

Ch 06b: IIIF Manifests, Canvases and Annotations

This guide describes the IIIF features provided by Heurist for creating, importing, viewing, editing and exporting IIIF Manifests, Canvases and Web Annotations.

Heurist supports two main workflows:

1. **Use Heurist as an annotation layer over existing IIIF Manifests (annotation overlay mode).** The external provider keeps ownership of the source Manifest and Canvas identifiers. Heurist stores and publishes local annotations.
2. **Use Heurist to manage the Manifest (full management mode).** Heurist stores Manifest, Canvas and Annotation records and generates a IIIF Presentation API v3 Manifest from those records.

Heurist also provides a dynamic IIIF server for ordinary record sets and registered media files, and can render external IIIF files and Manifests. In this sense it can act both as a IIIF client and as a IIIF server.

1. Preparation

1.1 Import the required definitions

Before using the IIIF annotation and Manifest tools in an existing database, import the new definitions from the `Heurist_Core_Definitions` database using **Design > Browse templates**. Heurist will prompt you to do this if you attempt to process Manifests without the required definitions.

Browse templates prompt

The new record types are in the **Documents** group. It is enough to select **IIIF Annotation**. The related record types **IIIF Manifest** and **IIIF Canvas** are downloaded alongside it.

IIIF Annotation template selection

The important record types are:

- **IIIF Annotation** (`RT_IIIF_ANNOTATION`, concept code 2-109)
- **IIIF Manifest** (`RT_IIIF_MANIFEST`, concept code 2-110)
- **IIIF Canvas** (`RT_IIIF_CANVAS`, concept code 2-111)

These definitions include fields for IIIF identity, original/source IIIF identity, Manifest links, Canvas links, annotation state, selector type/value, annotation JSON and related metadata.

1.2 Remove obsolete duplicate fields in old databases

Some older databases may contain a duplicated field named **IIIF Anotation 2** with:

- local ID: 1106
- concept code: 2-1098

This field is not used by any current IIIF record type. Remove it before using the new IIIF workflow, especially if it causes confusion in forms or import checks.

1.3 Recommended checks before testing

After importing definitions, check that the database contains the three IIIF record types above and that **Browse templates** no longer shows missing IIIF definitions in the Core definitions database.

For testing, start with a small Manifest first. A large external Manifest may fail for reasons unrelated to Heurist logic, such as network timeouts, remote annotation-list delays, or unavailable image services.

2. Key concepts

2.1 Manifest

A Manifest is the IIIF object that describes a digital object, such as a manuscript, book, image set or media collection. In Heurist, a Manifest may be:

- a registered external Manifest file or URL;
- a managed **IIIF Manifest** record;
- a dynamic Manifest generated from a record set or a single registered media file.

Managed Heurist Manifest output is generated as **IIIF Presentation API v3**. A registered IIIF Manifest file becomes managed only when an **IIIF Manifest** record references that file. If no such record exists, Heurist treats the registered Manifest file as an external/source Manifest and can use it as an annotation overlay target.

2.2 Canvas

A Canvas represents one viewable unit in a Manifest, for example a page, image, video or audio item. In full management mode, Heurist stores each Canvas as an **IIIF Canvas** record. Each managed Canvas normally points to a registered file or registered external media URL.

In annotation overlay mode, Canvas records are not imported or managed by Heurist. Instead, annotations remain linked to the original Canvas URI from the source Manifest.

2.3 Annotation

Annotations are stored as **IIIF Annotation** records. They may be created or edited in Mirador, mainly for defining the annotation area and initial text, or in the Heurist record editor for annotation attributes, which can be extended to support searching and custom reporting within Heurist.

Annotations store:

- text body / summary;
- motivation, such as commenting;
- language;
- original Canvas target URL;
- managed Canvas reference when applicable;
- selector type and selector value;
- raw IIIF/Web Annotation JSON;
- state, such as imported, Mirador-created, Heurist-created, modified, obsolete or removed.

3. Manual creation of a managed Manifest

Manual creation is used when you want Heurist to own and generate the Manifest rather than only overlay annotations on an external Manifest.

3.1 Create the Manifest record

Create a new **IIIF Manifest** record. Fill in Manifest-level metadata such as title, description and copyright/rights. These fields are used when Heurist generates the v3

Manifest output.

A managed Manifest can be empty. An empty managed Manifest still returns valid IIIF Presentation API v3 JSON with `items: []`, so viewers should not normally show a technical error.

