# Validation System

Our new validation system will be based on Kurt Whitsel's system that currently performs image processing, validation, and derivative creation for DIDO.

Some old workflow diagrams (file:///\blLdlpEuterpe/dlod/SystemDocumentation/IMAGE_PROCESSING/image_doc/index.html) describing this system are available. It is unclear how accuratly they represent the existing system. (Note that in the containing directory (file:///\blLdlpEuterpe/dlod/SystemDocumentation/IMAGE_PROCESSING/image_doc/index.html) there are some unlinked GIF files that may also be useful.)

David has been happy with Spring for managing flow in a web application. It is possible that we should use it in the validation system.

We need to see if the RIFF project's validation system can be integrated with ours (see their OpenRepositories slides).

## User Requirements for new system

1. Staff can easily add/modify validation criteria for a class of documents.
2. Staff can specify document classes that are eligible for ingest into a collection.
3. Digitizers can easily submit files (individually or in groups) for validation.
4. When processing a large number of files for a single collection, digitizers do not have to login and/or describe the file properties multiple times.
5. Digitizers should be notified of problems as early in the process as possible, so they don't have to enter a lot of extra data or wait a long time to find out their files need to be changed.
6. Digitizers can validate files without having them ingested.
7. Digitizers can specify that validated files should be ingested.
8. Programs (like a cataloging tool) can perform validation.

## Functional Requirements for new system

1. [Must] To properly handle internal structure, items need to be validated on multiple levels:
    - The individual media files must all meet the digitization criteria
    - The metadata must validate against a schema and must meet any collection-specific criteria
    - The metadata and media files must correspond.
2. [Must] The validation system had different profiles (or different applications) for different contexts. Files are validated differently when the context is derivative creation, Fedora ingest, or preservation integrity checking.
3. [Want] The vaildator is converted to an MVC architecture, so actions can be performed independent of the GUI.
4. [Want] Files can be validated multiple times without any side effects.
5. [Want] The validation module is completely separate from the ingest module. The ingest module can assume that files have been validated.

## See also

- Minimum Object Metadata
- Minimum Image Requirements
- Minimum Sheet Music Requirements

## Misc notes

- Current validation system takes a lot of work to configure for a new collection.
- Brian's "Manual QC script" allows running a profile with a perl script to get information about the files.
- If we deal with DIDO or archives collections, we will have to include a more complex workflow system.

## Validation tool outline

Steps a validation tool must perform, and their relationship to existing tools:

1. File submission, both media and metadata
    - for now, just a directory things are copied into, later maybe a webapp
    - imageproc can be called at a lower level to bypass its interface
2. Indication of post-submit action?
    - imageProc does this after something passes
3. Individual file validation
    a. media file properties
        - Imageproc does this, but it needs to be more configurable
        - Brian's manual QC script is a good model for the configuration
    b. metadata file content
        - Xubmit
    c. filename (follows rules and specifies correct file type)
        - imageproc does this, may need more configuration
4. File relationships (package) validation
    a. pages in sequence
        - imageproc does some
    b. proper file correspondence (# of derivatives, one PDF per book, etc.)

        - ingest tool does some
        - imageproc does some
5. If ingesting, move files to ingest directory
6. Post-ingest validation of the Fedora object
    - ingest tool
7. Result reporting (to where? who?)
    - always tell the user who submitted the file
    - do we log all the workflow information that imageproc does?

## Open questions

- (Jon, Jenn) How/what technical and process history metadata should be stored? Does this come from this tool, or from a workflow tool?