## Background & Purpose

JPEG 2000 (ISO/IEC 15444-1) is an image compression standard created in 2000 by the Joint Photographic Experts Group, intended as an evolution and improvement from the existing 1992-ratified JPEG image standard.  The standardized file extension for conforming files is *.jp2*.  The mime-type is *image/jp2*.

> *While there is a modest increase in compression performance of JPEG 2000 compared to JPEG, the main advantage offered by JPEG 2000 is the significant flexibility of the codestream. The codestream obtained after compression of an image with JPEG 2000 is scalable in nature, meaning that it can be decoded in a number of ways; for instance, by truncating the codestream at any point, one may obtain a representation of the image at a lower resolution, or signal-to-noise ratio – see scalable compression.[1]*

Although it has taken over a decade to gain wide acceptable, JP2 has made significant strides more recently within the cultural heritage and digital preservation communities. The format's characteristics and sustainability have been thoroughly analyzed by the Library of Congress[2], and organizations such as the Wellcome Trust Digital Library, Kings College, London are using JP2 as both a preservation and access file format.  In fact, their recommendation and case study on the matter, with some modifications, could serve a useful model to consider as a possible implementation scenario for RUcore.[3]  Additionally, Harvard University Library currently recognizes JP2 as a recommended deposit format for faculty and students depositing their works into the university's Digital Repository Service (DRS).[4]

This specification will explore the areas in RUcore that could benefit from the JPEG 2000 image format.  These areas will include:

- Creating JPEG2000 files - encoding
- Possible future use as the archival format for image based objects (pending resolution of certain known technical issues by the Joint Photographic Experts Group in an amended ISO standard)
- JPEG2000 Libraries
- JPEG2000 Servers for presentation
- High resolution image viewers
- Page turner for multipage objects

JP2 is a format that best serves digital surrogate type objects; that is, documents that have been scanned from an analog source.  In particular, historic maps, large bound volumes, printed photographs and manuscripts would be ideal candidates for serving as JP2-formatted content.  Born-digital items such as ETDs, recent-vintage faculty deposits and other items for which a digital file was the source would still be better served by their existing, often native formats (e.g., PDF).

Having established this, there is also significant investment required – in both deploying hardware and developing and implementing software – in order to effect a transition from RUcore's existing multi-format image delivery platform to one which utilizes JP2.  New pipeline methods for encoding JP2 files, delivering viewable content based on those new files, and providing searchable text generated through Optical Character Recognition (OCR) where appropriate will all need to be explored.

It should also be noted that for most born-digital objects ingested into and served by RUcore, other presentation formats (such as PDF) will continue to be the most practical and efficient method of content delivery.

## Creating JPEG 2000 files - encoding

There are a number of different ways to generate the requisite jp2 files from our existing content, as well as any new content we digitize going forward:

- Through the use of server-side software for large batch processing of existing images, as well as conversion of any user-uploaded content,
- Batch processing through desktop/client-side software
- Direct generation of JPEG2000 images as part of the image acquisition workflow

Realistically, a combination of all three options listed above could be implemented, and the capability for desktop/client-side and server-side options should be deployed. However, the option for server-side encoding should be made available in a manner similar to our current image pipeline for RUcore, in which uploaded TIFF files are converted to appropriate presentation formats.

At present, there is an increasingly wide variety of software packages within all three paradigms which support encoding images in JP2 format, including:

- Adobe Photoshop (Desktop: Manual, batch processing)
- Irfanview (Desktop: Manual)
- The GiMP (Desktop: Manual - additional plugin required)
- Apple Preview and Automator (Desktop: Manual, batch processing – Mac Only)
- JRA Publish (Desktop: Batch processing – Windows only)
- ImageMagick (server-side processing; can be used on desktops via command line)
- GraphicsMagick (server-side processing)
- Kakadu (server-side processing)

## Potential for future use as the archival format for image based objects

With a shift to JPEG2000 as an access format, there is a compelling case to be made for using the lossless compression profile for JP2 as our preservation format for still images. The advantage for making such a change rests primarily in a substantial reduction of resource overhead on our storage platform. In addition to the significant advantage JPEG2000 lossless compression affords over our current utilization of TIFF with LZW compression[5], the use of JP2 could also mean that our preservation datastreams could be used to easily and directly generate access copies on the fly.

Unfortunately, significant technical issues present a hurdle that complicates the idea of adopting the format for preservation purposes. The primary issue revolves around JP2's limited accommodations for ICC color profiles and color calibrations, of which RUL makes extensive use of as part of the digital curation workflow to ensure color accuracy in scanned objects and still images. JP2 does not fully support the complete ICC profile specification, relying instead on a limited and incomplete subset.[6]

These limitations would mean that we would be unable to guarantee to a satisfactory degree that color data had not been altered when migrating our existing TIFF archives to JP2. Nor could we reasonably guarantee the color accuracy of

any new objects going forward as we convert them from native scanner and digital camera formats (mostly derived from TIFF) to a JP2 datastream.

Having said this, there is still no reason why we should rule out the use of JP2 for access purposes.  Further, there does appear to be serious interest by the JPEG committee in addressing the shortcomings of the present JPEG2000 standard, and amend it to create a standardized way for all software platforms supporting JP2 to embed and make use of the entire ICC profile spec.  These developments should be watched closely, and there could be a possibility in the near future that an updated JP2-lossless spec could better justify the use of the format for preservation purposes.