3.2 Add Canvases one by one

Create **IIIF Canvas** records and link them to the Manifest. Each Canvas may reference:

- a locally uploaded registered file;
- a registered external media URL;
- an image served by a IIIF Image API;
- other supported media such as audio or video where configured.

The order of Canvas references on the Manifest record defines the order in the generated Manifest. The order can be changed within Heurist data entry by dragging the Canvas references up and down.

3.3 Add or edit annotations in Mirador

Open the managed Manifest in the Mirador Viewer. Use Mirador's annotation tools to add annotations to the selected Canvas. Heurist stores the annotation as an **IIIF Annotation** record and links it back to the relevant Canvas and Manifest context.

The internal Mirador viewer uses the default annotation lookup scope `canvas`, which reads annotations from `/api/{db}/annotations`. A Manifest-scoped endpoint is also available as `/api/{db}/annotations/{manifestRecID}` when `annotation_scope=manifest` is requested.

3.4 Edit annotations in the Heurist record editor

Annotations can also be edited directly as Heurist records. This is useful for correcting text, language, motivation or metadata.

Be careful when editing selector information manually:

- **Selector type** and **selector value** must remain consistent.
- A rectangular fragment selector and an SVG selector are not interchangeable.
- If the selected area is edited incorrectly, Mirador may display the annotation in the wrong place or fail to display the region.

In general, use Mirador for changing the selected area and use Heurist record editing for textual and descriptive metadata.

3.5 Open the Manifest, Canvases and Annotations from the Record View panel

From the **IIIF Manifest** record view, open the Manifest either as raw/generated IIIF content or in the Mirador Viewer.

Related records can also provide navigation back to the Manifest context:

- **IIIF Canvas** records may include a link to open the referenced Manifest in which the Canvas is used.
- **IIIF Annotation** records may include a link to open the referenced Manifest, so the annotation can be viewed in its wider Manifest context rather than as an isolated record.
- **IIIF Canvas** records can also be opened independently, in the same way as any Heurist record with a file field. This is useful when checking a single page/image/media item before opening the full Manifest.

For internal Mirador viewing, Heurist passes `omit_annotation_pages=1` to the generated Manifest URL where needed. This prevents the same database annotations from being loaded twice: once from embedded Manifest annotation-page links and once from

Mirador's annotation endpoint.

3.6 Add Canvases in a batch — planned feature

A planned batch action will allow users to select one or several ordinary records that already have file fields and create Canvas records from those files. This is intended to make managed Manifest creation faster for large image sets.

Until this is implemented, add Canvas records manually or import/process an existing Manifest in full management mode.

4. Import or process an existing IIIF Manifest

Use **Process IIIF Manifest** to work with a registered or uploaded IIIF Presentation Manifest. A Manifest can be registered as:

- an external IIIF Presentation Manifest referenced by a File field;
- a JSON Manifest uploaded to Heurist as a File field.

Process IIIF Manifest dialog

The default mode is **Full manifest management**, which creates or updates an **IIIF Manifest** record, imports **IIIF Canvas** records and imports available **IIIF Annotation** records.

Annotation overlay is different: it imports annotations only. It does not create an **IIIF Manifest** record. The registered Manifest file remains the source Manifest and Heurist stores local annotations against the original Canvas URIs.

4.1 Annotation overlay mode

Use **Annotation overlay** when the external Manifest remains the authoritative source for Canvas structure.

In this mode:

- only **IIIF Presentation API v3 Manifests** are supported;
- the source Manifest and its Canvas list remain owned by the external provider;
- Heurist does **not** create an **IIIF Manifest** record;
- Canvas identifiers are preserved from the source Manifest;
- annotations are imported into Heurist and linked to the original Canvas URIs;
- when `/api/{db}/iiif/manifest/{obfuscatedFileID}` is requested, Heurist can output a v3 overlay Manifest by replacing source `Canvas.annotations` with Heurist `AnnotationPage` links;
- local Heurist annotations are preserved on re-import/re-processing when they have been edited locally.

Do not use this mode for IIIF Presentation API v2 Manifests. For v2 source Manifests, use full management mode. If a managed **IIIF Manifest** record already references the selected registered Manifest file, annotation overlay mode is not available because the file is already under Heurist management.

4.2 Full manifest management mode

Use **Full manifest management** when Heurist should manage the Manifest structure.

