

Library Open Workflows

Library Open Workflows (LibOW) is Ex Libris' workflow automation platform for Alma libraries. It is based on the workflow automation tool n8n, and has been adapted by Ex Libris to work securely within the Alma environment.

At WRLC, LibOW is managed by WRLC HQ and is used to create automated processes that handle routine tasks — such as moving data between systems, generating reports, sending notifications, or updating records — without complex custom programming. Workflows may run across the entire Network Zone or be limited to a specific institution, depending on the need.

LibOW is intended to reduce repetitive manual work and support consortium-wide operations through shared, reusable solutions. Member libraries are encouraged to propose use cases for consideration by WRLC HQ. These workflows are documented openly, and designed to be transparent about what they do and what systems they affect.

This section of the wiki serves as the central hub for LibOW at WRLC, explaining how the platform is used and documenting active workflows.

- [How to Share a Use Case with WRLC HQ](#)
- [Workflow Catalog](#)
 - [CU Law: New Borrowing Request Notification](#)
 - [Delete NZ Bib Set](#)
 - [DRAFT Scan In Consortial Barcode](#)
 - [“Identify CZ records wo inventory, not used by Network” job](#)
 - [“Identify non-CZ e-book wo inventory, not used by Network” Job](#)
 - [OCLC Job Reports](#)
 - [OCLC Multi Match List Page](#)
 - [SCF - Update Internal Note 1](#)
 - [SCF - Update Internal Note 3](#)
 - [SCF - Update Item Call Number](#)
 - [SCF - Spreadsheet of exported SCF Requests](#)

- [Trinity Students/Faculty update purge dates](#)
 - [WRLC Retention](#)
 - [Subworkflow - Add WRLC Retention](#)
 - [Subworkflow - Remove WRLC Retention](#)
 - [WRLC - Journal Volume Overlap Tool](#)
 - [WRLC - Retention Copy Analysis Tool](#)
 - [Update Item Description with Excel](#)
-
- [Resource Hub](#)

How to Share a Use Case with WRLC HQ

Overview

WRLC welcomes ideas from the community about Alma workflows and integrations that could work better, faster, or more reliably. This page explains how to share a use case with WRLC HQ using the **Community Use Case Suggestion Form**—the primary way to propose workflow improvements, automation ideas, or Alma-adjacent process changes.

You do not need to be technical to submit a use case. If something feels manual, repetitive, or harder than it should be, that's exactly the kind of feedback we want to hear.

What is this form?

The Community Use Case Suggestion Form allows WRLC community members to describe:

- Workflow issues or challenges they encounter
- Opportunities for automation or process improvement
- Ideas that could benefit one institution or multiple institutions

These submissions help WRLC:

- Identify repetitive, time-consuming, or error-prone processes
- Understand real-world pain points across institutions
- Prioritize Library Open Workflow (LibOW) automations that deliver the most impact

Why submit a use case?

When you submit a use case, you're helping to:

- Improve workflows not just for yourself, but for the broader WRLC community
- Surface shared challenges that may not be visible at the central office
- Shape where WRLC focuses its LibOW development time

Even ideas that don't turn into an immediate workflow still help inform future priorities and improvements.

Where to find it

The **WRLC Community Use Case Suggestion Form** is available directly within Alma as a **Cloud App**. This allows you to submit ideas while you are already working in Alma—no separate login or external tools are required.

Before using the form for the first time, you will need to **activate the Library Open Workflows Cloud App**.

First-Time Setup: Activating the Library Open Workflows Cloud App

If you have not previously used the Library Open Workflows Cloud App, follow these steps to activate it:

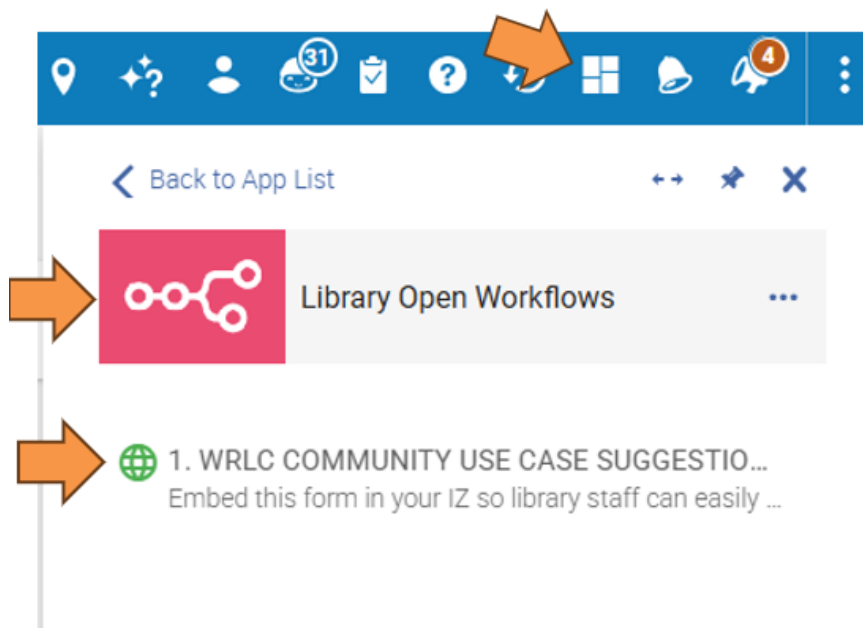
1. Open **Alma**.
2. Click the **Cloud Apps** icon in the Alma menu bar.
3. Select **Available Apps**.
4. Locate **Library Open Workflows** by either:
 - **Browsing** the list of available apps, or
 - **Searching** using the **magnifying glass** icon and typing keyword *Library*.
5. Select **Library Open Workflows** from the list.
6. Click **Activate**.
7. Click **Open**.

Once activated, the Cloud App will remain available in your **Activated Apps** list.

Accessing the Use Case Suggestion Form

After activating the Cloud App, accessing the form only takes a few clicks:

1. Open **Alma**.
2. Click the **Cloud Apps** icon.
3. Select **Library Open Workflows** from the **Activated Apps** list.
4. Choose **WRLC Community Use Case Suggestion Form**.



Who can access the form?

The Cloud App is available to Alma users with standard administrator, manager, and operator roles. For a full list of roles, [click here](#).

How to Submit a Use Case

The form asks a series of questions designed to provide WRLC HQ staff with enough context to evaluate, design, and (when possible) build a workflow. Below is what each question is asking—and why it matters:

1. Your name

So we can follow up with you if something is unclear, missing, or worth exploring further.

2. Your Institution (IZ)

Helps us understand whether an issue is local to one institution or shared across multiple IZs, which is a key prioritization factor.

3. Your Email Address

So we can follow up if something is unclear, missing, or worth exploring further. We won't spam you.

4. Summary of the Issue or Challenge

This is the high-level “what’s broken?” description. Think of it as the elevator pitch for the problem.

5. Desired Goal or Outcome

This tells us what “better” looks like to you. Sometimes the same problem can be solved in multiple ways—knowing the goal helps us design the right approach.

Example: “Automatically notify the team when a report contains a specific value.”

6. Current Workflow Description

Understanding the current steps helps us see:

- Where time is being spent
- Where errors creep in
- Which steps might be automated, simplified, or removed

If you would prefer to document this information as part of your attached file, simply note that fact here.

7. Current Workflow Frequency

How often a process happens directly affects the likely impact of an automation.

8. Stakeholders

Knowing who else benefits helps us assess broader community value—especially for workflows that could support multiple teams or roles or institutions.

9. Urgency

Urgency helps us distinguish between:

- Nice-to-have improvements
- Significant efficiency gains
- Issues that block work or create real risk

10. Process Time

Even rough estimates help us understand potential time savings and return on effort.

11. Additional Comments

This is your space to add context that didn't fit elsewhere—constraints, edge cases, or “things that always go wrong.”

12. Supporting Documentation (Optional but Very Helpful)

You may upload **one file (up to 4 MB)** to support your submission.

Helpful examples include:

- Screenshots of Alma workflows or settings
- Sample reports or exports
- Marked-up documents showing how data should look
- Step-by-step notes you already use for training or documentation

Attachments are optional, but they significantly speed up understanding and design, especially for complex workflows.

What to Expect After You Submit

Once submitted:

- Your use case is logged and reviewed by the WRLC LibOW team
- We may contact you if clarification or additional detail would help
- Submissions are prioritized based on multiple factors, including:
 - Urgency
 - Time savings or efficiency gains
 - Number of institutions or stakeholders affected
 - Feasibility and complexity
 - Alignment with current capacity and initiatives

Not all submissions will result in a workflow build, and some strong ideas may not be addressed immediately as priorities evolve. Even when a workflow is not built right away, every submission helps inform future LibOW development and highlights shared needs across the community. The team will do its best to keep the community informed of progress through direct outreach, the [WRLC Newsletter](#), and updates to the [Workflow Catalog](#).

Help or Follow-up

- Need help submitting a use case?
- Want to add additional information after submitting?
- Have questions about the status of a request?
- Experiencing issues with the form?

Please contact the [WRLC Service Desk using these instructions](#).

