Get objects

↔

support librarians!

Matthias Gross (Bavarian Library Network)
Importing objects into Digitool is something they just do time by time.
Support Ingest
Support Ingest

Web Ingest - Schritt 3/4: Parameters

Transformers
- File streams that will be loaded with no relationships
- File streams that will be loaded with manifestation relationships
- File streams that will become part of one parent record

Filter on File Extension(s):
- False

Filter on Files smaller than File Size:
- False

Filter on Files greater than File Size:
- False

Share this Metadata?
- No, I want to keep this local
Support Ingest

Pre-Ingest Services
• creating Mets-Files

Ingest Services after Upload
• semi-automatic ingest of newspapers
German law: Libraries are allowed to offer digital versions of their books within their reading rooms
- digitization of own copy
- one concurrent user per physical copy

Great chance to get objects into our repository

⇒ One licensed object per library
⇒ One catalogue record on the network level but many URLs
Catalogue expert

Only 1 URL per catalogue record!

The Solution: URL to a Gateway object
Gate Object

no stream/any stream

usage type: view_main

Partition A: bvb_gate

only the URL of the Gate (manifestation leader) will be passed to ALEPH

Manifestation 1

PDF
IP range limited
holder: Library A

Manifestation 2

METS
IP range limited
holder: Library B

the MARC data are shared

DTL-ALEPH Sync
Gate Object

*no stream/any stream*

*usage type: view_main*

*Partition A: bvb_gate*

The Gate manifestation can be handled by our gateway script ALEPH

the MARC data are shared

DTL-ALEPH Sync

**Manifestation 1**

PDF

*IP range limited*

*holder: Library A*

**Manifestation 2**

METS

*IP range limited*

*holder: Library B*
Gateway viewer = „Find-Appropriate-Manifestation Viewer“

JSP using DigiTool Java Libs
1. get DE by pid (pid from viewer url)
2. get all manifestation usage_type=VIEW
3. traverse DEs
   any access right matches user credentials?
     yes: get the url and redirect
     no: continue
4. finally: display fallback text
fallback page, dynamically created

BVB-Multimediaserver Gateway

By the address you have sent to our server http://bvbm1.bib-bvb.de:1802/webclient/DeliveryManager?pid=31575&custom_att_2=simple_viewer you could have been redirected to objects that were provided by one of the following libraries:

- Deutsches Museum
- Bayerische Staatsbibliothek
- Staatsbibliothek Bamberg
- Universitätsbibliothek Regensburg
- Universitätsbibliothek Augsburg
- UBA

Unfortunately our server could not find any licensed object for your IP address [193.174.96.1].

The libraries listed above are responsible for the definition of their access restrictions. Contact your library if you think you should have been given access from your computer.
Fallback page, dynamically created

- DirectStreamViewerPreProcessor
- URL created by VPP
  
  .../webclient/bvb_gateway.jsp?/2009/02/09/file_1/31575&usePid1...

PID of Gate object
Extending the extension

Some objects presented outside of DigiTool?

Create DigiTool object that contains the corresponding URL
Add access rights metadata on it
Add something in partition a to control to the appropriate viewer

Write a simple redirect-to-URL viewer (perl/cgi)
- reads the URL from the file
- redirects to the URL
Serials in Digitool

Bavarian Library Network Center

Munich, at the day of Mary of Bethany 2009
One PDF per issue

=> We love METS because the navigation by a table of contents on the left hand side seems to be the natural way to present a serial

Right now we have …

University Library of Bamberg
  one weekly journal with access rights

State Library of Neuburg/ Donau
  3 daily newspapers with access rights

Bavarian State Library
  34 serials of statistical reports of the Bavarian State Office for Statistics, free
example for a daily newspaper
.. rather manually (without any magic)?