In this mode:

- Heurist creates or updates Manifest, Canvas and Annotation records;
- the existence of the **IIIF Manifest** record is what marks the registered Manifest file as managed;
- Heurist owns the generated Manifest output, Canvas order and Canvas metadata;
- media may still be external registered resources or local uploads;

- media should be stored in Heurist where referenced resources are not held by a stable long-term repository or institutional service;
- Manifest-level metadata can be edited in Heurist;
- Canvas order comes from the Canvas references stored on the Manifest record;
- annotations are linked to managed Canvas records;
- generated IIIF output uses Heurist Canvas API URLs.

This is the preferred mode for IIIF v2 source Manifests, because the overlay workflow is v3-only.

4.3 Re-import / re-processing behaviour

On re-import, Heurist attempts to update imported records while preserving local work. Records that have been changed in Heurist or Mirador are preserved by default and reported separately as preserved local records.

The report includes:

- managed Manifest record ID, or not created for annotation overlay;
- total Canvases found;
- Canvas records added, updated, unchanged or preserved;
- total annotations found;
- annotation records added, updated, unchanged or preserved;
- issues encountered during import/processing.

4.4 Thumbnails

The import tool can create thumbnails for annotation records. This is useful for browsing annotations in Heurist, but it is slower because it may need to access remote images or render selected regions.

5. Add annotations for an arbitrary registered file or URL

You do not need a managed Manifest before annotating media.

You can open the Mirador Viewer for any registered media file or supported registered URL. Heurist dynamically creates a single-canvas Manifest for the media and lets you add annotations. These annotations are stored in Heurist against the Canvas URL used for that file.

If you later add the same file to a managed Manifest, the annotation can be preserved because the Canvas identity is based on the registered file's obfuscated ID. This allows annotation work to start before the final Manifest structure is prepared.

Typical uses:

- annotate a single image before adding it to a larger Manifest;
- annotate a registered external IIIF image;
- test annotation behaviour on one file before importing, processing or building a large Manifest.

6. Viewing in Mirador

Heurist provides a Mirador Viewer for:

- a managed IIIF Manifest record;
- a registered external Manifest file;
- a single registered media file;
- a dynamic Manifest generated from a query or selected record set.

Registered Manifest files are opened through

`/api/{db}/iiif/manifest/{obfuscatedFileID}`. If an **IIIF Manifest** record references the

file, the API returns the managed Manifest generated from Heurist records. Otherwise it returns the source Manifest: v2 sources are returned as-is, while v3 sources can be returned with Heurist annotation-page links overlaid.

The viewer supports two annotation lookup scopes:

- `annotation_scope=canvas` — default. Shows all annotations that target the same Canvas URL.
- `annotation_scope=manifest` — shows only annotations linked to the current Manifest record.

For internal Mirador viewing, Heurist avoids duplicate annotations by passing `omit_annotation_pages=1` to generated Manifest URLs where needed. External IIIF consumers can receive normal `Canvas.annotations` links when this parameter is not used.

7. Dynamic Manifests via Export IIIF

Heurist can generate IIIF output dynamically from ordinary record searches and file selections. This is useful when you want to view or share a record set without creating a permanent managed Manifest record.

7.1 Single registered media file

A single media file can be opened in Mirador or exported as a IIIF Manifest by using its registered file obfuscated ID. Heurist wraps the media in a single-canvas IIIF Presentation API v3 Manifest.

Useful for:

- quick viewing of one image, audio or video item;
- adding annotations to one registered file;

- testing IIIF output for one file.

7.2 One ordinary record with media files

When a record contains one or more suitable file fields, Export IIIF can generate a Manifest whose Canvases correspond to the media files linked to that record.

Useful for:

- records that represent objects with several images;
- quick Mirador viewing without creating explicit Canvas records;
- public sharing of record media as IIIF.

7.3 Several ordinary records with media files

When the current record set contains multiple records with suitable media, Export IIIF can generate a Manifest with one or more Canvases from those records, subject to the export limit.

Useful for:

- search results containing image records;
- temporary collections;
- comparing several media records in Mirador.

7.4 One registered IIIF Manifest in the record set

If a record set contains one registered IIIF Manifest and no generated media Canvases, Heurist can return that Manifest directly through the IIIF API.

Useful for:

- opening a registered external Manifest through Heurist;
- keeping a registered Manifest discoverable as a file in a record;

- testing external Manifest access.

7.5 Several registered IIIF Manifests in the record set

If a record set contains several registered IIIF Manifests, Heurist can generate a IIIF Collection that references those Manifests.

Useful for:

- publishing a set of related Manifests;
- opening several Manifests together in Mirador;
- grouping imported, processed or external Manifests without merging their Canvas structures.

