose **Use configuration defaults** inside a permission editor to stage an explicit selection of the shared scenario rule. **Save** applies that choice to the scenario draft; **Cancel** retains the previous rule. Visibility and editability remain independent. An omitted field rule retains existing configuration behavior, while an omitted action grants nothing.

The catalogue also discovers explicitly configured access-token audiences and hardcoded string, boolean or number claims from Keycloak clients and client scopes. Operator-registered criteria remain available. Dynamic claim mappers do not provide selectable values, and an empty group category can mean the realm has no groups. The catalogue does not show user memberships or automatically grant permissions.

The permission catalog scrolls continuously without pagination. Permission Identity precedes Source. Click a column heading to sort; the arrow shows its direction. Row three-dot menus open usage details.

5.15.1. Independent document permissions and upgrades

The Permissions editor lists Documents and Record Documents View separately, including content, download, annotation and history actions. A grant in one workspace does not grant the corresponding action in the other. Shared content and version feature switches remain under General and still limit availability in both. The record summary card has its own field-visibility permissions.

Current exports use schema version 2. Supported older imports and stored configurations copy their existing document view settings and permission selections into both document workspaces, retaining existing behavior. Subsequent edits are independent. Existing saved table/search preferences initialize the new view contexts when no new preferences exist. Record-document preferences are shared across records within the scenario; the selected record always restricts the document query.

CSV and ZIP export are not implemented. Their configuration switches remain off and disabled in every workspace. Saving through ConfigUI sets both export flags to false, including configurations imported with those flags enabled.

5.16. Configuration QA behavior

Creation and update forms follow the selected workspace’s detail field order, section headings, labels and read-only settings. In two-stage scenarios, record-scoped documents and the all-documents view retain their independent table, search, detail and permission settings. Disabling a shared attribute removes it from both document workspaces. Stored column preferences cannot restore fields that the current permissions hide.

A permission change also applies when a drawer or confirmation was already open: a revoked action cannot be submitted. Configuration administration alone does not grant business actions. Restoring historical versions, CSV/ZIP export, column-header filtering controls and document summary cards remain unavailable; record summary cards are supported.

5.17. Recycle-bin configuration (in development)

General Overview includes scenario-wide trash, trash-bin visibility and deletion-only-for-trashed settings. Search settings control whether each workspace offers Include trashed items in its table filter bar. The filter also requires the workspace's Trash, View trashed items and Include trashed items in search permissions. Clearing it returns to active items only.

Recycle bin appears below Documents and above History in scenario navigation. Its Records and Documents entries open separate lists with their configured columns; the Documents list also includes trashed documents whose record is active. One-stage bins show Documents only. The breadcrumb returns to the scenario. Trashed items are read-only. Editors offer Move to recycle bin and permanent Delete only when the current state, permissions and deletion policy allow them. The record summary card uses a three-dot menu for the same state-sensitive Edit, Delete, Trash and Restore actions; Restore has a recovery icon. A narrow status column keeps trash markers aligned separately from business values. Permanent deletion from the bin requires typing the displayed item name, or DELETE for bulk operations and items without a display name. Record scenarios provide independent automatic/ask modes for trashing children, restoring children and restoring a document's parent. Native repository cascades are identified as repository-managed and cannot be overridden. Existing disabled trash cascading preserves record-only behavior.

Confirmation shows affected counts and shared/global impact. When children are included in record restoration, all trashed children are restored, even those trashed earlier independently. Declining children applies the action to the record only; declining a required parent restoration cancels document restoration. Relationships are preserved. Each action still requires its scenario permissions. Bulk operations replace the journal; partial failures refresh repository state and can be acted on again using fresh confirmation, without progress/recovery controls.

NOTE	AFC-123 remains in development until replacement runtime, native-cascade and inheritance acceptance is complete. Database and configuration schema remain V2.

The ECR CoreTeam decision on reading retained content and annotations while trashed is pending, for both current and historical versions. Retention alone does not guarantee read access. The original delivered ECR 28.1.0 returns Not found for these reads while trashed; the same reads succeed after restoration. Do not promise preview or download access while trashed until the supported contract is decided and verified against an original delivered ECR artifact. For now, use the authorized Restore workflow and verify access after restoration, as described in Downloads, unavailable content and historical viewing; scenario settings cannot override this repository limitation.

5.18. Lifecycle policy defaults and repository constraints

Setting	Default	Consequence
Scenario Recycle Bin	Off	Enable only for types supporting the repository deleted system field. External views do not support trash.

Setting	Default	Consequence
Recycle Bin navigation / Include trashed items	Off	Navigation and each workspace filter need their own settings and permissions.
Deletion only for trashed items	Off	When enabled, active items must be trashed before irreversible deletion.
Delete documents with record	On for newly created scenarios	Applies with Recycle Bin on or off. Off clears an optional FK; a mandatory FK forces deletion.
Trash documents with record	On when omitted	Off permits record-only trash for explicit cascades; it conflicts with native repository CASCADE.
Trash children / Restore children / Restore parent mode	Automatic	Three independent settings. Ask offers one related-item choice in confirmation.

Repository retention and deletion restrictions are administered in the repository, not through scenario switches. No retention timeout does not mean trash is disabled. The effective inherited FK definition determines native CASCADE; type inheritance alone does not create a cascade. Native downward trash/restore cannot be overridden by a scenario or end-user decline. Restoring a document's trashed parent may also restore its trashed siblings through that native cascade; review the complete count.

