

Purpose

- 1) Describe the implementation of IIIF API and IIPIImage server in the RUcore framework
- 2) Outline the process to generate pyramidal TIFF datastreams
- 3) Outline the process for upgrading the JPEG thumbnail datastream to a higher base resolution of 220 pixels

IIIF Implementation and IIPIImage Server

IIIF API Base URI

Initially it was proposed the base URI for the International Image Interoperability Framework APIs in RUcore be <https://rucore.libraries.rutgers.edu/rutgers-lib/iiif>. After some consideration it is the recommendation of this document that the base URI be moved to the URI location in RUcore where other access and presentation APIs reside; <http://rucore.libraries.rutgers.edu/api/iiif>.

Image Server Access to Pyramidal TIFF datastreams

The IIPIImage server is being used to generate JPEG derivatives from pyramidal TIFF files via the IIIF API. When locating the pyramidal TIFF datastream on the file system the path and file name stored in Fedora contains plus sign(+) characters. An example is:

```
/mellon/datastreams/2016/0106/15/46/rutgers-lib_469+PTIF-1+PTIF-1.0
```

The IIPIImage server filters out the plus sign (+) from the file name of the datastream before attempting to render the request. The filtering is not an option that can be disabled. Symbolic links to the datastream file that do not contain characters that IIPIImage would filter out has been tested and is a viable solution. While symbolic links are not desirable the management and location of the link can be limited to a non-web accessible location. The symbolic link can be checked at the time of access and updated to the most recent version of the pyramidal TIFF when appropriate. The symbolic link would appear like so:

```
rutgers-lib_469_ptif-1 -> /mellon/datastreams/2016/0106/15/46/rutgers-lib_469+PTIF-1+PTIF-1.0
```

IIPIImage Server IP Restriction

Using the IIF specified API as a pass through for accessing the IIPIImage server the IIPIImage server itself can be IP restricted to localhost.

Generating Pyramidal TIFFs

Starting in 8.1 WMS will be able to generate pyramidal TIFF presentation datastreams and the discovery layer will be able to serve these datastreams using the high resolution image viewer OpenSeadragon.

To increase the impact of this feature an update of our legacy resources will need to be performed. I recommend starting with resources that are type 'StillImage' and expand to resources that are type 'Text' in the future.

Pyramidal TIFF's will be generated from archival files. Due to the legacy of the ARCH datastream most of the ARCH datastreams are still TAR files. Those TAR files need to be opened and examined. We could wait to perform this task

until after the legacy TAR ARCH datastreams have been migrated to the new ARCH/DARCH schema of one file per archival datastream. The PTIF generation would be far easier and much more accurate.

At this point I think the Software Architecture group should discuss whether to wait for the legacy ARCH migration or proceed and develop the logic of how to deal with both the current ARCH/DARCH schema and the legacy ARCH/TAR's. The result of the discussion should be reported to the Cyberinfrastructure working group for feedback.

Software Architecture Decision from 5/5/2016 meeting

Still image resources on the test environment should not have their archival master's bound in a TAR file anymore. This task is slated to be performed on the staging and production environments in the near future. The task of creating pyramidal TIFF files will be dependent on the completion of the archival TAR migration task.

Generating Larger THUMBJPEG-1 Datastreams

A script has been developed to complete the task of creating thumbnail JPEG's that are 220 pixels either wide or high. The logic of the script is as follows.

- 1) Using the Fedora datastream table, objectPaths, all valid tokens starting with 'rutgers-lib' are considered for this job.
- 2) The MODS for each rutgers-lib token is examined and the mods:TypeOfResource is leverage to decide which resource may need a new thumbnail created.
- 3) If the resource is type 'StillImage', 'Text', or 'MovingImage' further analysis is performed.
- 4) The current THUMBJPEG-1 for the resource is examined to determine if the current THUMBJPEG-1 already meets the standard. If so, a new THUMBJPEG-1 is not created. This is check optional, but recommended. There is no need to create an almost identical THUMBJPEG-1.
- 5) To create a THUMBJPEG-1 datastream a higher quality datastream is needed to derive the thumbnail from. Because of the size of the thumbnail using the archival masters would be overkill, presentation datastreams are used.
 - a. For resources that are type 'StillImage' and 'Text' the following logic is applied.
 - i. If a JPEG-1 exists use that
 - ii. If no JPEG-1 exists, look for a REF-1 datastream and use the first page of that datastream
 - iii. If no REF-1 exists, look for a PDF-1 datastream
 - iv. If none of those datastreams exist, mark the conversion as not complete
 - v. Generate thumbnail using ImageMagick with the PECL class
 - b. For resources that are type 'MovingImage' the following logic is applied.
 - i. If a MP4-1 exists use that
 - ii. If no MP4-1 exists, look for a MOV-1 and use that
 - iii. If no MOV-1 exists, look for a FLV-1 and use that
 - iv. If none of those datastreams exist, mark the conversion as not complete
 - v. Generate thumbnail using FFMPEG and the frame of video 5 seconds into the datastream. This is the same process that has been used by the analytic tool to generate dynamic thumbnails
- 6) If a new THUMBJPEG-1 is generated use the Fedora add datastream method. The old THUMBJPEG-1, if it exists, will be preserved as a version.

There are some details about running the script that will not be mentioned in this document, but rather in the scripts readme file. Those include dry runs and other script configuration.

WMS has been creating THUMBJPEG-1 at the new standard for several RUCore releases. The discovery layer has accommodated the mixed sizes of the THUMBJPEG-1 datastreams. For 8.1 the new larger sized THUMBJPEG-1 is assumed and the discovery layer will not elegantly support the older/smaller THUMBJPEG-1 datastreams. That being said, the script can be run at any time before 8.1 and should not be tied to the 8.1 RUCore installation process.

This script was already run on the development and test environments. The test environments summary is as follows.

```
Resources examined: 15,960
Total time spent: 179.21 minutes
Generated Thumbnails: 14,190
Generated Thumbnail file size: 228MB or ~16KB per thumbnail image
Current Thumbnails Ok, not needing updating: 314
```

Using this summary an estimated run-time on production would be approximately 6 hours based on a current set of roughly 33,000 resources on production that are type 'StillImage', 'Text', and 'MovingImage'.