**First import of already available issues**
- by METS ingest
  - => create a mets file
  - => webingest with access rights if necessary

**Import of new issues**
- by CSV ingest and editing the METS fileSec and structMap
Step 1: creating a CSV file

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dateiname</td>
<td>Title/Label</td>
<td>PID der Zeitung (METIS)</td>
</tr>
<tr>
<td>2</td>
<td>OK-N_2008_07_26.pdf</td>
<td>Donaukurier / Ausgabe Neuburg</td>
<td>1488669</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 173</td>
<td>2008-07-26</td>
</tr>
<tr>
<td>3</td>
<td>NR_2008_07_26.pdf</td>
<td>Neuburger Rundschau Jg. 64,</td>
<td>818866</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 173</td>
<td>2008-07-26</td>
</tr>
<tr>
<td>4</td>
<td>SZ_2008_07_26.pdf</td>
<td>Schrobenhausener Zeitung</td>
<td>81886</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 173</td>
<td>2008-07-26</td>
</tr>
<tr>
<td>5</td>
<td>OK-N_2008_07_28.pdf</td>
<td>Donaukurier / Ausgabe Neuburg</td>
<td>1488669</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 174</td>
<td>2008-07-29</td>
</tr>
<tr>
<td>6</td>
<td>NR_2008_07_28.pdf</td>
<td>Neuburger Rundschau Jg. 64,</td>
<td>818866</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 174</td>
<td>2008-07-29</td>
</tr>
<tr>
<td>7</td>
<td>SZ_2008_07_28.pdf</td>
<td>Schrobenhausener Zeitung</td>
<td>81886</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2008, Nr. 174</td>
<td>2008-07-29</td>
</tr>
</tbody>
</table>

filename, label for the single object, PID of parent object, DTL fileid to be referred to in the METS fileSec
Step 2: CSV ingest with a saved task chain

Step 3: Editing FileSec and StructMap in Meditor

Put in the right reference
**fileSec**: refer to the fileID specified in csv file and create an ID for this file

```xml
- <file ID="f-2008-07-25" MIMEType="application/pdf" GROUPID="g-2008-07-25">
  <FLocat xlink:href="2008-07-25" LOCTYPE="URL"/>
</file>
- <file ID="f-2008-07-26" MIMEType="application/pdf" GROUPID="g-2008-07-26">
  <FLocat xlink:href="2008-07-26" LOCTYPE="URL"/>
</file>
- <file ID="f-2008-07-28" MIMEType="application/pdf" GROUPID="g-2008-07-28">
  <FLocat xlink:href="2008-07-28" LOCTYPE="URL"/>
</file>
```

**structMap**: create a div node and refer to the ID given in the fileSec

```xml
  <fptr ID="DTL356" FILEID="f-2008-07-25"/>
</div>
- <div LABEL="Sa.&#47;So., 26.&#47;27.07.2008" ID="DTL357">
  <fptr ID="DTL358" FILEID="f-2008-07-26"/>
</div>
- <div LABEL="Mo., 28.07.2008" ID="DTL359">
  <fptr ID="DTL360" FILEID="f-2008-07-28"/>
</div>
- <div LABEL="Di., 29.07.2008" ID="DTL361">
  <fptr ID="DTL362" FILEID="f-2008-07-29"/>
</div>
```
This workflow won’t work for metadata larger than 50,000 characters 😞 due to the limited ability of Meditor …

So we were forced to improve the import in November 2008.
How to do this...  

... more automatically and with some magic?

The concept

Providing a script that

a) creates a csv file automatically,
b) starts the ingest,
c) supplements the structMaps and the fileSecs.

First we are going to show the result ...
1. The user uploads files by Webingest
2. calls a script with parameter workingdirectory and username.

2. Start a script, the workingdirectory where the files are uploaded to will be passed
3. fills in a form
   filenames, parent pid, access rights mid are filled in by the script
   other information is filled in automatically (guessed from the corresponding filename) but can be edited
4. clicks the start button for starting the ingest

Start ingesting

Herzlich Willkommen Staatliche Bibliothek Neuburg a.d. Donau!