Workflow Catalog

This chapter provides a comprehensive catalog of automated workflows and cloud applications built by the WRLC HQ.

CU Law: New Borrowing Request Notification

Notifies CU Law staff when an ILL request is submitted via Primo.

At a glance

- **Status:** Active
- **Environment / Tags:** Live; Community Use Case
- **Applies consortium-wide?:** No
- **Runs on:** CU Law Alma production environment
- **Trigger:** Event-driven. It starts whenever Alma reports a **REQUESTS** update.
- **Primary outcome:** Sends an email to a shared staff inbox when a new interlibrary loan borrowing request is created.
- **Who receives results:** `cs1-circ@cua.edu`

Why this exists

When a patron submits an interlibrary loan request through Primo, Alma creates the borrowing request, but staff may not see it until they manually check Alma. This workflow closes that gap by sending an email notification as soon as a qualifying new borrowing request is created. For member library staff, that means faster response time and less repeated checking of Alma when nothing new has arrived.

What it does

- Listens for Alma **REQUESTS** update events.
- Checks whether the event is specifically a newly created request.
- Checks whether the request is specifically a **borrowing** request in resource sharing.
- Stops immediately if the event does not meet both conditions.
- For matching requests, looks up the patron record in Alma using the requester's primary user ID.
- Pulls the patron's full name and user group so the email is more useful to staff.
- Sends an HTML email from `Lib0W@wr1c.org` to the CU Law shared circulation inbox.
- Uses the subject line "**New ILL Request in Alma (CU LAW)**".
- Includes request details in the email body, including material type, requester, title, author, volume, issue, and part.
- Converts Alma material type codes in the email body so `BK` displays as **BOOK** and `CR` displays as **ARTICLE**.

Where it runs

- **Alma IZ(s):** CU Law production
- **Systems touched:**
 - **Alma Trigger** for request events
 - **Alma API** to retrieve patron details
 - **Alma Mail** to send the notification email
- **Reports / queries used:**
 - No Alma Analytics report is used.
 - Alma API call: `get/almaws/v1/users/{user_id}` in the **users** area.

How it works

- **If results exist:**

If the incoming Alma event says the request was created **and** the resource sharing status is `REQUEST_CREATED_BOR`, the workflow looks up the patron and sends the notification email.
- **If no results:**

If either condition is not met, the workflow follows the false branch to **No Operation, do nothing** and ends without sending any email.

Artifacts produced:

The output is an HTML email notification sent to the shared inbox.

Delete NZ Bib Set

Form accepts an NZ Set ID, displays a deletion review screen with record count, and submits Alma's Delete Bibliographic Records job (M28) to permanently remove all records in that set.

At a glance

- **Status:** Active
 - **Environment / Tags:** Network Zone; Live
 - **Applies consortium-wide?:** Yes — operates on Alma Network Zone bibliographic sets
 - **Runs on:** Alma Network Zone (NZ)
 - **Trigger:** Manual form submission (Set ID entry)
 - **Primary outcome:** Submits an Alma job to permanently delete all bibliographic records in a specified NZ set.
 - **Who receives results:** The submitting user sees an on-screen confirmation or error message (no email is sent).
-

Why this exists

Network Zone cleanup workflows often produce itemized bibliographic sets that are ready for deletion after review. Rather than requiring staff to manually navigate Alma job screens, this workflow provides a controlled, review-based interface to:

- Confirm the correct set
- Display record count
- Submit the deletion job
- Provide immediate feedback

It centralizes destructive actions behind a visible approval step.

What it does

- Presents a form requesting an **NZ Set ID**.
- Retrieves the set from Alma.
- Displays:
 - Set ID
 - Set name
 - Number of records in the set
- Warns that deletion is permanent.
- Upon confirmation, submits Alma Job **M28 (Delete Bibliographic Records)**.
- Displays a completion message.

- If retrieval fails, displays the Alma error message.
-

Where it runs

- **Alma IZ(s):**
 - Alma **Network Zone**
- **Systems touched:**
 - Alma APIs (Set retrieval and job submission)
- **Reports / queries used:** None

Credential referenced:

- **NZ Production - Alma APIs - Read & Write**
-

How it works

Logic overview

1. Form Submission

The workflow begins with the form:

Delete NZ Bibliographic Set

Required field:

- `Set ID`

The description indicates this Set ID may come from a LibOW email notifying staff that a set is ready for deletion.

2. Retrieve Set

The workflow calls:

```
GET /almaws/v1/conf/sets/{set_id}
```

If retrieval fails:

- The workflow branches to an error form.
- The Alma error message is displayed.

If retrieval succeeds:

- The workflow proceeds to a review screen.

3. Review Screen (Manual Approval Step)

The review form displays:

- Set ID
- Set Name
- Record Count (`number_of_members`)
- A prominent warning that deletion is permanent

⚠ Current safeguards:

- No typed confirmation required
- No checkbox approval
- No automatic size threshold
- No content validation

Approval is implicit by proceeding.

4. Submit Deletion Job

The workflow submits Alma Job **M28 (Delete Bibliographic Records)** with:

- `HANDLE_INVENTORY_BIBS = false`
- `HANDLE_RELATED_BIBS = false`
- `set_id = retrieved set`
- Job name:
 - `"Delete Bibliographic records - via API - {set name}"`

This deletes all bibliographic records in the set without handling inventory or related bibs.

5. Completion Message

The final screen displays:

“The set is currently being deleted in the NZ.”

No email confirmation is sent.

If results exist

- The deletion job is successfully submitted.
- Alma processes deletion asynchronously.
- User sees confirmation message.
- All bibliographic records in the specified NZ set are permanently deleted.

If no results

- If the Set ID is invalid or retrieval fails:
 - An error message is displayed.
 - No deletion job is submitted.
 - If the set contains zero records:
 - The job is still submitted.
 - Alma processes with zero deletions.
-

Artifacts produced

- Alma Job submission:
 - Job ID: M28 (Delete Bibliographic Records)
- No files, reports, or emails.
- No audit export generated by the workflow itself.

DRAFT Scan In Consortial Barcode

At a glance

- **Status:** Inactive / not yet built for production use
- **Environment / Tags:** Community Use Case
- **Applies consortium-wide?:** Yes
- **Runs on:** Member Alma institutions
- **Trigger:** Staff submit a form with an item barcode
- **Primary outcome:** Eventually, this workflow will scan in a WRLC item automatically at the circulation desk where the user is physically located.
- **Who receives results:** The staff member submitting the form sees either a success or error message.

Why this exists

This workflow is intended to make it faster and easier for member library staff to scan in consortial items. Instead of manually determining where an item belongs, staff would enter or scan a barcode into a simple form. The workflow would then identify the item and scan it in through Alma.

This workflow is different from other cataloged workflows because it is not ready for rollout yet. WRLC has requested an update to the CloudApp form API so the form submission can include the user's location. That location is needed so the item can be scanned in at the correct circulation desk automatically.

What it does

- Presents a simple form titled "**Scan in Consortial Barcode.**"
- Collects one input from the user: **Barcode.**
- Uses the barcode prefix to identify the owning WRLC institution.
- Looks up the item in Alma; uses network-level Alma logic to retrieve item information from the owning institution.
- Then accesses the user's Institution Zone in Alma; scans the item in using Alma's item scan-in API.
- Shows a success message if the scan-in step completes.
- Shows an error message if the item lookup encounters an Alma error.
- Currently uses a placeholder circulation desk value, so it cannot fully function as intended until user location is available from the form API.

Where it runs

- **Alma IZ(s):** Multiple WRLC member institutions, determined by barcode prefix and location of user
- **Systems touched:** Alma APIs
- **Reports / queries used:** None
- **Credentials referenced:**
- NZ Production - Alma APIs - Read & Write

How it works

Logic overview

- **If the barcode maps to a WRLC institution:** the workflow assigns the matching institution code and continues to item lookup.
- **If the item is found:** the workflow retrieves bibliographic, holdings, and item data, then attempts to scan in the item.
- **If scan-in succeeds:** the user sees a success screen.
- **If Alma returns an error during item lookup:** the user sees an error screen with the Alma error message.

Artifacts produced: None.

Pending dependency

Full rollout depends on a CloudApp form API update that provides user location at form access time. This location will be used to determine the circulation desk for scan-in.

“Identify CZ records wo inventory, not used by Network” job

Runs biweekly to identify the set created by the Alma NZ job “Identify records that are not used in the Network - NZ titles linked to CZ, no NZ inventory”, then emails the NZ Manager a review-and-approval link for potential deletion.

At a glance

- **Status:** Active
 - **Environment / Tags:** Network Zone; Live
 - **Applies consortium-wide?:** Yes — operates on Alma Network Zone bibliographic records
 - **Runs on:** Alma Network Zone (NZ)
 - **Trigger:** Scheduled — runs **every two weeks on Monday at 8:30 AM**
 - **Primary outcome:** Identifies a set of NZ bibliographic records linked to the Community Zone (CZ) with no NZ inventory and emails details for review and potential deletion.
 - **Who receives results:** WRLC NZ Manager
-

Why this exists

Over time, the Network Zone can accumulate bibliographic records that:

- Are linked to the Community Zone (CZ),
- Have **no Network Zone inventory**, and
- Are not used by member institutions.

