

# CLAW IR MVP

## [Document Summary](#)

## [Phases of CLAW IR MVP Development](#)

[Phase 0: Gather Feature Requests from Islandora IR Community & Codify into CLAW IR MVP Document](#)

[Phase 1: Begin Work on Implementing CLAW IR MVP](#)

[Phase 2: Post-MVP & Post-Migration](#)

## [CLAW IR MVP Required Features](#)

[Types of Objects](#)

[PDF Objects](#)

[Compound Objects](#)

[Binary Objects](#)

[Embargoes](#)

[Usage Stats](#)

[Customizeable Submission Workflows](#)

[Google Scholar & SEO](#)

## Document Summary

This document represents an agreement between the various members of the Islandora IR user community about the minimum set of required functionalities that must be present in CLAW in order for existing Islandora 7.x-1.x IR users to feel comfortable migrating.

## Phases of CLAW IR MVP Development

The following phases are major milestones for IR development in the CLAW project.

### Phase 0: Gather Feature Requests from Islandora IR Community & Codify into CLAW IR MVP Document

- Get MVP functional requirements from:
  - [CLAW IR MVP GitHub Issues thread](#)
  - Islandoracon Hack/Doc Scholar MVP track
  - Feedback on CLAW IR MVP document

- Create rough draft of CLAW IR MVP doc & present to Islandora IR community for feedback, incorporate feedback & represent to IR community iteratively
- Present CLAW IR MVP document to CLAW community & get feedback
- Create final draft of CLAW IR MVP doc & present to Roadmap Committee

## Phase 1: Begin Work on Implementing CLAW IR MVP

- Build required features into CLAW & provide demos to community for feedback
- Help create tools & migration workflows/documentation
- Work with IR community to begin migration work

## Phase 2: Post-MVP & Post-Migration

- Begin building of nice-to-have features, rather than have-to-have features

# CLAW IR MVP Required Features

## Types of Objects

### PDF Objects

CLAW must be able to support PDF objects that make use of attached PDFs by allowing preview reading, creating derivatives of the front page, and full-text extraction. PDFs are the primary file format for institutional repositories, so this is a foundational requirement.

### Compound Objects

CLAW must be able to support the attachment of an object to another object (parent object to child object) in order to support complex object structures, like a journal article with a supplementary data set. While PCDM Objects support this use, the CLAW UI will have to enable the attachment, detachment, and re-ordering of compound objects in order for this requirement to be satisfied. Compound objects should also be able to be created both via batch ingest and web forms.

### Binary Objects

Binary objects in Islandora 7.x-1.x work well for research data in various formats that fall outside the standard object types. CLAW should support the use of binary objects that make no assumptions about the type of binary file being attached, but allow for download of these objects.

## Embargoes

CLAW must be able to support embargoes that apply access-restriction rules to PCDM Collections, Objects and Files. Embargoes must allow for whitelisted IP ranges that can bypass an embargo, automatic expiration of embargoes, and automatic email notifications for expiration notices and warning. There should also be an “embargo log” that notes when objects are embargoed and when embargoes are lifted, so that users can check for objects when embargoes were lifted prematurely.

Embargoed objects may also need to have “tombstones” in feeds, but this feature may or may not be a requirement and requires further exploration.

## Usage Stats

Many faculty members see the usage reports as the primary benefit of submitting materials into an IR. Many scholarly communications librarians and repository managers also depend on having this information available for reporting to organizations like ARL. With this in mind, making sure that CLAW tracks usage of PCDM Collections, Objects, and Files is of the utmost importance.

With the popularity of aggressive caching increasing in the Drupal community, CLAW's method of collection for usage statistics should not be collected on the backend as it was in Islandora 7.x-1.x. Stats should be collected via JavaScript embedded in pages, with Piwik being the primary choice due to its privacy policy and ability to be used in multiple countries that do not allow the use of Google Scholar.

Users of CLAW should also be able to aggregate views/downloads of a Collection's children into a report, as well as create exportable/emailable custom reports to track things like uploads over time, growth in GB over time, etc. Since this functionality is not available in Islandora 7.x-1.x and will require significant work, it is not a functional requirement of the MVP, but it is included because the need for this feature should inform the design of CLAW's usage stats solution.

## Customizable Submission Workflows

CLAW should support customizable submission workflows for mediated and unmediated deposit. There is no one-size-fits-all mediated submission workflow; every organization has a slightly different organizational structure and legal context that informs how they need to construct their submission workflows, and a single institution may require several different submission workflows based on the type of content, its point of origin, or the collection it is intended to be part of.

For the CLAW IR MVP, we will need at least a minimal toolkit for allowing IR managers to model whatever submission process they need and then create and use it, taking advantage of authentication when available.

## Google Scholar & SEO

Good SEO should be a goal for any digital repository, and institutional repositories also have Google Scholar integration with which to contend. For the CLAW IR MVP, we will not only require that all standard SEO best practices are followed (robots.txt, sitemap.xml, google.com/webmaster integration, etc.), but also that special features that optimize Google Scholar indexing are included for IR objects.

## Object Importing

CLAW should be able to create new objects based on DOIs, PubMed IDs, and possibly other external sources. Additionally, CLAW should also support standard batch ingestion of object packages the way 7.x-1.x allows for zip file imports.

## Migration Utilities

In order to move from Scholar 7.x-1.x to CLAW, current users will need access to a well-tested toolkit that can transform 7.x-1.x data into CLAW data. This should cover the transformation of MODS data into RDF, as well as the conversion of 7.x-1.x IP embargoes, Scholar embargoes, and XACML POLICY datastreams into RDF or Drupal permissions.