Digital File Policy Management:

Configuration Module Enhancement and Inter-application Policy Sharing

Overview

Digital file policies determine how digital files are processed before being deposited into RUcore repository. WMS has a built-in module for configuring the software so that the file policies set by the decision-making bodies are enforced in the workflow. This document specifies the requirement for further expanding the WMS file policy configuration functionalities, mainly to add collection level file policy customization capability, to improve the configuration workflow, and to provide the service for other RUcore software constituents to share the policies set in WMS.

Enhancement for Policy Configuration Module

The enhancement includes adding the capability of collection level file policy customization, improving the configuration workflow for better efficiency and user experience, and adding generic display label as a configuration option.

Collection Level File Policy

The current WMS configuration module allows configuring system-wide digital file policies. The key word here is “system-wide”. File policies set for each content model will apply to all collections. This becomes a problem for certain collections that have special needs for file types allowed and file processing procedures different from the rest. Examples: collections in the Faculty Deposit and ETD.

WMS works around this issue by either creating a “special content model” to cover the special need of certain collections (e.g., ETD), or embedding some file type filtering logic into the software code that works only for collections with special requirements (faculty deposit). The drawbacks of these approaches are obvious: Adding “special content model” breaks the concept of content model employed in RUcore repository, and the hard coded policy filter in the software destroys the design principle of WMS, and has been proven to be extremely error-prone and hard to maintain.

To solve the problem, WMS needs following changes:

1. System-wide digital file policies become the default policies for all collections.
2. Add the capability of configuring collection-level digital file policies.
3. Collection-level policies, if set, override the default policies.
4. Collection-level policies should obey the collection hierarchy principles – children collections inherit parent collection policies unless a different set of policies have been set specifically for a child collection.
5. WMS digital file handling module needs to be modified to implement the above policy scheme changes.

Policy Configuration Workflow Enhancement

The workflow of WMS file policy configuration module needs to be simplified. One currently needs to go through and set policies for every single one of all file types for all archive types in all content models. When the number of content models and file types grow, especially when software installation and migration must take place and fresh new policies need to be set, this becomes an overwhelming task, and has been proven to be very error-prone.

Proposed WMS changes:

WMS file policy configuration module provides a hierarchical policy tree as follows:

1. How each possible archival file type (or more technically, archiveType–generation–fileType combinations) should be processed is called “File Processing Rules”. The rules that will apply to most of content models are called “default rules”, and they need to be manually set, but only once for each archival type. The default rules for all archival file types become a “pool of default rules” and will be later used for configuring content model file policies.

   Example:
   Rule for master-original-tiff
   Rule for master-original-wav
   Rule for presentation-pdf
   Rule for thumbnail-jpeg
   ……

2. If a different set of rules must be specified for certain archival types in certain content model, WMS should allow setting of “named rules” and form the “pool of named rules”.

3. Setting file policies for each content model becomes simply selecting from existing rules from the pools created in step 1 and/or 2.

<u>Add Generic Datastream Label as Configuration Option</u>

RUcore and DLR websites need user customizable label for displaying datastream url. WMS should provide following mechanism to allow flexible label creation for each datastream:

1. In WMS configuration module, digital file section, add an option for entering generic label for each archival file type. This entry will be used as the url label if not overridden later by cataloger.

2. In WMS digital file handling module, add a datastream label entry field for each file type to be processed. This field should be an optional entry field. If user entry is found, the value will replace the generic label set in step 1.

Policy Service – Sharing the Policies within RUcore Software Suite

<u>Digital File Policies</u>

Digital file policies are used not just in WMS, but in other applications in the RUcore software suite as well. DLR application needs to know the allowable datastreams and their mimetypes for its add-datastream functionality. RUcore website get API also needs this information to deliver datastreams to the end user, and so on. Currently each of these applications acts on its own based on programmers' understanding of the policies on paper. This has caused unnecessary duplicate work and has the potential for errors that are difficult to diagnose. The proposed solution is to have a centralized location where digital file policies on paper are converted into software-understandable representation. WMS file policy configuration module, which allows easy configuration of digital file policies, is the logical candidate for undertaking this role.