These records may be candidates for cleanup. This workflow does **not delete records directly**. Instead, it:

- Detects the set created by a separate Alma identification job,
- Retrieves its details,
- Sends an email with the set ID and record count,
- Provides a link to the controlled deletion workflow.

This separates identification from deletion, creating a manual approval checkpoint.

What it does

- Runs every two weeks.
 - Looks back over the previous 13 days.
 - Retrieves instances of a specific Alma job that identifies unused NZ titles linked to CZ.
 - Checks for both:
 - COMPLETED_SUCCESS
 - COMPLETED_FAILED
 - Filters for the correct job name.
 - Locates the itemized set created by that job.
 - Retrieves full set details (including record count).
 - Sends an email with:
 - Set name
 - Set ID
 - Record count
 - Creation date
 - Direct link to the [Delete NZ Bib Set](#) approval form
 - Takes no further action unless manual deletion is approved.
-

Where it runs

- **Alma IZ(s):**
 - Alma **Network Zone**
- **Systems touched:**
 - Alma APIs (job instances and set retrieval)
 - Gmail (notification email)
- **Reports / queries used:** None

Credential referenced:

- **NZ Production - Alma APIs - Read & Write**
 - **Gmail Credentials - Generic LibOW**
-

How it works

Logic overview

1. Scheduled Trigger

- Runs every two weeks:
 - Monday
 - 8:30 AM
- Calculates:
 - Date 13 days ago
 - Today's date

- These define the job instance search window.

2. Retrieve Identification Job Instances

The workflow retrieves instances of Alma job:

S31366584960004101

Date range:

- 13 days ago → today

It explicitly checks both statuses:

- COMPLETED_SUCCESS
- COMPLETED_FAILED

This ensures no run is missed due to job outcome.

⚠ If the job ID changes, this workflow will stop functioning.

3. Filter for Correct Job

A Filter node ensures only jobs whose name contains:

“Identify records that are not used in the Network - NZ titles linked to CZ, no NZ inventory”

If Alma job naming changes, this filter may fail.

4. Retrieve Set Created by Job

The workflow:

- Retrieves job instance details.
- Searches for sets where:
 - name ~ job counter value
- Assumes:
 - Set type = ITEMIZED
 - Content type = BIB_MMS

It then retrieves full set information, including:

- Set ID
- Set name
- number_of_members
- Created date

⚠ Assumes set naming convention includes job counter value.

5. Send Notification Email

An email is sent to the WRLC Consortial NZ Manager

Email includes:

- Set Name
- Record Count
- Date
- Set ID
- Direct URL to the [Delete NZ Bib Set workflow](#) form

The email clearly states:

If no action is taken, no deletion will occur.

Deletion requires manual approval via the separate workflow.

If results exist

- A qualifying set is found.
- Email is sent with deletion approval link.
- No records are deleted automatically.

If no results

- If no job instances exist in date range:
 - No email is sent.
 - If job runs but produces no set:
 - No deletion notification occurs.
 - If job ID or naming changes:
 - Workflow may fail silently to identify sets.
-

Artifacts produced

- Email notification to NZ Manager.
 - No sets created by this workflow.
 - No deletion job submitted.
 - No files generated.
-

Relationship to other workflows

This workflow depends on:

- An Alma scheduled job that creates the identification set.

- This workflow sends notification and approval link.
- The [Delete NZ Bib Set workflow](#), which performs the actual deletion after manual approval.

This workflow only performs identification and notification.

“Identify non-CZ e-book wo inventory, not used by Network” Job

Runs biweekly to identify the set created by the Alma NZ job “Identify records that are not used in the Network - NZ Book - Electronic records without inventory, non-CZ”, then emails the NZ Manager a review-and-approval link for potential deletion.

At a glance

- **Status:** Active
 - **Environment / Tags:** Network Zone; Live
 - **Applies consortium-wide?:** Yes — operates on Alma Network Zone bibliographic records
 - **Runs on:** Alma Network Zone (NZ)
 - **Trigger:** Scheduled — runs **monthly on the 3rd day at 8:30 AM**
 - **Primary outcome:** Identifies a set of NZ non-CZ e-book bibliographic records with no NZ inventory and emails details for review and potential deletion.
 - **Who receives results:** WRLC NZ Manager
-

Why this exists

The Network Zone can accumulate electronic bibliographic records that:

- Are **not linked to the Community Zone (non-CZ)**
- Represent **e-book titles**
- Have **no Network Zone inventory**
- Are not used by member institutions

These records may be candidates for cleanup.

This workflow does **not delete records automatically**. Instead, it:

- Detects the set created by a scheduled Alma identification job,
- Retrieves set details,
- Notifies the NZ Manager,
- Provides a link to a separate, approval-based deletion workflow.

This separation ensures deletion remains a deliberate, manual action.

What it does

- Runs automatically once per month.
 - Looks back 13 days.
 - Retrieves instances of a specific Alma identification job.
 - Checks for both successful and failed job outcomes.
 - Filters for the correct job name.
 - Locates the itemized bibliographic set created by that job.
 - Retrieves full set details (ID, name, record count, created date).
 - Sends an email including:
 - Set Name
 - Record Count
 - Set ID
 - Creation date
 - Direct link to the [Delete NZ Bib Set](#) approval form.
 - Takes no deletion action unless manually approved.
-

Where it runs

- **Alma IZ(s):**
 - Alma **Network Zone**
- **Systems touched:**
 - Alma APIs (job instances and set retrieval)
 - Gmail (notification email)
- **Reports / queries used:** None

Credentials referenced:

- **NZ Production - Alma APIs - Read & Write**
 - **Gmail Credentials - Generic LibOW**
-

How it works

Logic overview

1. Scheduled Trigger

- Runs on the **3rd day of each month at 8:30 AM.**
- Calculates:
 - Date 13 days ago
 - Today's date
- These define the job instance search window.

2. Retrieve Identification Job Instances

The workflow retrieves instances of Alma job:

S31366564050004101

Date range:

- o 13 days ago → today

It checks both:

- o COMPLETED_SUCCESS
- o COMPLETED_FAILED

This ensures the workflow captures job runs regardless of outcome.

⚠ If the job ID changes, the workflow will no longer retrieve the correct instances.

3. Filter for Correct Job

A Filter node ensures only jobs whose name contains:

“Identify records that are not used in the Network - NZ Book - Electronic records without inventory, non-CZ ”

This prevents accidental processing of unrelated jobs.

⚠ If the Alma job name wording changes, the filter may fail.

4. Retrieve the Set Created by the Job

The workflow:

- o Retrieves job instance details.
- o Searches for sets where:
 - o name ~ job counter value
- o Assumes:
 - o Set type = ITEMIZED
 - o Content type = BIB_MMS

It then retrieves:

- o Set ID

- Set name
- `number_of_members`
- Creation date

⚠ This assumes the Alma job's set naming convention includes the job counter value.

5. Send Notification Email

An email is sent to the WRLC Consortial NZ Manager

The message includes:

- Set Name
- Record Count
- Creation Date
- Set ID
- Direct link to the [Delete NZ Bib Set workflow](#)

The email clearly states:

If no action is taken, no deletion will occur.

Deletion requires manual approval via the separate workflow.

If results exist

- A qualifying set is identified.
- Email is sent with set details and deletion approval link.
- No records are deleted automatically.

If no results

- If no job instances exist in the date range:
 - No email is sent.
 - If the job runs but no set is created:
 - No deletion notification is triggered.
 - If job ID or naming conventions change:
 - Workflow may fail to identify the correct set.
-

Artifacts produced

- Email notification to NZ Manager.

- No sets created by this workflow.
 - No deletion job submitted.
 - No files generated.
-

Relationship to other workflows

This workflow is part of a cleanup chain:

1. Alma scheduled job creates the identification set.
2. This workflow sends notification and approval link.
3. [Delete NZ Bib Set workflow](#) performs deletion after manual approval.

This ensures identification and deletion are separated by human review.

OCLC Job Reports

Gathers job report statistics from Network Zone import profiles (specifically OCLC Overlay import profiles) and places them into an Excel spreadsheet on a daily basis.

At a glance

- **Status:** Active
- **Environment / Tags:** Network Zone
- **Applies consortium-wide?:** Yes
- **Runs on:** Alma Network Zone (NZ)
- **Trigger:** Scheduled — runs **every day at 3:45 AM**
- **Primary outcome:** Collects statistics from completed **OCLC Daily Bib Overlay** jobs and appends them to a shared Excel tracking table.
- **Who receives results:** No direct notifications; results are written to a shared Microsoft Excel workbook.

Why this exists

WRLC staff need a reliable, low-effort way to track how daily OCLC bibliographic overlay jobs are performing over time. This workflow makes that work visible and auditable by automatically capturing key job statistics (record counts and match outcomes) and logging them in a shared spreadsheet, creating a historical record without manual data gathering.