<table>
<thead>
<tr>
<th>Date for StructMap leaf</th>
<th>DTL- Label for child object</th>
<th>Month and year for StructMap nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DK-NL_2009_02_28.pdf</td>
<td>Donaukurier/Ausgabe Neuburg 2009, Nr. 49</td>
<td>Februar 2009 145669 219226</td>
</tr>
<tr>
<td>DK-NL_2009_03_02.pdf</td>
<td>Donaukurier/Ausgabe Neuburg 2009, Nr. 50</td>
<td>März 2009 145669 219226</td>
</tr>
<tr>
<td>DK-NL_2009_03_03.pdf</td>
<td>Donaukurier/Ausgabe Neuburg 2009, Nr. 51</td>
<td>März 2009 145669 219226</td>
</tr>
<tr>
<td>NPL_2009_02_28.pdf</td>
<td>Neuburger Rundschau Jg. 65, 2009, Nr. 49</td>
<td>Februar 2009 81586 219226</td>
</tr>
<tr>
<td>NPL_2009_03_02.pdf</td>
<td>Neuburger Rundschau Jg. 65, 2009, Nr. 50</td>
<td>März 2009 81586 219226</td>
</tr>
<tr>
<td>NPL_2009_03_03.pdf</td>
<td>Neuburger Rundschau Jg. 65, 2009, Nr. 51</td>
<td>März 2009 81586 219226</td>
</tr>
<tr>
<td>SZ_2009_03_04.pdf</td>
<td>Schrobenhausener Zeitung 2008, Nr. 54</td>
<td>März 2009 81580 72913</td>
</tr>
<tr>
<td>SZ_2009_02_25.pdf</td>
<td>Schrobenhausener Zeitung 2009, Nr. 49</td>
<td>Februar 2009 81588 72913</td>
</tr>
<tr>
<td>SZ_2009_03_02.pdf</td>
<td>Schrobenhausener Zeitung 2009, Nr. 50</td>
<td>März 2009 81588 72913</td>
</tr>
</tbody>
</table>

Ingest mit diesen Werten starten

Start ingesting
5. The user gets a report about success or failure, which he has to check

Finished!
For creating the fill-in form and for performing the ingest you have to know

a) the working load directory

b) the username

You may either ask the user or just take it from the jsp script

digitool-mng.ear/digitool-webingest.war/views/admin/main.jsp

If you are interested in details, see next page. If you aren’t, just skip it.
How to get the directory where the files have been loaded into

In `digitool-mng.ear/digitool-webingest.war/views/admin/main.jsp` there are useful variables:

- `${row.appId}` = ingest ID
- session variable `adminUnitOfLoggedInUser` = Admin Unit
- `${row.assignTo}` = Username
- from javascript you can get the port by `window.location.port` for the DTL slot;

You can mix this together in order to create a call like:

`http://servername_and_port/path_to_script?workingLoadId=/exlibris/dtl/j3_1/digitool/home/profile/units/DTL02/load/load_ing123&user=ITSME`
For the form:

1. order the files by date (in this case simply alphabetical) so that the sequence in the StructMap(s) will be correct

2. provide fields for the branches you have in your structMap
   
   *here two: year and month*

3. provide fields for all information that isn’t shared by all objects
   
   *here parent-PID, accessrights-mid*

4. any information that will be shared by all objects is defined in the mapping.xml
How to do a little magic

Processing the fill-in-form ...
1. create the values.csv file in the working directory
2. copy some files into the working directory:
   - mapping.xml
   - ingest_settings.xml
3. start tasker.sh by calling the shell
4. start a script that is adding the entries to fileSec and structMap
   the necessary information about the branches and the DTL-fileIDS can be taken from values.csv
We are processing these structural metadata by help of soap calls.
Have a look at Repository Web Services in the General Configuration Guide
This is what we are currently working on to present digitized serials.

Our idea:
1. Do one METS-Ingest per issue
2. Provide a HTML page where you provide the links to the single issues. They should open in a new window. Maybe you use handles instead of urls, so that the links will be persistent.
3. Alternative: Provide a METS file for the TOC of the serial. On the right side the link or handle to the issue will be shown. Extending the viewers you can do a lot of magic things.

**What we would love**

The capability of DigiTool to handle METS that is referencing METS-objects. Making Ex Libris to provide an ingest that will add files to METS-Objects. Will we ever be able to convince Ex Libris??? Perhaps we can manage that together.