

IIIF Configuration

- [Overview](#)
 - [Format Support](#)
- [Enable IIIF Support on Backend](#)
 - [Install a IIIF Image Server](#)
 - [Installing and Configuring Cantaloupe](#)
 - [Required IIIF Configuration](#)
 - [Additional Configuration Options](#)
 - [CORS Configuration](#)
 - [IIIF Search API](#)
- [Enable/Install the Mirador Viewer on Frontend](#)
 - [Configuring Mirador](#)
- [Configure IIIF viewer via Metadata Fields](#)

Overview


Supported in 7.1 or above

 IIIF support was first added to DSpace in version 7.1. It was not available in 7.0 or below.

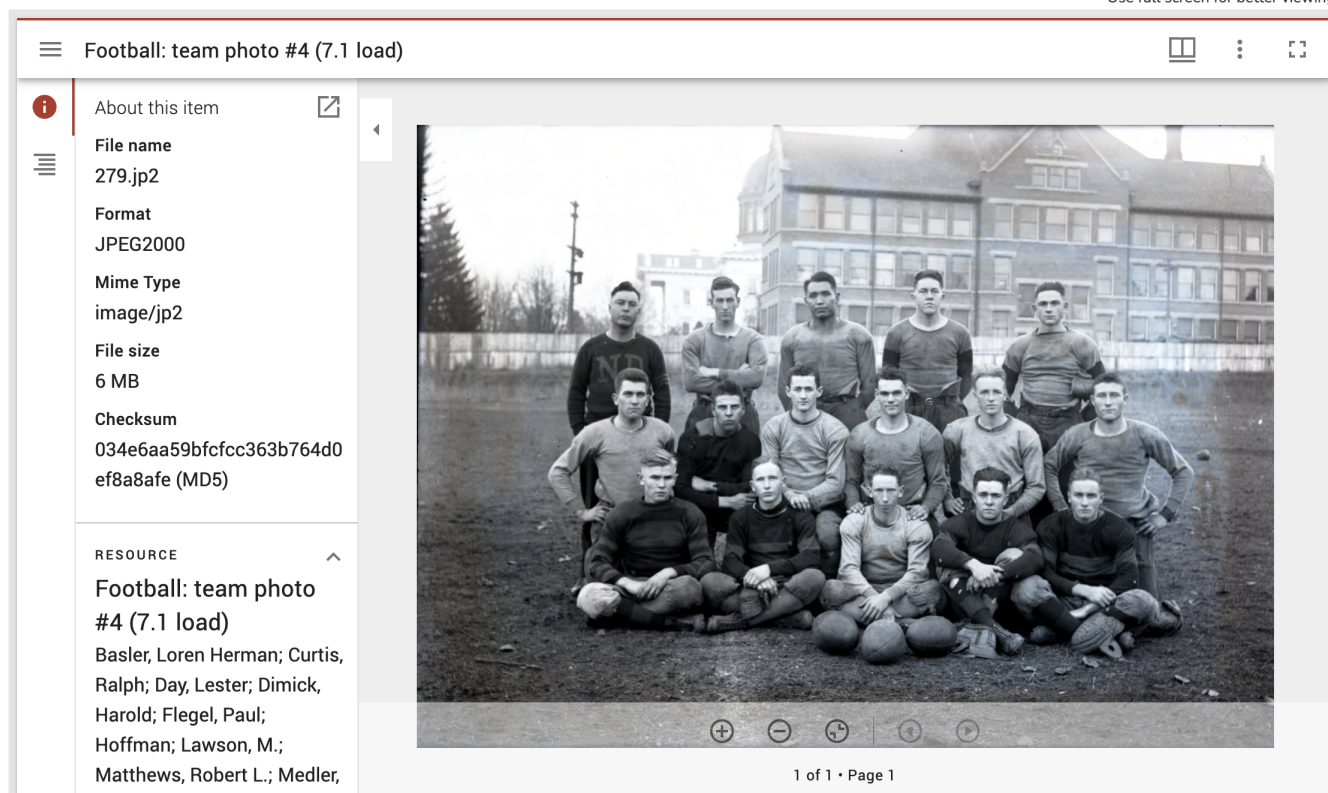
DSpace supports the [International Image Interoperability Framework \(IIIF\)](#). The DSpace REST API implements the [IIIF Presentation API version 2.1.1](#), [IIIF Image API version 2.1.1](#), and the [IIIF Search API version 1.0](#) (experimental). The DSpace Angular frontend uses the [Mirador 3.0](#) viewer.

Administrators can configure IIIF behavior at the Collection, Item, Bundle and Bitstream levels using metadata. To support additional sharing, viewing, comparing, and annotating, DSpace can be configured to share IIIF metadata with external IIIF clients (see [CORS Configuration](#)). IIIF REST endpoints implement the same security protocol as the primary REST API so that DSpace authorization policies are enforced for IIIF access as well.