What it does

- Runs automatically every morning.
- Looks up Alma **Network Zone import profiles**.
- Identifies active import profiles related to **OCLC Daily Bib Overlay**.
- Finds job instances from **yesterday** that completed successfully.
- Excludes FTP-specific job instances that aren't relevant to reporting.
- Extracts summary statistics (total records, match types).
- Deduplicates results so each job instance is logged once.
- Appends a new row of statistics to a shared Excel table.

Where it runs

- **Alma IZ(s):**
 - Alma **Network Zone**
- **Systems touched:**
 - **Alma APIs** — used to retrieve import profiles and job instance details
 - **Microsoft Excel (Office 365)** — used as a shared reporting log
- **Reports / queries used:**

- None (this workflow reads Alma job metadata, not Analytics reports)

How it works

Logic overview

Execution flow (plain language):

1. **Scheduled trigger**
 - The workflow starts every day at **3:45 AM**.
2. **Determine reporting date**
 - Calculates **yesterday's date** in `YYYY-MM-DD` format.
3. **Retrieve import profiles**
 - Pulls all **REPOSITORY-type import profiles** from the Alma Network Zone.
4. **Profile filtering**
 - Keeps only profiles whose name contains **"OCLC Daily Bib Overlay"**.
 - Keeps only profiles that are **active**.
5. **Retrieve job instances**
 - For each qualifying import profile:
 - Retrieves job instances submitted yesterday
 - Limits results to jobs with status **COMPLETED_SUCCESS**
6. **Instance cleanup**
 - Removes job instances whose names contain **"w/FTP"**.
 - Removes duplicate instances to avoid double-counting.
7. **Retrieve job details**
 - Fetches full details for each remaining job instance.
8. **Extract reporting fields**
 - Pulls and normalizes:
 - Job name
 - Status and status date
 - Total records processed
 - Single matches
 - Multiple matches
 - Records not matched/imported
9. **Write to Excel**
 - Appends a new row to a shared Excel table, mapping:
 - Job name → import type (New, Updated, Merged, Deleted)
 - Status date → reporting date
 - Counters → match and record totals

If results exist

- Each qualifying job instance produces **one new row** in the Excel table.
- The spreadsheet accumulates a running, date-based history of OCLC overlay activity.

If no results

- If no qualifying job instances ran yesterday:
 - The workflow completes without writing any new rows.
 - No errors or notifications are generated.

Artifacts produced

- **Excel table rows** in a shared workbook:
 - Columns include date, import type, total records, single matches, multi matches, and unmatched records.
- No emails or files are generated outside the Excel workbook.

OCLC Multi Match List Page

Automatically generates a daily Excel report of all OCLC multi-match bibliographic records from the previous day's NZ import jobs.

At a glance

- **Status:** Active
 - **Environment / Tags:** Network Zone
 - **Applies consortium-wide?:** Yes (*The workflow operates in the Alma Network Zone and reports on Network Zone import jobs.*)
 - **Runs on:** Alma Network Zone (NZ)
 - **Trigger:** Scheduled — runs **daily at 10:30 PM**
 - **Primary outcome:** Generates a spreadsheet listing all **OCLC multi-match bibliographic records** from the previous day's import jobs.
 - **Who receives results:** No direct notification; results are written to a shared Excel workbook.
-

Why this exists

When OCLC import jobs encounter **multi-matches**, Alma cannot automatically determine which existing record should be updated. These records require review.

This workflow creates a structured, review-ready spreadsheet listing all multi-match records from the previous day's OCLC import jobs, including:

- OCLC number
- Network ID
- Institution holdings
- Import profile type
- Title

This supports consortium-level quality control and faster resolution of ambiguous matches.

What it does

- Runs nightly.
- Identifies specific OCLC import jobs by job ID.
- Retrieves completed job instances from the previous day.
- Excludes FTP-related job variants.

- Retrieves all **multi-match records** from those jobs.
 - Splits comma-separated Network IDs into one row per record.
 - Retrieves full NZ bib records for each match.
 - Parses MARC XML to extract:
 - First 035\$a OCLC number
 - All AVA fields (institution holdings)
 - Produces one row per holding per bib.
 - Removes duplicate institution holdings.
 - Creates a new dated worksheet in Excel.
 - Appends formatted rows to a structured table.
-

Where it runs

- **Alma IZ(s):**
 - Alma **Network Zone**
 - **Systems touched:**
 - Alma APIs (read-only for job data and matches; read/write for bib retrieval)
 - Microsoft Excel (Office 365)
 - **Reports / queries used:** None (operates directly on Alma job APIs)
-

How it works

Logic overview

1. Schedule & Date Setup

- Runs daily at **10:30 PM**.
 - Calculates **yesterday's date**.
 - Uses that date to retrieve completed job instances.
-

2. Identify Relevant Import Jobs

- Uses a hard-coded list of OCLC import job IDs.
- Retrieves job instances submitted yesterday.
- Keeps only those with status **COMPLETED_SUCCESS**.
- Filters out instances whose name contains **"w/FTP:"**.

⚠ If import profile IDs change, this workflow must be updated.

3. Retrieve Multi-Match Records

- Calls Alma's job matches endpoint with population = `MULTI_MATCHES`.
- Extracts:
 - Incoming record ID

- MMS Ids (comma-separated list)
- Job name (used to determine import type)
- Submission date
- Determines **Import Type** (Deleted, Merged, New, Updated) using pattern matching on job name.

⚠ If naming conventions change, import type detection may break.

4. Separate Network IDs

- Splits comma-separated MMS IDs into **one row per Network ID**.
 - This is required for downstream bib-level processing.
-

5. Retrieve and Parse Bib Records

For each Network ID:

- Retrieves the full NZ bib record.
- Parses MARC XML using regex to extract:
 - First 035\$a beginning with (0CoLC)
 - All AVA fields:
 - subfield 0 → MMS Id (institution-level)
 - subfield a → Institution code

Returns one row per AVA field (institution holding).

⚠ Assumptions:

- MARC structure remains consistent.
 - Only the first (0CoLC) number is used.
 - Regex-based MARC parsing may fail if XML structure changes.
-

6. Normalize and Deduplicate

- Merges import metadata with parsed MARC data.
 - Removes duplicate rows based on MMS Id (institution holding).
 - Prepares structured fields for Excel.
-

7. Excel Output

For each run:

1. Creates a **new worksheet** named with yesterday's date (yyyy-MM-dd).
2. Waits 15 seconds (ensures Excel sheet creation completes).

3. Appends a header row.
4. Creates a table.
5. Appends all processed rows.

Columns written:

- Network Id
- OCLC Number in Alma
- Incoming Record Id
- Title
- MMS Id
- Institution
- Import Profile
- Import Date

Network IDs and MMS IDs are written as text to prevent numeric reformatting.

If results exist

- A new worksheet is created for the date.
- Each multi-match record appears as one row per institution holding.
- Spreadsheet becomes a permanent audit record for review.

If no results

- A new worksheet is still created.
 - Only the header row is present.
 - No data rows are appended.
-

Artifacts produced

- **Excel workbook:**
 - *OCLC Multi Match List*
- One worksheet per day.
- Structured table with normalized fields.

No emails or additional reports are generated.

SCF - Update Internal Note 1

A form is used to input and update the Internal Note 1 field in both the Shared Collections Facility (SCF) IZ and the IZ that owns the item.

At a glance

- **Status:** Active
- **Applies consortium-wide?:** No
- **Runs on:** SCF Production Alma environment, with conditional updates to owning Institution Zones (*Inferred from credential labels and network logic; you may want to check the workflow export.*)
- **Trigger:** Manual form submission — staff enter a barcode and tray location via a web form
- **Primary outcome:** Updates **Internal Note 1** on an item record with a tray location, in SCF and (when applicable) the owning institution's Alma IZ.
- **Who receives results:** The submitting staff member (on-screen success or error message)

Why this exists

Shared Collections Facility (SCF) staff need a fast, reliable way to record **tray location information** on physical items in both the SCF Institution Zone and the IZ of the Owning Institution. This workflow provides a simple form-based tool that updates the Alma item's *Internal Note 1* and *Item Call Number* field without requiring staff to navigate full Alma item editing screens, while also keeping owning institutions' records in sync when appropriate. However, this needs to be two separate workflows for quality control.

What it does

- Presents a simple form asking staff to enter an **item barcode**.
- Looks up the item in Alma using that barcode.
- Determines whether the item belongs only to SCF or also has an owning Institution Zone (IZ).
- Prompts the user to enter a **Tray Location**.
- Writes the tray location into the item's **Internal Note 1** field.
- Updates:
 - SCF's item record, and
 - the owning IZ's item record *when the item belongs to a WRLC member institution*.
- Displays a clear success or error message explaining what was updated.