Changes in WMS:
WMS needs to output following information from file policies to share with other applications:

- Content models approved for RUcore.
- Archival types allowed for each content model.
- File types under each archive type for a given content model.
- Miscellaneous information about each file type, such as mimetype, filename extension, etc.

Digital File Presentation Rules

For each datastream stored in Fedora repository, RUcore website needs to know how to render its information and deliver it to the end user. Currently the rules are manually created and stored in an xml file. It is proposed that the process of setting these rules be consolidated into WMS configuration utility.

Changes in WMS:

A) A new presentation rule configuration module needs to be added to WMS.

B) WMS needs to output these rules to share with RUcore website application.

   WMS must be able to set and share following rules:

   - sequence (does this datastream support sequencing?)
   - separator (what separator is used between dsid and sequence number?)
   - protect (protect from retrieving?)
   - xacml (xacml policy exists for this datastream?)
   - briefList (return in a brief list of datastream information)
   - fullList (return in a full list of datastream information)
   - disposition (instructs how a MIME user agent should display an attached file. Values are 'inline', 'attachment', or 'hidden'.)
   - mediaserver (what media server should be used to render this datastream. Values are 'none', 'wowza', 'darwin', and 'djatoka'.)

Sharing the Digital File Policies and Presentation Rules

To share the most updated WMS file policies with all the other RUcore applications, we proposed following WMS changes:

- Output the digital file policies and presentation rules as an xml file and save it in a common location in the file system accessible by all applications. Whenever changes are made to the file policy in WMS, dumping of the policy xml is triggered.

- The policy xml should be viewable in WMS file policy configuration page.

The xml file will have following structure (dummy xml example, not a schema):

```
<dfPolicies>
 <policyGrp type="ContentModel" name="Book">
  <policyGrp type="TypeOfResource" name="Text">
   <policy archType="master">
    <datastream id="tiff" type="DigitalObject" generation="original">
     <mimeType>image/tiff</mimeType>
     <extension>tif</extension>
     <extension>tiff</extension>
     <presentation>
      <display>TRUE</display>
      <sequence>FALSE</sequence>
      <separator>-</separator>
      <protect>FALSE</protect>
      <xacml>FALSE</xacml>
      <brieflist>FALSE</brieflist>
      <fulllist>TRUE</fulllist>
      <disposition>inline</disposition>
      <mediaserver>none</mediaserver>
     </presentation>
    </datastream>
    <datastream>
       …
    </datastream>
    <datastream>
       …
    </datastream>
    …

   </policy>
   <policy archType="presentation">
    <datastream id="djvu" type="DigitalObject" generation="none">
     <mimeType>image/x.djvu</mimeType>
     <extension>djvu</extension>
    </datastream>
    <datastream id="pdf" type="DigitalObject" generation="none">
     <mimeType>application/pdf</mimeType>
     <extension>pdf</extension>
    </datastream>
    <datastream>
       …
    </datastream>
    …

   </policy>
   <policy archType="searchxml">
    <datastream id="xml" type="DigitalObject" generation="none">
     <mimeType>text/xml</mimeType>
     <extension>xml</extension>
    </datastream>
    <datastream>
       …
    </datastream>
    …

   </policy>
   <policy archType="thumbnail">
    <datastream id="jpeg" type="DigitalObject" generation="none">
     <mimeType>image/jpeg</mimeType>
     <extension>jpeg</extension>
     <extension>jpg</extension>
    </datastream>
    <datastream>
       …
    </datastream>
    …

   </policy>
  </policyGrp>
 </policyGrp>

  <policyGrp >
    ...
  </policyGrp >

  < policyGrp type="ContentModel" name="Audio">
    < policyGrp type="TypeOfResource" name="Sound”>
      < policy archiveType ="master”>
        ...
      </ policy>
      ...
    </ policyGrp >
    ...
  </policyGrp>
  ...
</dfPolicies>
```