IIIF Image Server

 Running IIIF in production requires an IIIF-compatible image server. You are free to use any compatible image server you choose. However, instructions for configuring the [Cantaloupe Image Server](#) are included below. A [preconfigured Cantaloupe image server can be started via docker-compose](#) to simplify evaluation and testing.

Use full screen for better viewing.



Format Support

Currently, DSpace only supports IIIF viewing of Image formats (any format whose MIME type starts with "image/*"). For example, PDF viewing is not currently supported.

Enable IIIF Support on Backend

DSpace IIIF support is not enabled by default. To enable IIIF, you first need to install a IIIF Image Server, and then update your DSpace configuration as described below.

Install a IIIF Image Server

The [Cantaloupe Image Server](#) is currently recommended for use with DSpace, but you are free to use the image server of your choice. [A list of IIIF-compliant image servers](#) is maintained by the IIIF community.

Here is a brief overview of how the IIIF image server works with DSpace.

First, the base path to the image server is defined in `config/modules/iiif.cfg`.

```
iiif.image.server = https://imageserver.mycampus.edu/image-server/cantaloupe/iiif/2/
```

Given this configuration, the IIIF manifest returned by the DSpace backend will include an image resource annotation like the following:

IIIF Image Resource Annotation

```
resource: {
  @id: "https://imageserver.mycampus.edu/image-server/cantaloupe/iiif/2/4b415036-57a8-42f4-a971-c5e982f55f92/full/full/0/default.jpg",
  @type: "dctypes:Image",
  service: {
    @context: "http://iiif.io/api/image/2/context.json",
    @id: "https://imageserver.mycampus.edu/image-server/cantaloupe/iiif/2/4b415036-57a8-42f4-a971-c5e982f55f92",
    profile: "http://iiif.io/api/image/2/level1.json",
    protocol: "http://iiif.io/api/image"
  },
  format: "image/jpeg"
}
```

The Mirador viewer (see below) uses this annotation to communicate with the image server using the IIIF Image API.

Finally, notice that the image server needs to retrieve the requested bitstream from DSpace. There are a number of ways to do this and the details vary with the image server chosen. The easiest approach is for the image server to request the bitstream via HTTP and the DSpace API, e.g.:

```
http://dspace.mycampus.edu:8080/server/api/core/bitstreams/4b415036-57a8-42f4-a971-c5e982f55f92/content
```

Installing and Configuring Cantaloupe

The Cantaloupe [getting started page](#) provides installation instructions. The basic installation process is simple.

The simplest way to configure Cantaloupe to retrieve images from DSpace is to use [HTTPSource](#) with the following configuration.

```
HttpSource.BasicLookupStrategy.url_prefix = <dspace-url>/server/api/core/bitstreams/
HttpSource.BasicLookupStrategy.url_suffix = /content
```

Required IIIF Configuration

To enable IIIF, edit `config/modules/iiif.cfg` or your `local.cfg` file and set `iiif.enabled` to be `"true"`.

```
iiif.enabled = true
```

In addition, you need to provide the URL for your newly installed IIIF image server. e.g.:

```
iiif.image.server = http://localhost:8182/iiif/2/
```

Finally, update `dspace.cfg` or your `local.cfg` file by adding `"iiif"` to the default event dispatcher, as shown below:

```
event.dispatcher.default.consumers = versioning, discovery, eperson, iiif
```

With these changes in place, DSpace will be ready to respond to IIIF requests. Restart your DSpace backend (i.e. Tomcat) for these changes to all take effect.

Additional Configuration Options

The full set of IIIF configuration options can be found in [config/modules/iiif.cfg](#).

Property	Description
iiif.enabled	Enables the DSpace IIIF service.
iiif.image.server	Base URL path for the IIIF image server. e.g. http://localhost:8182/iiif/2/
iiif.document.viewing.hint	Default viewing hint. Can be overridden with the metadata setting described below.
iiif.logo.image	Optional URL for a small image. This will be included in all IIIF manifests.
iiif.cors.allowed-origins	Comma separated list of allowed CORS origins. The list must include the default value: <code>\${dspace.ui.url}</code> .
iiif.metadata.item	Sets the Dublin Core metadata that will be added to the IIIF resource manifest. This property can be repeated.
iiif.metadata.bitstream	Sets the Bitstream metadata that will be added to the IIIF canvas metadata for individual images. This property can be repeated.
iiif.license.uri	Sets the metadata used for information about the resource usage rights.
iiif.attribution	The text to use as attribution in the iiif manifests. Defaults to: <code>\${dspace.name}</code>
iiif.document.viewing.hint	Either "individuals", "paged" or "continuous". Can be overridden with the metadata setting described below.
iiif.canvas.default-width	Default value for the canvas size. Can be overridden at the item, bundle or bitstream level.
iiif.canvas.default-height	Default value for the canvas size. Can be overridden at the item, bundle or bitstream level.

Canvas Dimensions

As of 7.2, the canvas dimension options (*iiif.canvas.default-width* and *iiif.canvas.default-height*) are updated with additional behaviors.