Where it runs

- **Alma IZ(s):**
 - Shared Collections Facility (SCF) Institution Zone
 - Owinging WRLC member Institution Zones (conditional)
- **Systems touched:**
 - **Alma APIs** — used to retrieve and update item records
 - **Alma Network Zone logic** — used to identify and route updates to the owinging IZ
- **Reports / queries used:** None (no Alma Analytics usage)

How it works

Logic overview

Execution flow (plain language):

1. **Form submission**
 - Staff submit a barcode via a web form titled “*Update Internal Note 1.*”
2. **Retrieve item using barcode**
 - Alma API retrieves the item record from SCF.
 - If the barcode is invalid or not found, the workflow stops and shows an error.
3. **Owinging IZ check**
 - The workflow checks the item’s *provenance* value.
 - If the provenance matches a known WRLC member IZ code, the item is treated as having an owinging IZ.
 - Otherwise, it is treated as an SCF-only item.
4. **Tray location entry**
 - Staff are prompted to enter a **Tray Location**, with the item title shown for confirmation.
5. **Update logic**
 - The tray location is written to `item_data.internal_note_1`.
 - For SCF-only items:
 - The item is updated in SCF only.
 - For items with an owinging IZ:
 - The item is updated in SCF.
 - The workflow retrieves the owinging IZ’s item record and updates it as well.
6. **Completion**
 - A success message confirms where the update occurred.
 - If the SCF update succeeds but the owinging IZ update fails, a partial-success error message is shown.

If results exist

- The item is found and updated.
- **Internal Note 1** is overwritten with the entered tray location.
- A success message clearly states whether the update occurred:
 - in SCF only, or
 - in both SCF and the owinging Institution Zone.

If no results

- If the barcode lookup fails:
 - The workflow stops immediately.
 - The user sees a descriptive error message returned from Alma.
- No updates are made.

Artifacts produced

- **Updated Alma item records**
 - Field affected: `Internal Note 1`
- No reports, files, or emails are generated.

SCF - Update Internal Note 3

A form is used to input and update the Internal Note 3 field in the Shared Collections Facility (SCF) IZ only for local informational purposes.

At a glance

- **Status:** Active
- **Applies consortium-wide?:** No
- **Runs on:** SCF Production Alma environment
- **Trigger:** Manual form submission — staff enter a barcode and note text via a web form
- **Primary outcome:** Updates **Internal Note 3** on an item record with a tray location, in SCF and (when applicable) the owning institution's Alma IZ.
- **Who receives results:** The submitting staff member (on-screen success or error message)

Why this exists

Shared Collections Facility (SCF) staff need a fast way to put notes about the current status and whereabouts of an item in the SCF processing area. This workflow provides a simple form-based tool that allows better internal tracking.

What it does

- Presents a simple form asking staff to enter an **item barcode**.
- Looks up the item in the SCF IZ using that barcode.
- Prompts the user to enter some text (ex. Item on Refile Shelf)
- Writes the information into the item's **Internal Note 3** field.
- Updates SCF's item record
- Displays a clear success or error message explaining what was updated.

Where it runs

- **Alma IZ(s):**
 - Shared Collections Facility (SCF) Institution Zone
- **Systems touched:**
 - **Alma APIs** — used to retrieve and update item records
- **Reports / queries used:** None (no Alma Analytics usage)

How it works

Logic overview

Execution flow (plain language):

1. Form submission

- Staff submit a barcode via a web form titled *"Update Internal Note 3."*

2. Retrieve item using barcode

- Alma API retrieves the item record from SCF.
- If the barcode is invalid or not found, the workflow stops and shows an error.
-

3. Tray location entry

- Staff are prompted to enter note text, with the item title shown for confirmation.

4. Update logic

- The tray location is written to `item_data.internal_note_3` in the SCF IZ
 - The item is updated in SCF only.

5. Completion

- A success message confirms where the update occurred.

If results exist

- The item is found and updated.
- **Internal Note 3** is overwritten with the entered tray location.
- A success message clearly states whether the update occurred

If no results

- If the barcode lookup fails:
 - The workflow stops immediately.
 - The user sees a descriptive error message returned from Alma.
- No updates are made.

Artifacts produced

- **Updated Alma item records**
 - Field affected: `Internal Note 3`
- No reports, files, or emails are generated.

SCF - Update Item Call Number

A form is used to input and update the Item Call Number field in both the Shared Collections Facility (SCF) IZ and the IZ that owns the item.

At a glance

- **Status:** Active
- **Applies consortium-wide?:** No
- **Runs on:** SCF Production Alma environment, with conditional updates to owning Institution Zones (*Inferred from credential labels and network logic; you may want to check the workflow export.*)
- **Trigger:** Manual form submission — staff enter a barcode and tray location via a web form
- **Primary outcome:** Updates Item Call Number on an item record with a tray location, in SCF and (when applicable) the owning institution's Alma IZ.
- **Who receives results:** The submitting staff member (on-screen success or error message)

Why this exists

Shared Collections Facility (SCF) staff need a fast, reliable way to record **tray location information** on physical items in both the SCF Institution Zone and the IZ of the Owning Institution. This workflow provides a simple form-based tool that updates the Alma item's *Internal Note 1* and *Item Call Number* field without requiring staff to navigate full Alma item editing screens, while also keeping owning institutions' records in sync when appropriate. However this needs to be two separate workflows for quality control.

What it does

- Presents a simple form asking staff to enter an **item barcode**.
- Looks up the item in Alma using that barcode.
- Determines whether the item belongs only to SCF or also has an owning Institution Zone (IZ).
- Prompts the user to enter a **Tray Location**.
- Writes the tray location into the item's Item Call Number field.
- Updates:
 - SCF's item record, and
 - the owning IZ's item record *when the item belongs to a WRLC member institution*.
- Displays a clear success or error message explaining what was updated.

Where it runs

- **Alma IZ(s):**
 - Shared Collections Facility (SCF) Institution Zone
 - Owing WRLC member Institution Zones (conditional)
- **Systems touched:**
 - **Alma APIs** — used to retrieve and update item records
 - **Alma Network Zone logic** — used to identify and route updates to the owning IZ
- **Reports / queries used:** None (no Alma Analytics usage)

How it works

Logic overview

Execution flow (plain language):

1. **Form submission**
 - Staff submit a barcode via a web form titled “*Update Item Call Number.*”
2. **Retrieve item using barcode**
 - Alma API retrieves the item record from SCF.
 - If the barcode is invalid or not found, the workflow stops and shows an error.
3. **Owning IZ check**
 - The workflow checks the item’s *provenance* value.
 - If the provenance matches a known WRLC member IZ code, the item is treated as having an owning IZ.
 - Otherwise, it is treated as an SCF-only item.
4. **Tray location entry**
 - Staff are prompted to enter a **Tray Location**, with the item title shown for confirmation.
5. **Update logic**
 - The tray location is written to `item_data.alternative_call_number`.
 - ****note** - item call number field was originally called alternative call number in early Alma releases and that name persists in the data library
 - For SCF-only items:
 - The item is updated in SCF only.
 - For items with an owning IZ:
 - The item is updated in SCF.
 - The workflow retrieves the owning IZ’s item record and updates it as well.
6. **Completion**
 - A success message confirms where the update occurred.
 - If the SCF update succeeds but the owning IZ update fails, a partial-success error message is shown.

If results exist

- The item is found and updated.
- **Item Call Number** is overwritten with the entered tray location.
- A success message clearly states whether the update occurred:
 - in SCF only, or

- in both SCF and the owning Institution Zone.

If no results

- If the barcode lookup fails:
 - The workflow stops immediately.
 - The user sees a descriptive error message returned from Alma.
- No updates are made.

Artifacts produced

- **Updated Alma item records**
 - Field affected: `Item Call Number`
- No reports, files, or emails are generated.

SCF - Spreadsheet of exported SCF Requests

At a glance

- **Status:** Active
- **Environment / Tags:** Live; SCF Workflow
- **Applies consortium-wide?:** Shared Collections Facility IZ only
- **Runs on:** Alma APIs
- **Trigger:** Staff submit an authenticated form
- **Primary outcome:** Creates and emails a spreadsheet of recent SCF request exports.
- **Who receives results:** The email address entered in the form

Why this exists

This workflow lets staff quickly receive a spreadsheet of recent requests for SCF materials without manually checking multiple SFTP folders or parsing Alma request export files. It supports visibility into requests sent to SCF and helps staff confirm what Alma processed during recent “Send Requests to Remote Storage” job runs.

What it does

This workflow lets staff request a spreadsheet of recent SCF material requests. A staff member selects a time period and enters an email address. The workflow then looks for SCF request files from that period, turns the request data into a spreadsheet, and emails it to the requester.

If no recent request files are found, the workflow tells the user that no requests were found instead of sending a blank spreadsheet.

Where it runs

- **Alma IZ(s):** SCF IZ
- **Systems touched:**
 - n8n form
 - WRLC SFTP server
 - Alma remote storage request export files
 - Gmail
- **Credentials referenced:**
 - WRLC SFTP server
 - Gmail Credentials - Generic Lib0W

How it works

Logic overview

1. A staff member submits the **WRLC SCF Requests** form.
2. The workflow reads the selected time period.
3. It calculates a cutoff timestamp based on that time period.
4. It formats today's date for use in the email subject.
5. It creates a list of SCF SFTP request folders to check.
6. It lists files in each folder.
7. It filters the results to keep only recent files modified after the cutoff time.
8. The workflow checks whether any matching files were found.
9. If files were found:
 - It downloads the request files from SFTP.
 - It extracts XML content from the files.
 - It transforms the XML into request text.
 - It parses that text into structured request fields.
 - It converts the parsed data into an Excel spreadsheet.
 - It emails the spreadsheet to the address submitted in the form.
 - It displays a success message.
10. If no files were found:
 - It displays a "No requests found" message.
 - It explains that either no requests were made or Alma's remote storage jobs may not have run as scheduled.

