Single or Multiple Digital Objects: Workflow for WMS File Upload and Metadata Structure

Overview

A resource in RUcore repository could have either a single or multiple digital objects associated with it. When archiving the digital resources, handling resources with multiple digital objects becomes a challenge – we need to preserve the information and structure of the original digital files in both the file system and metadata. The current solution is to walk around the issue by packaging the digital file(s) of a resource, single or multiple, into a tar file and archiving as such. This approach buries all the file and file structure information in the tar file, and becomes a problem when, for archival or presentation purpose, original file information as well as the relationships among them are needed, or an individual file in a multi-file package needs to be processed.

Software architecture working group has decided (February 9, 2012) to abandon the practice of packaging archival objects into a tar file. Digital objects will be archived in their original form and structure, individual file information will be recorded in the metadata, and original file structure will be documented in structure map. The change will be applied to all resources, no matter they contain single or multiple digital objects.

This document specifies the workflow and metadata changes to be made in WMS to accomplish the above new approach.

WMS file upload workflow

To accommodate the fundamental RUcore infrastructure changes (abandon tarring of all master files, and allow uploading of multiple master files with or without directory structure), WMS file upload workflow need to be modified. Following changes are proposed:

1. WMS file policy configuration module needs to add the capability for configuring directory as the upload option.

2. In the digital file handling module, make following changes to the workflow to handle multi-file or directory upload:

    A. User must choose whether he/she is uploading individual files or a directory (the two options cannot be mixed).

    B. If directory is chosen, WMS assumes that there could be more than one file type in the directory hierarchy. In this case, WMS identifies the file type by the file extension (in the future, by using utilities like exifTool or mediaInfo). The file types contained in the directory hierarchy must exist in the file policies specified in the WMS configuration module. WMS will upload all the acceptable files contained in the directory, create a structure map that records the original directory structure (specs for the structure map is in a separate document), and create appropriate metadata (specs in later sections).

There is a difference between server and local upload of a directory: for server upload, user can directly select a file directory and upload it; for local PC upload, user must package the directory into a zip or tar file first before uploading.

C. If user uploads a zip file (or tar, or any other file that is created by packaging a set of files or a directory, abbreviated as zip for short in the text that follows), WMS will process the file as follows:

   i) If the same file type is specified by the user (tar, zip, gz, etc.), WMS will upload the file as is, no further processing will be done.

   ii) If a different (non-zip) file type is specified by the user, such as tiff, pdf, etc., WMS will treat the zip file as a directory that contains only the specified file type. The zip file will be extracted, and the extracted directory will be processed as in B.

D. If individual, non-zip file is chosen, WMS will process the files as instructed. Multiple files will be processed sequentially and be put into system as well as metadata. A structure map will also be generated, in which there will just be a record of flat sequence of all the files.

E. User should be able to review the structure map generated at the upload status page.

<u>RUL techMD</u>

RUL technical metadata record the technical details of the digitization process and the digital characteristics of the resulting digital materials. If multiple digital objects are generated, multiple set of technical metadata must exist that each describes the corresponding digital object. To reduce catalogers' workload and potential errors, the goal is that generation of technical metadata should be automated as much as possible. As the first step, WMS in release 7.0 should automatically generate a technical metadata section for each digital object uploaded, with only the content model field populated. All the other fields should still be manually filled out by the cataloger using WMS cataloging utility. In the future release, it has been proposed that WMS use the utilities such as exifTool and mediaInfo to automatically populate most of technical metadata fields (specs pending).

In release 7.0, WMS should make following changes:

1. Stop the current practice of automatically copying to technical metadata following information for individual digital file:
   mimetype
   file size
   checksum
   checksum method

   The above information should be recorded in fileSec (METS) or datastreamVersion and datastreamVersion/contentDigest (FOXML).

2. When a user has successfully uploaded a set of digital files, WMS automatically generates a technical metadata section for each digital file. The technical metadata will contain only content model field populated with correct value.

3. In the WMS cataloging utility, technical metadata section should provide a dropdown list of uploaded digital files. Cataloger must choose a digital file before entering metadata. This will ensure the correct linkage between the digital file and the technical metadata describing it.

4. If a user deletes a digital file, WMS should automatically remove the technical metadata section that corresponds to that specific file.

5. Linking the techMD to the corresponding digital objects is done with ADMID in METS and fedora:isMetadataFor in RELS for FOXML, as described below.

METS fileSec metadata

Though METS is no longer the xml format we use in RUcore repository, it is the internal metadata xml format for WMS. The core functionalities of WMS depend on it, and other formats that WMS outputs (e.g., Fedora FOXML) are derived from this internal METS xml. It is therefore vital that WMS knows how to create METS metadata so that all file and file-metadata relationship information about the digital objects are preserved.