- If you do not provide your own default dimensions in *iiif.cfg*, DSpace will attempt to optimize canvas dimensions when dimension metadata is missing from the first bitstream in the item. This will often produce more accurate viewer layouts, but note that it is not sufficient to assure accurate layouts in all cases.
- If you decide to add your own default dimensions in *iiif.cfg* file your dimensions are used for every bitstream that lacks dimension metadata.
- You may also set both default dimensions in *iiif.cfg* to the value `-1`. In this case, DSpace creates accurate default dimensions for every bitstream that lacks dimension metadata. Note that this impacts performance.

It is recommended that *iiif.image.width* and *iiif.image.height* metadata be added to Item, Bundle, or Bitstream metadata to assure accurate layout and top performance. Default dimension configurations are intended to improve the user experience when dimension metadata has not yet been added.

CORS Configuration

The wildcard `***` configuration is the default CORS setting for IIIF. With this setting, all remote viewers and applications can retrieve manifests, assuring maximum interoperability. You can restrict CORS origins using the *iiif.cors.allowed-origins* property defined in *iiif.cfg*. Remove the wildcard and add a comma-separated list of origins instead.

IIIF Search API

DSpace includes a plugin to support the IIIF Search API. This plugin is designed to work specifically with the [Solr OCR Highlighting Plugin](#) and METS /ALTO data. You are welcome to experiment with the plugin. To do so, uncomment the following settings in *config/modules/iiif.cfg*:

```
iiif.search.url = ${solr.server}/word_highlighting
iiif.search.plugin = org.dspace.app.rest.iiif.service.WordHighlightSolrSearch
```

Once you have successfully indexed ALTO files using the Solr plugin, you can enable search within a DSpace Item by adding the *iiif.search.enabled* metadata field.

Indexing Support

Support for indexing OCR files using the the Solr OCR Highlighting Plugin or other services is not currently provided by DSpace. Institutions will need to develop their own approach to indexing their data.

Enable/Install the Mirador Viewer on Frontend

The [Mirador 3.0 viewer](#) is included in the dspace-angular (UI) source code. Before enabling Mirador, be sure to review [the instructions for installing the Angular frontend](#) if you haven't already.

To add the Mirador viewer to your DSpace frontend installation, run the following command:

```
# This builds and runs the DSpace UI with the Mirador Viewer in a single step
yarn run start:mirador:prod
```

This will build and copy Mirador to your `dist/` directory and start the frontend server.

The actual steps for deploying the Angular UI with Mirador into Production will likely vary with your setup. However, one possible command-line scenario is the following:

```
# Build Mirador viewer
yarn run build:mirador
# Build DSpace UI for production
yarn run build:prod
# Run the DSpace UI with Mirador viewer
yarn run serve:ssr
```

Running in Development

 In the Dspace 7.1 release, the Mirador viewer cannot be used when running in development mode. For now, you need to use a production build.


Configuring Mirador

The Mirador viewer is highly configurable. The [Mirador configuration file for DSpace](#) includes a number of settings that you can override manually, including CSS values for styling. Note that some of the Mirador behavior (like the inclusion of thumbnail navigation on the right) is set by the Angular component at runtime. You can choose to override these runtime settings if you like.

Configure IIIF viewer via Metadata Fields

IIIF configuration at the Item-level is quite flexible and is managed using metadata. Canvas sizes, image labels, ranges and other settings are controlled by using the following fields.

Required Field

 Note that the `dspace.iiif.enabled` metadata field **MUST** be added to the Item and set to "true". Otherwise, the Item display will use the default DSpace view.

Schema	Element	Qualifier	Scope	Description
dspace	iiif	enabled	Item	Stores a boolean text value (true or false) to indicate if the iiif feature is enabled or not for the dspace object. If absent the value is derived from the parent dspace object.
iiif	label		Bitstream	Metadata field used to set the IIIF label associated with the canvas resource otherwise the system will derive one according to the configuration setting or the canvas.naming metadata field.
iiif	description		Item	Metadata field used to set the IIIF description associated with the resource.
iiif	canvas	naming	Item	Metadata field used to set the base label used to name all the canvas in the Item. The canvas label will be generated using the value of this metadata as prefix and the canvas position. e.g. Page 1, Page 2, etc.
iiif	viewing	hint	Item	Metadata field used to set the viewing hint overriding the configuration value if any. Possible values are "individuals" and "paged". Default value: individuals.
iiif	image	width	Item, Bundle, or Bitstream	Metadata field used to store the width of an image in pixels. Determines the canvas size.
iiif	image	height	Item, Bundle, or Bitstream	Metadata field used to store the height of an image in pixels. Determines the canvas size.
iiif	toc		Bitstream	Metadata field used to set the position of the iiif resource in the "table of contents" structure.
iiif	search	enabled	Item	Metadata field used to enable the IIIF Search service at the item level. This feature is experimental and requires additional setup.