Artifacts produced

- **Spreadsheet attachment:** SCF Requests
- **Email subject pattern:** Incoming SCF Requests yyyy-MM-dd
- **Email recipient:** The address entered in the form

Trinity Students/Faculty update purge dates

Creates a set of non-expired patron accounts for Trinity University faculty and students in the Shared Collections Facility (SCF) IZ and updates their purge date to a date 18 months in the future.

At a glance

- **Status:** Active
- **Environment / Tags:** SCF Workflows; Live
- **Applies consortium-wide?:** No
- **Runs on:** SCF Production Alma environment
- **Trigger:** Scheduled — runs at **5:30 AM on the 1st day, every 4 months**
- **Primary outcome:** Builds a current logical set of Trinity faculty/students and runs an Alma job to **set a purge date 18 months out** for those users.
- **Who receives results:** No direct distribution (no email/report delivery in workflow export)

Why this exists

Libraries often need user accounts to remain available for active patrons, while ensuring inactive/expired accounts are queued for cleanup according to policy. This workflow automates a repeatable “maintenance” step: it identifies current Trinity faculty/students (based on user group and expiry date) and sets a future purge date, reducing manual intervention and helping keep user data lifecycle rules consistent.

What it does

- Runs on a schedule (every four months).
- Captures **today's date** in `YYYY-MM-DD` format.
- Creates an Alma **logical set** named `Current Trinity Faculty and Students <YYYY-MM-DD>`.
- The set's query targets Alma user records where:
 - user group is **Trinity Faculty** or **Trinity Student** (`tr fac` / `tr stud`), **and**
 - **expiry date is after today.**
- Calculates a date **18 months from today.**
- Converts that 18-months-from-now date into a **Unix millisecond timestamp.**
- Runs an Alma “Update/Notify Users – via API” job (job id `M148`) against the created set.
- Sets the job name dynamically to:
`Update/Notify Users - via API - <set name>`
- Updates **only** the user **Purge Date** (enabled), leaving other update options disabled.

Where it runs

- **Alma IZ(s):** Shared Collections Facility (SCF) Institution Zone
- **Systems touched:**
 - **Alma APIs** (Ex Libris Alma REST APIs) — used here to create a logical set and run a configured Alma job.
- **Reports / queries used:**
 - Alma logical set query (embedded in the set creation):
 - `USER where USER ((user_group OUTER_EQUAL (tr fac or tr stud)) AND USER (expiry_date AFTER <today>))`

How it works

Logic overview

Execution order (nodes):

1. **Schedule Trigger**
 - Role: Starts the workflow on a timed schedule.
 - Key behavior: “Runs at 5:30am on day 1 every 4 months.”
2. **Today’s Date (Date & Time → formatDate)**
 - Role: Produces today’s date formatted as `yyyy-MM-dd`.
 - Output used in the set name and set query.
3. **Create Logical Set (Alma API → POST /almaws/v1/conf/sets)**
 - Role: Creates a logical set of current Trinity faculty/students.
 - Key configuration:
 - Set name pattern: `Current Trinity Faculty and Students <today>`
 - Type: `LOGICAL`
 - Content: `USER`
 - Query: Trinity faculty/student user groups with expiry date after today
 - **Credential label referenced:** `SCF Production - Alma APIs - Read & Write`
4. **Date in 18 Months (Date & Time → addToDate)**
 - Role: Computes a date 18 months after today.
5. **Unix Ms Date (Date & Time → formatDate)**
 - Role: Converts the 18-month future date into a Unix millisecond timestamp (`x`).
 - Output feeds the purge date value used by the Alma job.
6. **Run Job (Alma API → POST /almaws/v1/conf/jobs/{job_id} with op=run, job_id=M148)**
 - Role: Runs an Alma job against the logical set, updating user fields.
 - Key configuration that affects behavior:
 - Targets `set_id` from “Create Logical Set”
 - **Purge Date update is enabled** (`UPDATE_User_Information_Purge_Date_Active = true`)
 - Purge date value is the Unix ms timestamp for “today + 18 months”
 - Most other update options are explicitly disabled (e.g., status/user group/roles/notifications are not activated)

If results exist

- Alma creates the logical set with matching users, then runs the job on that set.
- For each user in the set, the job updates the **Purge Date to 18 months from the run date** (expressed as a Unix ms timestamp in the job parameters).

If no results

- The workflow still creates the logical set and runs the Alma job.
- Expected behavior is that the job completes with **0 affected records** (no special branching, alerts, or alternate path in the workflow export).

Artifacts produced

- **Alma logical set** created each run:
 - Naming pattern: `Current Trinity Faculty and Students <YYYY-MM-DD>`
- No files, emailed outputs, or stored reports are produced by this workflow

WRLC Retention

Processes a user-uploaded CSV of barcodes to bulk add or remove WRLC retention commitments across institutions, then emails a summary of results.

At a glance

- **Status:** Active
 - **Applies consortium-wide?:** Yes — processes barcodes across multiple Alma Institution Zones via subworkflows
 - **Runs on:** Alma Institution Zones (via Network routing in subworkflows)
 - **Trigger:** Manual form submission with CSV upload
 - **Primary outcome:** Bulk adds or removes WRLC retention flags on item records.
 - **Who receives results:** The email address provided in the form submission.
-

Why this exists

WRLC retention commitments must be applied or removed at scale when retention decisions change. Performing this manually in Alma would require record-by-record updates across multiple institutions.

This workflow enables controlled bulk processing by:

- Accepting a structured CSV upload
- Confirming the number of records before execution
- Calling standardized subworkflows for each barcode
- Emailing a clear results summary

It centralizes retention management while preserving audit visibility.

What it does

- Presents a web form allowing staff to:
 - Choose **Add WRLC Retention** or **Remove WRLC Retention**
 - Upload a CSV file (maximum 4,000 barcodes)
 - Provide an email address for results
- Counts rows in the uploaded file.
- Displays a confirmation screen showing:
 - Number of barcodes detected
 - Whether retention will be added or removed
- After confirmation:

- Processes each row individually.
 - Calls the appropriate subworkflow:
 - [Subworkflow - Add WRLC Retention](#)
 - [Subworkflow - Remove WRLC Retention](#)
 - Aggregates results (successes and errors).
 - Sends a formatted summary email.
-

Where it runs

- **Alma IZ(s):**
 - Dynamically determined per row using the “Institution Code” column in the CSV.
 - **Systems touched:**
 - Alma APIs (via subworkflows)
 - Email (sent from `reports@wrlc.org`)
 - **Reports / queries used:** None
-

How it works

Logic overview

1. Form Submission

The workflow begins with a form requiring:

- Dropdown selection:
 - Add WRLC Retention
 - Remove WRLC Retention
- CSV upload (must contain $\leq 4,000$ rows)
- Email address

Required CSV format:

Barcode	Institution Code
----------------	-------------------------

⚠ The workflow does **not validate header names**; it assumes:

- First row is header.
 - First column = Barcode.
 - Second column = Institution Code.
-

2. Row Counting & Review Step

Before processing:

- The uploaded file is decoded from base64.
- Blank lines are removed.
- Header row is subtracted from count.
- A confirmation screen displays:
 - Number of barcodes detected.
 - Whether retention will be added or removed.

This confirmation step prevents accidental bulk updates.

3. Branch Based on User Selection

An **If** node evaluates the dropdown choice:

- If **Add WRLC Retention** → calls:
 - Subworkflow – Add WRLC Retention
- Otherwise → calls:
 - Subworkflow – Remove WRLC Retention

Each CSV row triggers one subworkflow execution (mode: each).

⚠ Errors are set to `continueRegularOutput`, meaning:

- Processing continues even if some rows fail.
 - Failures are captured and summarized later.
-

4. Subworkflow Processing

Each row:

- Routes to the correct Institution Zone.
- Retrieves the item by barcode.
- Adds or removes WRLC retention fields.
- Returns success or error information.

(Full logic documented in the respective subworkflow entries, [Add WRLC Retention](#) and [Remove WRLC Retention](#))

5. Aggregate Results

After all rows are processed:

The **Summarize results** node:

- Counts total processed rows.
- Counts successful updates.
- Collects error details.
- Attempts multiple barcode fallbacks:
 - `_inputBarcode`
 - `item_data.barcode`
 - `Barcode`

Errors are retained even if barcode is missing.

6. Email Results

Sends an HTML-formatted email containing:

- Total processed
- Successful updates
- Error count
- Detailed error list (if any)

From address: `reports@wrlc.org`

Recipient: Email provided in form

If results exist

- Each barcode is processed independently.
- Successful updates are counted.
- Errors are listed individually.
- User receives detailed summary email.