## JPEG2000 Libraries

Several JPEG 2000 libraries are currently available.  The three that appear to have the most promise are Jasper, Kakadu, and OpenJPEG.  The following is a simple chart of those libraries feature compared:

| Library | Language | License | Last Release | Advanced Read/Write |
|---------|----------|---------|--------------|---------------------|
| JasPer | C | MIT License-style | 2007-01-19 | No |
| Kakadu | C++ | Proprietary | 2010-10-06 | Yes |
| OpenJPEG | C | BSD | 2011-01-02 | Unknown |

Both the Internet Archive and the IIPImage project identify Kakadu as the best performing library of the three with respect to speed and full image support.

## JPEG2000 Servers for presentation

A JPEG2000 server is needed to provide the sections, or tiles, of the image being viewed.  Ideally the server software that is chosen should be compatible with both the high resolution image viewer and page turner.  Other items should be taken into consideration; they are outlined in the comparative matrix below.

| Server | License | Last Release | OS Support | Library |
|--------|---------|--------------|------------|---------|
| Djakota | LGPL | 2009-06-18 | Linux, Solaris, OS X, Windows | Kakadu |
| Aware ArchivePack | Proprietary | ? | Linux, Solaris | Variants of JasPer |
| IIPImage | GNU v3 | 2011-4-13 | Linux, OS X, *BSD, UNIX, Debian, Windows | Kakadu |
| Lizardtech ExpressServer | Proprietary | ? | Windows, RedHat | ? |
| LuraWave JP2 Image Content Server | Proprietary | ? | Linux | ? |

## High resolution image viewers

There are several options for viewing JPEG 2000 images.   Some important criteria to consider are:

- JavaScript solution, HTML 5
- Embeddable

- Zoom
- Multiple screen resolutions and metrics support

| Viewer | HTML5 | Zoom | Embeddable | Server support | License | Last Release |
|--------|-------|------|------------|----------------|---------|--------------|
| **IIPMoo Viewer** | Yes | Yes | Unknown | Multiple | GNU v3 | 2011-08-06 (beta) |
| **Zoomify** | Yes | Yes | Yes | Multiple | Proprietary | unknown |

*IIPMooViewer example*:

Magna Carta by Los Alamos National Laboratory Research Library – http://african.lanl.gov/adore-djatoka/

*Zoomify example*:

Cancer Images Database - http://cancerimages.nci.nih.gov/caIMAGE/simple.jsp

## Page turner for multipage objects

For multipage objects presentation has been limited to PDF or DjVu file formats. While those have been sufficient under most circumstances, there is always room to improve the user experience in this area. Providing a page turner that can tightly integrate into a web browser without the need for plug-ins, Java applets or third party software is highly desirable.

The Internet Archive BookReader is worth exploring in this area. It is an actively developed project supported by archive.org. This reader has a rich set of features that include:

- JavaScript solution, HTML 5
- Single-Page, Two-page, and Thumbnail view
- Zoom
- Right-to-left page progression (e.g. for Yiddish and Chinese)
- Full-text search with highlighting of search results
- Support for foldouts and variable page size
- In-Browser Text-To-Speech
- Embeddable
- Works with a variety of image servers, or a simple directory of images
- Support for tablet devices, page swiping

For more information please visit the following website. http://openlibrary.org/dev/docs/bookreader

The Internet Archive also provides the steps on how to deliver books stored in the JPEG 2000 file format to the IA BookReader.  For more information on this please visit the following site.

http://raj.blog.archive.org/2011/03/17/how-to-serve-ia-style-books-from-your-own-cluster/

The IA BookReader should also be explored as a high resolution image viewer solution.  If BookReader provides an acceptable level performance and features, supporting a single presentation tool for both user situations would be desirable.

## Moving Forward

We will investigate using JPEG2000 as a presentation format.  To do this we will research the feasibility of using the Djakota server with the appropriate JPEG2000 library to serve JPEG2000 images.  We will research the delivery of both single images and multi-page images using the IIPMoo Viewer and the Internet Archives BookReader respectively.

After a thorough investigation is completed a recommendation will be provided.  If the recommendation is accepted an implementation specification will be drafted and a timeline for delivery will be determined.

## References

[1] http://en.wikipedia.org/wiki/JPEG_2000#Aims_of_the_standard

[2] http://www.digitalpreservation.gov/formats/fdd/fdd000143.shtml#sustainability

[3] Buckley, Robert.  "JPEG 2000 as a Preservation and Access Format for the Wellcome Trust Digital Library" KDCS Digital Consultancy. http://library.wellcome.ac.uk/assets/wtx056572.pdf

[4] Harvard University Library. DRS: Recommended File Formats.
http://hul.harvard.edu/ois/digpres/guidance.html#recommend

[5] Khademi A, Krishnan S. "Comparison of JPEG 2000 and Other Lossless Compression Schemes for Digital Mammograms." Conference Procceedings, IEEE Eng Med Biol Soc. 2005;4:3771-4.
http://www.ncbi.nlm.nih.gov/pubmed/17281050

[6] van der Knijff, Johan. "JPEG 2000 for Long-term Preservation: JP2 as a Preservation Format" D-Lib Magazine, May/June 2011, Vol. 17 No 5/6. http://www.dlib.org/dlib/may11/vanderknijff/05vanderknijff.html