An imported legacy restore-cascade false becomes Ask; true or omitted becomes Automatic. Explicit modes take precedence. Saving or editing another mode preserves that choice; disabling trash cascading does not disable child restoration.

For active strict records, the delete/unlink confirmation covers every linked child, including trash. A cascade needs child Delete permissions for active items and Permanent delete plus View trashed items for trash. Unlink needs record-scoped Edit and an editable relationship field. Restore trashed children before unlinking. If trash is disabled, use an authorized enabled-bin scenario to resolve linked trash; disabling a feature does not grant purge access.

For a trashed record, permanent deletion always removes its linked trashed children first, regardless of the active-record cascade toggle. Active linked children block purge: trash them first. m:n record and global document irreversible deletion remain unavailable. This safeguard also applies when another scenario exposes the same shared type or attempts to cascade-delete it. Optional strict-FK unlink can still preserve those shared documents. Configuration deletion removes a scenario, not its repository data.

Deletion removes metadata irreversibly. Repository document purge additionally attempts immediate content cleanup; ordinary deletion can queue content cleanup. Retention, repository permissions or integrity rules may reject either operation. Separate child/parent mutations are not atomic. After partial failure, a lost response or changed scope, refresh and review actual state before confirming again.

5.19. Recycle Bin view configuration

Recycle Bin reuses the scenario's existing view settings. Trashed records use the **Records** configuration. Trashed documents use the standalone **Documents** configuration, including in a Record scenario; they do not use **Documents within a record**. Changes to those views also apply in Recycle Bin. The configuration editor explains this inheritance beside the relevant settings. Recycle Bin retains separate permissions, trash-only searches and saved view preferences. There are no separate view configuration fields for trashed items.

Chapter 6. Configure litigation holds

In General / Overview, the overall Litigation switch controls hold actions and the separate Litigation-view switch. Both are disabled by default. If the selected types do not support litigation and retention, the editor locks the settings and explains the reason directly beneath the heading. Turning the overall switch off locks dependent options and permission editing but keeps their saved values for re-enablement. Map their action permissions using the existing Permissions editor. Set and remove permissions are independent for documents, records and record-scoped documents. The repository independently checks its own authorization; a scenario grant cannot override it. Users who set or remove holds also need the required repository authority in their access token. Provision this authority in the identity provider; the client does not infer it from token roles.

The additional held-item edit protection is enabled by default. Map Edit held entities to the desired roles or rights; users also need the ordinary permissions for the attempted change. Disabling this additional protection does not disable repository rules.

For strict record-document relationships without native hold inheritance, configure separate Disabled, Ask or Automatic inclusion modes for setting and removing visible child holds. Configure parent inclusion when setting a child hold and, separately, visible sibling inclusion after including that parent. All inclusion modes default to Disabled. These options do not provide automatic propagation for many-to-many relationships.

When removing an inherited document hold, the confirmation always asks users to include the source record, independently of the parent-inclusion setting for setting holds. This requires record removal permission and affects all documents inheriting the record hold.

Native hold inheritance takes precedence over scenario inclusion choices. The client updates the source and lets the repository apply inherited effects. Retention support on the entity type is required, but a populated retention date is not. No Litigation Management Service integration is required.

6.1. Understanding type settings without a separate reference

General / Type definitions shows read-only repository feature badges beside each selected record or document type. These describe repository capabilities, not grants. The feature summary and its explanations are inside the collapsible Type-definition features section. Other help appears beneath the corresponding Attributes, Content or Versions table, keeping collapsed sections compact. **Available** in the shared attribute list makes a field eligible for scenario configuration; each view and permission rule can restrict it further. **Use view-specific label** changes only the label on that surface, leaving the repository attribute and other views unchanged. Separate Content help explains the default original/rendition, optional MIME/file-name mappings, availability, downloads and annotation permissions.

The overall Recycle Bin switch also preserves saved navigation, search and lifecycle options while disabling dependent editing. Its saved permissions remain intact. QA scenarios document their

configuration and seed ownership in their descriptions. On the dashboard, hover over a scenario card or list row for one second, or focus its title with the keyboard, to read the full description in a scrollable enlarged card. Escape closes the preview.

Chapter 7. Configure retention management

In General / Overview, configure retention management, read-only table/detail display and availability of the protected-lifecycle override independently. All are disabled by default. Map document, record and record-scoped document retention-management permissions in the existing Permissions editor. First assignment after creation uses the same permission as prolongation. Map the separate protected lifecycle permission only to authorized roles; it never overrides litigation hold or grants ordinary delete/trash rights. For strict one-to-many scenarios, configure child, parent and sibling inclusion independently as Disabled, Ask or Automatic. All default to Disabled. Siblings are considered only when their parent is included. This is action-only orchestration of visible related items; creation and later linking do not synchronize dates. Native repository retention inheritance takes precedence and writes only the source record after explicit confirmation. The Retention QA category supplies separate native-inherited and configured-propagation families. QA managers can set/prolong dates; the full QA administrator separately receives protected-lifecycle privileges. Normal users cannot trash/delete entries under active retention. Every user is blocked by litigation hold. Upstream Arveo authorization remains authoritative.

Native destructive cascades into retention-capable child types are blocked when protection of unreadable targets cannot be established. A visible child listing is not proof of complete native scope; no privileged identity is substituted to discover hidden data.