CS5604: Information Storage and Retrieval

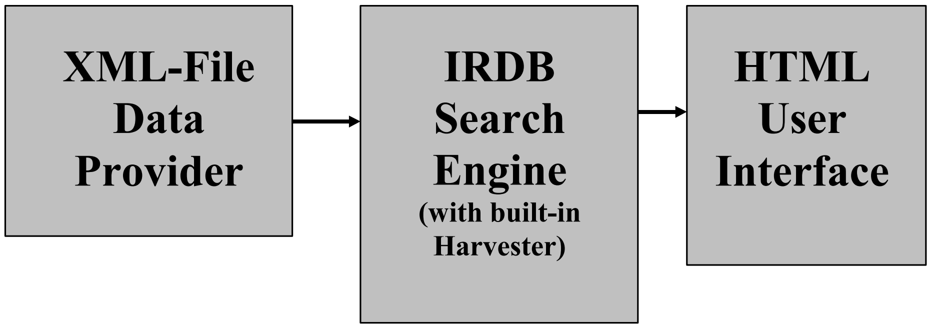
OAI / ODL Component Composition Exercise

(Be sue to read all the README files for individual component!)

This is an in-class exercise to demonstrate the installation, use, and composition of Open Archives and Open Digital Library components. It comprises the following steps:

1. Installation of a simple Open Archive (XMLFile)
2. Harvesting from an Open Archive
3. Connecting an Open Archive to a search engine component (IRDB)
4. Adding a simple WWW interface to the search engine component

The goal of the exercise is to create a simple digital library system out of components, with the following system architecture:



The exercise requires telnetting (or sshing) into "somemachine.your.domain" with the login name "yourloginname" and password "yourpassword". You will use your unique pid wherever reference is made to a "userid" and all your work will be done in the subdirectory "/home/yourloginname/public_html/cgi-bin/<userid>".

Before you starting doing this exercise, make sure your Apache is configured correctly.

1. Perl module installed (Perl5 needed).
2. suexec support:

"Upon startup of Apache, it looks for the file "suexec" in the "sbin" directory (default is "/usr/local/apache/sbin/suexec"). If Apache finds a properly configured suEXEC wrapper, it will print the following message to the error log:

```
[notice] suEXEC mechanism enabled (wrapper: /path/to/suexec)
```

If you don't see this message at server startup, the server is most likely not finding the wrapper program where it expects it, or the executable is not installed *setuid root*."
For more detail, see http://httpd.apache.org/docs/suexec.html

3. public_html user directory support

set public_html is the default directory to put webpage and cgi script. Change the public *html and cgi-bin* permission to 755. Uncommented the following lines in httpd.conf

```
<IfModule mod_userdir.c>
   UserDir public_html
</IfModule>
```

4. ExecCGI on a per-user basis for user-level CGI applications. ExecCGI : Execution of CGI scripts is permitted. Uncommented the following lines in httpd.conf

```
<Directory /home/*/public_html/cgi-bin>
    Options ExecCGI FollowSymLinks
    SetHandler cgi-script
</Directory>
```

After logging in for the first time, create a user directory for yourself by running:

```
mkdir ~/public_html/cgi-bin/<userid>
```

The exercise also requires the use of a WWW browser. Internet Explorer is preferred but Netscape Navigator will suffice.

Comments about some of the components and tools are in italics. These are purely informational and not part of the sequence of steps for the exercise.

After the exercise I ask that you please fill out a simple survey that will assist our research efforts in componentized software for digital libraries. Thanks !

Hussein Suleman 15 September 2002

---

1. Installation of a simple Open Archive (XMLFile)

XML-File is a data provider module that creates an OAI-compliant repository out of a set of XML files that contain the metadata. Using this requires minimal effort while retaining all the flexibility of the OAI protocol.

1.1. Installation

We will install the component into a directory from which the scripts can be executed using CGI (the Common Gateway Interface protocol used by web servers to execute dynamic scripts).