If no results

- If file contains zero data rows:
 - Processing completes immediately.
 - Email reflects zero processed.
 - If all rows fail:
 - Email lists all errors.
 - Workflow does not halt due to individual row failures.
-

Artifacts produced

- Updated Alma item records (via subworkflows).
- Summary email to submitting user.
- No files, sets, or spreadsheets are created.

Subworkflow – Add WRLC Retention

This subworkflow applies a WRLC retention commitment to a specified item by setting the retention-related fields in Alma for a given barcode and institution.

At a glance

- **Status:** Inactive ([parent workflow](#) is Active)
 - **Applies consortium-wide?:** Yes
 - **Runs on:** Alma Institution Zones via Network routing
 - **Trigger:** Executed by another workflow (subworkflow only; no independent trigger)
 - **Primary outcome:** Marks an item as committed to WRLC retention in Alma.
 - **Who receives results:** No direct output; results are returned to the calling workflow.
-

Why this exists

WRLC retention commitments are recorded at the item level in Alma using structured retention fields. When an item is designated for WRLC retention, those fields must be set consistently across institutions.

This subworkflow standardizes how retention commitments are applied, ensuring that parent workflows can add WRLC retention flags in a consistent and reusable way without duplicating logic.

What it does

- Receives input from a [parent workflow](#).
 - Preserves the incoming barcode and institution code for reporting.
 - Routes the API call to the correct Alma Institution Zone.
 - Retrieves the item by barcode.
 - Sets WRLC retention-related fields.
 - Writes the updated item back to Alma.
 - Returns the updated item data to the calling workflow.
-

Where it runs

- **Alma IZ(s):**
 - Dynamically determined via `_institutionCode` passed from the parent workflow.

- **Systems touched:**
 - Alma APIs (item retrieval and update)
 - Alma Network routing node
 - **Reports / queries used:** None
-

How it works

Logic overview

1. Trigger (Subworkflow Execution)

- Starts via **“When Executed by Another Workflow.”**
- Requires input fields:
 - `Barcode`
 - `Institution Code`

If these fields are missing, the workflow may fail or route incorrectly.

2. Preserve Input

The **“Stash barcode”** code node:

- Copies:
 - `Barcode`
 - `Institution Code`
- Stores them internally as:
 - `_inputBarcode`
 - `_institutionCode`

This ensures the calling workflow can reference the original input values for error handling or reporting.

3. Route to Correct Institution

The **Alma Network** node:

- Uses `_institutionCode` to dynamically select the appropriate Institution Zone.
- Does not target all network members.

⚠ If `_institutionCode` is incorrect or missing, the API call may fail or affect the wrong IZ.

4. Retrieve Item

- Calls Alma API:

- `GET /almaws/v1/items`
 - Uses the provided barcode.
 - Always outputs data (even on error), allowing the parent workflow to handle failures.
-

5. Apply WRLC Retention Fields

The **“Add WRLC Retention Code”** node updates:

- `item_data.committed_to_retain`
 - Sets `value = true`
 - Sets `desc = "Yes"`
- `item_data.retention_reason`
 - Sets `value = WRLCRetentionDONOTDELETE`
 - Sets `desc = "WRLC Retention"`
- `item_data.retention_note`
 - Sets to empty string

⚠ Important behaviors:

- This workflow **does not validate** whether the item is already marked as retained.
- It forcibly sets the retention-related fields.
- It assumes the Alma item structure includes:
 - `item_data.committed_to_retain`
 - `item_data.retention_reason`

If Alma’s API structure changes, this node may fail.

6. Update Item

- Calls Alma API:
 - `PUT /almaws/v1/bibs/{mms_id}/holdings/{holding_id}/items/{item_pid}`
 - Writes the modified item back to Alma.
-

If results exist

- The item is marked as committed to WRLC retention.
- Retention reason is set to **WRLCRetentionDONOTDELETE (WRLC Retention)**.
- Updated item JSON is returned to the parent workflow.
- No independent notifications are sent.

If no results

- If the barcode is invalid or retrieval fails:
 - The workflow still outputs data (due to `continueRegularOutput`).
 - Error handling must be managed by the parent workflow.

Artifacts produced

- Updated Alma item record with WRLC retention applied.
- No sets, reports, spreadsheets, or email notifications.

Subworkflow – Remove WRLC Retention

This subworkflow removes WRLC retention commitments from a specified item by clearing retention-related fields in Alma for a given barcode and institution.

At a glance

- **Status:** Inactive ([parent workflows](#) are Active)
 - **Applies consortium-wide?:** Yes (
 - **Runs on:** Alma Institution Zones via Network routing
 - **Trigger:** Executed by another workflow (subworkflow only; no independent trigger)
 - **Primary outcome:** Clears WRLC retention indicators from an item record in Alma.
 - **Who receives results:** No direct output; results are returned to the calling workflow.
-

Why this exists

WRLC retention commitments are recorded at the item level in Alma. When a retention commitment needs to be removed (for example, as part of a broader retention management process), this subworkflow standardizes how those fields are cleared.

By isolating this logic into a reusable subworkflow, parent workflows can remove retention data consistently across institutions without duplicating code.

What it does

- Receives input from a [parent workflow](#).
 - Preserves the incoming barcode and institution code for error reporting.
 - Routes the API call to the correct Alma Institution Zone.
 - Retrieves the item by barcode.
 - Clears WRLC retention-related fields.
 - Writes the updated item back to Alma.
 - Returns the updated item data to the calling workflow.
-

Where it runs

- **Alma IZ(s):**
 - Dynamically determined via `_institutionCode` passed from the parent workflow.

- **Systems touched:**
 - Alma APIs (item retrieval and update)
 - Alma Network routing node
 - **Reports / queries used:** None
-

How it works

Logic overview

1. Trigger (Subworkflow Execution)

- Starts via **“When Executed by Another Workflow.”**
- Requires input fields:
 - Barcode
 - Institution Code

If these fields are missing, the workflow will fail or route incorrectly.

2. Preserve Input

The **“Stash barcode”** code node:

- Copies:
 - Barcode
 - Institution Code
- Stores them internally as:
 - `_inputBarcode`
 - `_institutionCode`

This ensures error reporting in the parent workflow can reference the original input values.

3. Route to Correct Institution

The **Alma Network** node:

- Uses `_institutionCode` to dynamically select the appropriate Institution Zone.
- Does not target all network members.

⚠ If `_institutionCode` is incorrect or missing, the update may fail or affect the wrong IZ.

4. Retrieve Item

- Calls Alma API:
 - `GET /almaws/v1/items`

- Uses the provided barcode.
 - Always outputs data (even on error), allowing parent workflow error handling.
-

5. Remove WRLC Retention Fields

The **“Remove WRLC Retention Code”** node modifies:

- `item_data.committed_to_retain`
 - Sets `value = false`
 - Sets `desc = "No"`
- `item_data.retention_reason`
 - Clears value and description
- `item_data.retention_note`
 - Clears value

⚠ Important behaviors:

- This workflow **does not validate** whether the item is currently marked as WRLC retention.
- It forcibly clears the fields.
- It assumes the Alma item structure includes:
 - `item_data.committed_to_retain`
 - `item_data.retention_reason`

If Alma’s API structure changes, this logic may break.

6. Update Item

- Calls Alma API:
 - `PUT /almaws/v1/bibs/{mms_id}/holdings/{holding_id}/items/{item_pid}`
 - Writes the modified item back to Alma.
-

If results exist

- The item’s retention fields are cleared.
- The updated item JSON is returned to the parent workflow.
- No independent notification is sent.

If no results

- If the barcode is invalid or the item cannot be retrieved:
 - The workflow still outputs data (due to `continueRegularOutput` settings).
 - Error handling must be managed by the parent workflow.
-

Artifacts produced

- Updated Alma item record.
- No files, sets, reports, or email notifications.

WRLC - Journal Volume Overlap Tool

At a glance

- **Status:** Active
 - **Environment / Tags:** Analytics
 - **Applies consortium-wide?:** Yes
 - **Runs on:** Alma Network Zone (NZ) and Shared Storage Facility (SCF) Alma environments
 - **Trigger:** Manual web form—runs when a user submits a barcode *or* ISSN
 - **Primary outcome:** Displays all SCF-held volumes for a given journal title based on barcode or ISSN lookup
 - **Who receives results:** The user who submitted the form (interactive display in browser)
-

Why this exists

When processing journal volumes for accession into the Shared Collections Facility (SCF), staff need to quickly determine whether volumes of the same title are already held.