7.6 Mixed record set: registered Manifests and media files

If a v3 dynamic export contains both registered Manifests and ordinary media Canvases, Heurist can generate a Collection. Registered Manifests become Manifest items in the Collection; generated media Canvases are grouped into a generated Manifest item.

Useful for mixed search results where some records already contain IIIF Manifests and others contain image/audio/video files.

7.7 IIIF v2 output policy

Heurist no longer generates IIIF Presentation API v2 output. Dynamic export and managed Manifest output are v3-only. Heurist can still import v2 and hybrid v2 source Manifests in **Full manifest management** mode and then publish them as generated v3 Manifests.

8. Recommended workflow examples

8.1 Annotate an external v3 Manifest without taking over its structure

1. Register or upload the v3 Manifest JSON.
2. Open **Process IIIF Manifest**.
3. Select **Annotation overlay**.
4. Import/process annotations.
5. Open the registered Manifest file in Mirador. The viewer uses `/api/{db}/iiif/manifest/{obfuscatedFileID}` and the annotation endpoint.
6. Add or edit annotations.
7. Use the same API URL when external viewers need the v3 source Manifest with Heurist AnnotationPage links.

8.2 Import a v2 Manifest with many Canvases and annotations

1. Register or upload the v2 Manifest JSON.
2. Open **Process IIIF Manifest**.
3. Select **Full manifest management**.
4. Import/process Canvases and annotations.
5. Inspect the report for failed remote annotation lists or unavailable image resources.
6. Open the managed Manifest in Mirador.

If the v2 Manifest is very large, test first with a trimmed Manifest containing a few Canvases.

8.3 Start with one image and later build a Manifest

1. Register or upload an image.

2. Open the image in Mirador.
 3. Add annotations.
 4. Later create a managed Manifest and add that file as a Canvas.
 5. The annotation can be preserved because it targets the file-based Canvas identity.
-

9. Troubleshooting

The import widget says required definitions are missing

Import **IIIF Annotation** from `Heurist_Core_Definitions`. The related Manifest and Canvas record types should be imported with it.

The database contains an old field named “IIIF Anotation 2”

Remove the obsolete duplicate field with local ID `1106` and concept code `2-1098`. It is not used by the current IIIF record types.

Overlay mode rejects a v2 Manifest

This is expected. Annotation overlay mode is v3-only because it stores annotations against original v3 Canvas URIs and can publish v3 `Canvas.annotations` `AnnotationPage` links. Import v2 Manifests in **Full manifest management** mode.

Overlay mode is disabled for a selected registered Manifest file

This means an **IIIF Manifest** record already references the selected registered Manifest file. That file is already managed by Heurist, so use **Full manifest management** mode.

Mirador shows duplicate annotations

Use the internal Heurist Mirador viewer, which passes `omit_annotation_pages=1` for generated Manifest URLs where required. This avoids loading the same annotations both from Manifest `Canvas.annotations` and from Mirador's annotation endpoint.

Import fails on a very large Manifest

Try a small trimmed Manifest first. Failures may be caused by remote annotation-list access, timeouts, malformed source JSON, unavailable image services, or network interruptions.

10. Summary of ownership by mode

Feature	Annotation overlay	Full manifest management
Supported source Manifest version	v3 only	v2 and v3
Source Manifest ownership	External provider / registered file	Imported into Heurist management
Generated Manifest output	Source v3 Manifest with Heurist AnnotationPage links when requested through the IIIF API	Heurist managed v3 output
Canvas list ownership	External provider	Heurist
Canvas identifiers	Original source Canvas URIs	Heurist Canvas API URLs
Canvas records created	No	Yes
Annotation records created	Yes	Yes
Manifest metadata editable in Heurist	No managed Manifest record is created	Yes, used in generated output
Best use	Add Heurist annotations to an existing v3 Manifest without creating a Manifest record	Build or take over a Manifest in Heurist

Process IIIF Manifest

Reads a registered **IIIF Presentation Manifest** and creates or updates records representing its contents. A manifest describes a digital resource's ordered pages or views, known as *canvases*, together with media references, metadata and annotations. The manifest may be uploaded locally or registered through an external URL.

Two processing modes are available:

- **Full manifest management:** creates or updates Manifest, Canvas and Annotation records. Heurist manages the resulting manifest structure, including canvas order and metadata, and can generate its own IIIF manifest output. Referenced media may remain external or be stored locally.
- **Annotation overlay:** imports annotations while retaining the source manifest's canvas structure and identifiers. No Heurist Manifest record is created. This mode supports IIIF Presentation API version 3 and is unavailable where the selected file already has a managed Manifest record.