In METS, basic information about each digital file (e.g., mimetype, filesize, and checksum) as well as the file-file and file-metadata relationships is recorded in fileSec. WMS should output METS following the METS specification about the use of fileSec. The output METS metadata should include a listing of all the digital files. As a reference, following is a list of attributes in fileSec and its subelements:

fileSec	**fileGrp**	**file**	**FLocat**
@ID	@ID	@ID	@ID
	@ VERSDATE	@SEQ	@USE
	@ ADMID	@OWNERID	@LOCTYPE
	@ USE	@ADMID	@OTHERLOCTYPE
		@DMDID	@xlink:simpleLink
		@GROUPID	
		@USE	
		@BEGIN	
		@END	
		@BETYPE	
		@MIMETYPE	
		@SIZE	
		@CREATED	
		@CHECKSUM	
		@CHECKSUMTYPE	

WMS should do the following:

1. Use top level “fileGrp” for archive type, indicated by the value of attribute “USE”. e.g.,

   ```
   <fileGrp ID="ARCH" USE="master">
   <fileGrp ID="THUMB" USE="thumbnail">
   ```

2. Use “file” and “FLocat” to document file information and location. Repeat “file” element under each “fileGrp” for multiple digital objects of the same archive type.

3. Use “OWNERID” in “file” element to store original file name (this will be translated into ALT_ID when converted to foxml).

4. ADMID in “file” element is the techMD ID that describes this digital object.

5. GROUPID in “file” element represents the archival object grouping.

6. Example (dummy xml, not a schema):

   ```
   <fileGrp ID="ARCH" USE="master">
           <file ID="ARCH-TIFF-1" SEQ="1" ADMID="TECH-1.11" DMDID=”DMD-1”
           OWNERID="bk_page1.tiff" GROUPID="xxxx" MIMETYPE="image/tiff" SIZE="xxxx"
           CREATED="xxxx" CHECKSUM="xxxx" CHECKSUMTYPE="xxxx">
                   <FLocat ID="???" LOCTYPE="URL" href="ttp://localhost/dummy1.tiff"
                   title="dummy title 1"/>
           </file>
           <file ID="ARCH-TIFF-2" SEQ="1" ADMID="TECH-1.12" DMDID=”DMD-1”
           OWNERID="bk_page2.tiff" GROUPID="xxxx" MIMETYPE="image/tiff" SIZE="xxxx"
           CREATED="xxxx" CHECKSUM="xxxx" CHECKSUMTYPE="xxxx">
   ```

```
                <FLocat ID="???" LOCTYPE="URL" href="ttp://localhost/dummy2.tiff"
                title="dummy title 2"/>
        </file>
</fileGrp>
<fileGrp ID="PRES" USE="presentation">
        ……
</fileGrp>
……
```

FOXML metadata

FOXML is the xml format used in RUcore repository, and is generated in WMS by converting METS into FOXML when user ingests a resource into Fedora. The conversion process needs following modifications to accommodate the changes in multiple digital file handling:

1. In FOXML, metadata and digital objects are all treated as datastreams. When multiple digital objects occur under fileSec in METS, they should be simply converted into multiple datastreams in FOXML.

2. The relationships between digital objects and the metadata that describe them (indicated by ADMID in METS), as well as the relationships among different digital objects (indicated by GROUPID in METS), need to be translated into Fedora representation. In Fedora, when FOXML is used, the relationships between datastreams are specified in object-to-object relationship metadata (RELS). Current WMS has RELS-INT metadata to specify the relationship among different metadata sections. To preserve the relationships mentioned above, WMS could add to the existing RELS-INT metadata section a set of RELS relationship descriptions illustrated below:

   A. Metadata – digital file relationship:

   ```
   <rdf:Description rdf:about="info:fedora/fedpid:xxxxx/TECHNICAL1">
           <fedora:isMetadataFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH1"/>
           <fedora:isMetadataFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH2"/>
   </rdf:Description>
   <rdf:Description rdf:about="info:fedora/fedpid:xxxxx/TECHNICAL2">
           <fedora:isMetadataFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH3"/>
           <fedora:isMetadataFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH4"/>
   </rdf:Description>
   ```

   B. Digital file – digital file relationship:

   ```
   <rdf:Description rdf:about="info:fedora/fedpid:xxxxx/JPEG1">
           <fedora:isPresentationFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH1"/>
   </rdf:Description>
   ```

```
<rdf:Description rdf:about="info:fedora/fedpid:xxxxx/THUMB1">
	<fedora:isThumbnailFor rdf:resource="info:fedora/fedpid:xxxxx/ARCH1"/>
</rdf:Description>
```

RULIB metadata – *Added by Chad Mills 2/12/2013*

The RULIB administrative metadata will begin to be stored under separate sections using the following root element names:

- RULTechMDDocument
- RULRightsMDDocument
- RULSourceMDDocument

After R7.0, a specification will be prepared for developing a RULIB metadata schema and in that specification more detail will be given about the implementation of this change.