This workflow provides a fast, self-service lookup tool that eliminates the need for manual searching across Alma. It helps staff:

- Avoid duplicating volumes already held at SCF
 - Make informed retention and accession decisions
 - Save time during intake and processing
-

What it does

- Presents a web form asking the user to enter either:
 - A **barcode**, or
 - An **ISSN** (but not both)
- Checks whether a barcode was entered
- If **no barcode is provided**, treats the request as an ISSN lookup
- If **a barcode is provided**, performs a barcode-based lookup first
- Queries Alma Analytics reports to retrieve matching SCF holdings

- Uses two different Analytics queries depending on input type
 - Collects results including: ISSN, title, barcode, call number, description, and provenance
 - Formats results into a simple HTML table
 - Displays results directly in the browser
 - Shows a clear message if the lookup cannot be performed (e.g., missing ISSN)
-

Where it runs

- **Alma IZ(s):**
 - Network Zone (NZ)
 - Shared Storage Facility (SCF) institution
 - **Systems touched:**
 - Alma Analytics (reporting system used to query library data)
 - n8n web form interface
 - **Reports / queries used:**
 - /shared/Washington Research Library Consortium (WRLC) Network/Reports/CannedReports
 - *VOLUMES HELD AT SCF - Barcode Lookup*
 - /shared/Shared storage institution/Reports/SCF Reports
 - *VOLUMES HELD AT SCF - ISSN*
-

How it works

Logic overview

1. User submits the form with either a barcode or ISSN
2. Workflow checks whether the barcode field is empty
3. Branches into one of two paths:

Barcode path

- Runs **Barcode Lookup** report in the Network Zone
- Then runs **ISSN Lookup** in SCF (to retrieve all volumes for the same title)
- Formats results into an HTML table
- Displays results

ISSN path

- Runs **ISSN Lookup** directly in SCF
- Formats results into an HTML table
- Displays results

If results exist

- All matching SCF volumes are displayed in a table with:
 - ISSN
 - Title
 - Barcode
 - Item call number
 - Description (e.g., volume/issue info)
 - Provenance code
- A count of total results is shown at the top

If no results

- The results table displays a row stating:
“No results found.”
- If the lookup fails due to missing or invalid ISSN:
 - User sees an error message:

“It appears your item did not have a valid ISSN for lookup. You may need to do a more detailed Alma search.”

Artifacts produced

- **No files generated**
- Output is a **dynamic HTML results page** displayed immediately to the user
- Includes a restart button to run another search

WRLC - Retention Copy Analysis Tool

At a glance

- **Status:** Active
 - **Environment / Tags:** Analytics
 - **Applies consortium-wide?:** Yes (Inferred)
 - **Runs on:** Alma Network Zone (NZ) (Inferred)
 - **Trigger:** Manual web form—runs when a user submits a barcode
 - **Primary outcome:** Displays all copies of a title (same Network ID) along with retention and processing details
 - **Who receives results:** The user who submitted the form (interactive display in browser)
-

Why this exists

When evaluating retention commitments across the consortium, staff need to understand how many copies of a title exist and which institution is responsible for retaining them.

This workflow provides a quick way to analyze all copies tied to the same **Network ID** (i.e., the same bibliographic record or edition). It helps staff:

- Identify which institution has committed to retain a title
 - Review retention decisions before making changes
 - Understand duplication across partner libraries
 - Support retention reassignment workflows
-

What it does

- Presents a web form asking the user to enter a **barcode**
- Uses the barcode to look up the associated item in Alma Analytics
- Extracts the **Network ID** (shared bibliographic identifier) for that item
- Uses the Network ID to retrieve **all matching copies** across the consortium
- Pulls detailed data for each copy, including:
 - Title

- Barcode
 - Process type (e.g., item status)
 - Retention commitment status
 - Retention reason
 - Call number
 - Fulfillment note
 - Institution name
 - Formats results into a table for easy review
 - Displays the results directly in the browser
 - Shows an error message if no valid Network ID is found
-

Where it runs

- **Alma IZ(s):**
 - Network Zone (NZ)
 - **Systems touched:**
 - Alma Analytics (reporting system used to query library data)
 - n8n web form interface
 - **Reports / queries used:**
 - `/shared/Washington Research Library Consortium (WRLC) Network/Reports/CannedReports`
 - *Retention Reassignment Form - Barcode Lookup*
 - *Retention Reassignment Form*
-

How it works

Logic overview

1. User submits a barcode via the form
 2. Workflow runs **Barcode Lookup** report to retrieve item data
 3. Extracts the **Network ID** from the result
 4. Runs a second Analytics query using that Network ID to find all related copies
 5. Branches based on whether the Network ID lookup succeeds
-

If results exist

- All matching copies (same Network ID) are displayed in a table
- Each row includes:
 - Title

- Barcode
 - Process type
 - Retention commitment (Yes/No)
 - Retention reason
 - Call number
 - Fulfillment note
 - Owning institution
 - A total result count is shown at the top
-

If no results

- If the Network ID cannot be found or is invalid:
 - User sees an error message:

“Your item does not appear to have a valid network ID. You may need to do a more detailed search in Alma.”

- If no matching copies are returned:
 - The table displays:
“No results found.”
-

Artifacts produced

- **No files generated**
- Output is a **dynamic HTML results page** displayed immediately to the user
- Includes a restart option to run another search

Update Item Description with Excel

At a glance

- **Status:** Active
- **Environment / Tags:** Network Zone
- **Applies consortium-wide?:** No — the workflow processes one selected Institution Zone per run
- **Runs on:** One Alma Institution Zone chosen by the user at submission time
- **Trigger:** Manual form submission with CSV upload
- **Primary outcome:** Bulk updates item description, enumeration, and chronology fields on Alma item records from a CSV file.
- **Who receives results:** The email address provided in the form submission

Why this exists

Staff sometimes need to update item description fields in bulk, especially for serials, multipart items, or other records where description, enumeration, and chronology need to be corrected or added consistently. Doing this one item at a time in Alma is slow and error-prone. This workflow provides staff at HQ an upload form that applies those updates via the Network Zone in bulk for a selected Institution Zone, and then emails a summary of the results.

What it does

- Presents a form where staff choose an **Institution Zone**, upload a **CSV file**, and enter an **email address**.
- Accepts CSV files with up to **4,000 barcodes**.
- Counts the number of data rows in the uploaded file.
- Shows a confirmation screen before processing begins.
- Routes processing to the selected Institution Zone.
- Retrieves each item by barcode.
- Merges the Alma item record with the spreadsheet row.
- Updates item fields from the CSV:
 - Description
 - Enumeration A-H
 - Chronology I-M
- Blanks those Alma fields if a corresponding spreadsheet column is missing or empty.
- Writes the updated item back to Alma.
- Summarizes successes and errors.

- Emails the submitting user a results summary.

Where it runs

- **Alma IZ(s):** Select one of the 16 IZs in our Network Zone
- **Systems touched:**
 - Alma APIs
 - Alma Network routing
 - Email via Alma letter API
- **Reports / queries used:** None

How it works

Logic overview

1. Form submission

The workflow begins with a form titled **Update Item Record with Excel**.

Required inputs:

- **Institution Zone** dropdown
- **Upload file of barcodes** (.csv)
- **Email**

The form description specifies that the CSV should contain:

- first column header: Barcode
- remaining headers: Description, Enum A - H, Chron I - M

2. Pre-processing and confirmation

The workflow reads the uploaded CSV and counts non-blank lines, subtracting the header row to calculate the number of barcodes. It then shows a review screen telling the user how many barcodes were detected and that the selected item records will be updated in Alma.

This is a confirmation step only. It does not validate the CSV headers beyond assuming the first row is a header row.

3. Extract CSV rows

A separate branch extracts the CSV contents into rows for processing. This branch is merged back after the user confirms submission.

4. Route to the selected Institution Zone

After confirmation, the workflow uses the selected **Institution Zone** value to route API calls through the Alma Network node to the correct institution.

5. Retrieve each item by barcode

For each CSV row, the workflow retrieves the Alma item record using the barcode in that row.

6. Build the update payload

The workflow combines the spreadsheet row with the retrieved Alma item record, then updates these Alma item fields from the spreadsheet columns:

- `description` from **Description**
- `enumeration_a` through `enumeration_h` from **Enum A** through **Enum H**
- `chronology_i` through `chronology_m` from **Chron I** through **Chron M**

Important behavior:

- Spreadsheet-only columns are removed from the payload before the item is sent back to Alma.
- If a column is present but blank, the Alma field is set to blank.
- If a column is missing entirely, the Alma field is also set to blank.

This means the workflow is not just adding values; it can also clear existing Description / Enum / Chron values.

7. Update Alma item records

The workflow sends the modified item record back to Alma using the item update endpoint.

8. Summarize results and email the user

After all item updates complete, the workflow counts:

- total processed
- successful updates
- errors

It then emails the submitting user a summary report.

If results exist

- Matching item records are updated in the selected Institution Zone.
- Description / enumeration / chronology values are overwritten from the CSV.
- The user receives an email summarizing how many records were processed successfully and how many errors occurred.

If no results

- If a barcode cannot be retrieved or an update fails, processing continues for the remaining rows.

- Errors are captured and included in the results summary.
- The workflow does not stop on individual row failures.

Artifacts produced

- Updated Alma item records
- Email summary sent to the submitting user
- No spreadsheets, sets, or reports are created by the workflow itself

Resource Hub

This page brings together materials to support learning, adoption, and ongoing use of LibOW across WRLC.

- [Expanding Services for Alma - WRLC Town Hall Presentation - April 2026.pdf](#)
- [Ex Libris Developer Network: LibOW Blogs](#)
- [LibOW Release Notes](#)
- [LibOW Nodes](#)