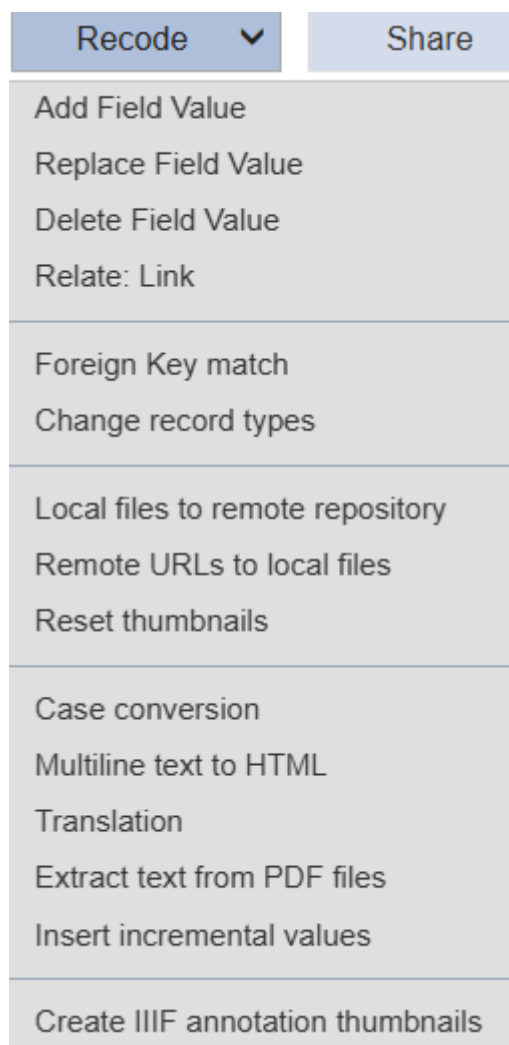
On re-import, records modified locally in Heurist or Mirador are preserved by default and reported separately.

Annotation thumbnails can subsequently be generated for selected annotation records using **Recode > Create IIIF annotation thumbnails** above the results in the Explore menu.

Explore > Recode > Create IIIF annotation thumbnails

TODO: duplicated at end of chapter 6b, need to decide where it belongs and simply reference

Creates a thumbnail showing the image region identified by each IIIF annotation. For example, an annotation marking a detail in a manuscript page receives a thumbnail of that detail, making it easier to recognise the annotation in record lists.



The operation processes **IIIF Annotation records in the current search results**, subject to access permissions. Other record types are excluded. In the current implementation, ticking individual records does not restrict the operation: first use a search to obtain the annotations you want to process.

Creating or replacing thumbnails

The dialogue contains a checkbox, **Create thumbnails for missed only**, selected by default.

- **Checked:** creates thumbnails for annotations without a thumbnail reference and retains existing references.
- **Unchecked:** recreates thumbnails for all annotations being processed, replacing their existing thumbnail references when creation succeeds.

Use the default setting to fill gaps after importing annotations. Clear it when thumbnails need updating—for example, after changing an annotation’s selected region or its source image.

How the thumbnail is generated

Heurist reads the annotation’s region selection and locates the image associated with its canvas. The image can come from an external IIIF image service, a locally stored file or a directly accessible remote image.

Supported selections include:

- **Rectangular regions**, stored as a pixel-based fragment selector.
- **SVG regions**, such as polygons or other supported shapes. The thumbnail shows the rectangular area enclosing the shape, including any surrounding image within that rectangle.

The cropped image is scaled to fit within **200 × 200 pixels**, preserving its proportions, stored locally and attached to the annotation’s Thumbnail field. The annotation text, selected region and original image are unchanged.

Requirements and results

The database must contain the **IIIF Annotation record type, Concept ID 2-109**, and the necessary annotation fields. Each annotation needs a supported region selection and a resolvable canvas image.

A thumbnail may fail to generate if the region is missing or unsupported, the canvas image cannot be identified, the remote resource is inaccessible, or image processing or file storage fails.

The operation reports processed records and failures. **The processed count includes annotations whose existing thumbnail references were retained**, so

it does not necessarily equal the number of newly generated images.

Missing-only mode checks for an existing thumbnail reference; it does not verify that the referenced image file still exists. To repair a broken thumbnail, clear the checkbox and recreate it.

■

Revision #8

Created 2026-07-01 12:00:04 UTC by IanJohnson

Updated 2026-10-06 12:39:07 UTC by IanJohnson