

fedora



Creating Disseminators

Chris Wilper, Cornell University
Fedora Training @ Yale University
May 3rd, 2007

General Steps

- Understand the interface of the service
 - How do you invoke it with a web browser?
 - What parameters does it take?
- Create a Behavior Definition Object
- Create a Behavior Mechanism Object
- Create Disseminators on your objects to use these

Installing a Local Fedora

Set up the environment

```
set JAVA_HOME=C:\j2sdk... (whatever) ...
set FEDORA_HOME=C:\fedora-2.1.1
set PATH=%JAVA_HOME%\bin;%PATH%
set PATH=%FEDORA_HOME%\server\bin;%PATH%
set PATH=%FEDORA_HOME%\client\bin;%PATH%
```

Unpack Fedora

```
c:
cd \
jar -xvf Z:\fedora-2.1.1-server.zip
jar -xvf Z:\fedora-2.1.1-client.zip
```

Initialize and start Fedora and McKoi

```
fedora-setup no-ssl-authenticate-apim
mckoi-init fedoraAdmin fedoraAdmin
mckoi-start
fedora-start mckoi
```

<http://memex.cs.cornell.edu:8080/training/install.html>

Getting Familiar

- Open the admin client
 - type “fedora-admin”
 - Password is “fedoraAdmin”
- Create a content model object: “demo:Image”
- ...with a datastream: “IMAGE”
- ...a “Reference” datastream to <http://example.org/>

Create an Instance

- With PID: “demo:FirstImage”
- With content model: “demo:Image”
- With datastream “IMAGE”
- Make it a real image this time...
 - Managed content
 - image/jpeg
 - Import...from File
 - \fedora-2.1.1\server\jakarta-tomcat-5.0.28\...
...webapps\ROOT\demo\simple-image-demo\...
...colesium-high.jpg

Disseminate the Datastream

- Point your browser to:
 - <http://localhost:8080/fedora/get/demo:FirstImage>
- “View Item Index”...click the label to disseminate

Create a BDef

- Admin client:
 - Builders -> Behavior Definition Builder
- General Tab:
 - Use PID: “demo:Thumbnail”
 - Object Name: “Thumbnail”
 - Object Description: “Thumbnail”
 - DC Title: “Thumbnail”

Create a Bdef (Cont'd)

- Abstract Methods Tab:

- New

- `getThumbnail`

- Gets a thumbnail

- Documentation Tab:

- Any label

- Any URL

- `text/html`

- Ingest!

Create a BMech

- Admin client:
 - Builders -> Behavior *Mechanism* Builder
- General Tab:
 - Use PID: “demo:Thumbnail*Mech*”
 - Object Name: “ThumbnailMech”
 - Object Description: “ThumbnailMech”
 - Bdef Contract: “demo:Thumbnail”
 - DC Title: “ThumbnailMech”

Create a Bmech (Cont'd)

- Service Profile Tab:
 - Output MIME Type: “image/jpeg”
- Service Methods Tab:
 - *Look at the docs of service we'll be using:*
 - <http://localhost:8080/imagemanip/>
 - Base URL:
 - [http://localhost:8080/imagemanip/ImageManipulation?url=\(IMAGE\)&op=resize&newWidth=100](http://localhost:8080/imagemanip/ImageManipulation?url=(IMAGE)&op=resize&newWidth=100)
 - Select getThumbnail row...click “Properties”
 - (cont'd)

Create a Bmech (Cont'd)

- Service Methods Tab (Cont'd):

- Method Parameter Definitions:

- Param name: “IMAGE”

- Param type: “DATASTREAM”

- Required?: “YES”

- Pass By: “URL_REF”

- Return Types: “image/jpeg”

- OK

- Datastream Input Tab:

- MIME Type: “image/jpeg”

Create a Bmech (Cont'd)

- Documentation Tab:
 - Label: any
 - URL: <http://example.org/>
 - MIMEType: text/html
- Ingest!
- Close the Bmech builder window

Workaround for “slash” bug

- Export the BMech we just created:
 - File -> Export -> One Object
 - Choose C:\fedora-2.1.1\client
 - Enter the PID: “demo:ThumbnailMech”
- Purge the Bmech:
 - File -> Purge Object -> “demo:ThumbnailMech”
- notepad c:\fedora-2.1.1\client\demo_ThumbnailMech.xml
- Find the trailing slash in the URL to the service
- Hint: search for “imagemanip”

Workaround for “slash” bug (Cont'd)

- Remove the single trailing slash and save the file.
- Re-ingest the fixed object:
 - File -> Ingest -> One Object
 - C:\fedora-2.1.1\client\demo_ThumbnailMech.xml

Create the Disseminator

- Open “demo:FirstImage”
- Go to “Disseminators” -> New
- Behavior: “demo:Thumbnail”
- Mechanism: “demo:ThumbnailMech”
- Add.. choose IMAGE
- Save!

Disseminate the Thumbnail

- Point your browser to:
 - <http://localhost:8080/fedora/get/demo:FirstImage>
- “View Dissemination Index”
- Click the getThumbnail dissemination
- It worked (hopefully!)

Exercise: Grayscale Dissemination

- Create a Bdef to define the behavior:
 - PID: demo:Grayscale
 - Method name: getGrayscale
- Create a Bmech to point it to the Image Manipulation service in a different way...
 - PID: demo:GreyscaleMech
 - Same datastream input requirements...
 - See <http://localhost:8080/imagemanip/>

Exercise: Combined Disseminations

- This time we want a grayscale thumbnail dissemination.
 - Bdef PID: demo:GrayscaleThumbnail
 - Method name: getGrayscaleThumbnail
 - Bmech PID: demo:GrayscaleThumbnailMech
- Disseminator-as-datastream