Change to "~/public_html/cgi-bin/<userid>" where <userid> is your userid e.g., user01

```
cd ~/public_html/cgi-bin/<userid>
```

Download the component from the OAI-VT website if you don't already have it

```
wget http://www.dlib.vt.edu/projects/OAI/software/xmlfile/OAI-
XMLFile-2.0.tar.gz
```

Decompress the file

```
gzip –cd OAI-XMLFile-2.0.tar.gz | tar –xf -
```

We will use the default test data for this exercise so no configuration is necessary.

1.2. Testing

1.2.1. Direct execution

First we can test by directly invoking the script to see if the script executes without any errors. Change to the "OAI-XMLFile/XMLFile/test1" directory

```
cd OAI-XMLFile/XMLFile/test1
```

and run the following command:

```
QUERY_STRING='verb=Identify' ./oai.pl
```

You should see the XML response to the Identify OAI-PMH service request.

1.2.2. Internet Explorer

Run Internet Explorer and type in the following URL:

```
http://rocky.dlib.vt.edu/~yourloginname/cgi-bin/<userid>/OAI-
XMLFile/XMLFile/test1/oai.pl?verb=Identify
```

You should get the same response as before.
(This also works in Netscape 6 but you have to "View Source" to see the output nicely formatted.)

1.2.3. Repository Explorer (RE)

The Repository Explorer is a tool for testing Open Archives. You can issue individual commands and validate the results (using XML Schema). You can also perform a sequence of automatic tests.

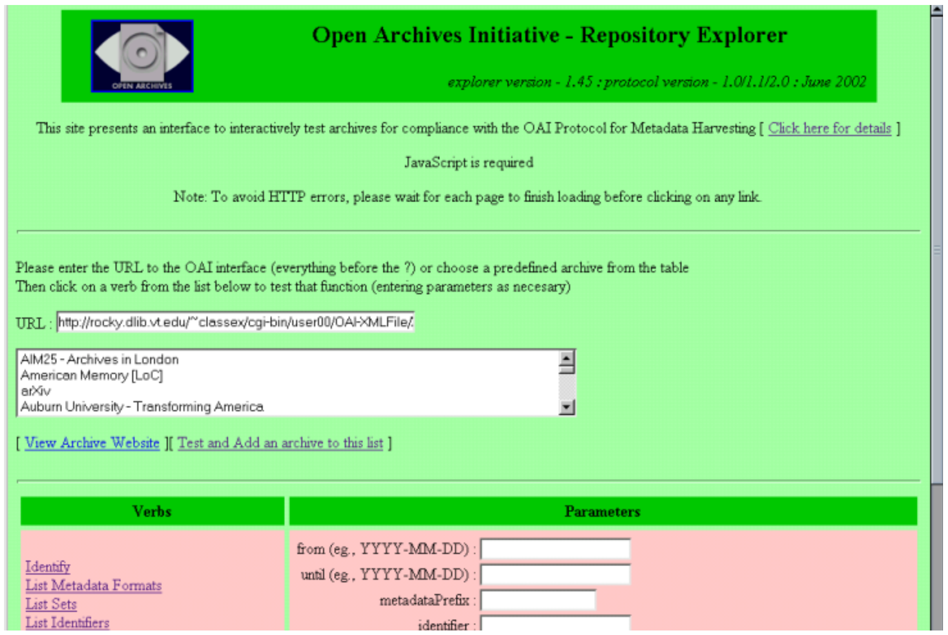
Open the Repository Explorer in a browser window using the following URL:

```
http://purl.org/net/oai_explorer
```

The full baseURL that is needed to access your Open Archive from the WWW is

```
http://rocky.dlib.vt.edu/~yourloginname/cgi-bin/<userid>/OAI-
XMLFile/XMLFile/test1/oai.pl
```

Enter this baseURL in the RE and click on Identify, as shown below:



Open Archives Initiative - Repository Explorer

explorer version - 1.45 : protocol version - 1.0/1.1/2.0 : June 2002

This site presents an interface to interactively test archives for compliance with the OAI Protocol for Metadata Harvesting [Click here for details]

JavaScript is required

Note: To avoid HTTP errors, please wait for each page to finish loading before clicking on any link.

Please enter the URL to the OAI interface (everything before the ?) or choose a predefined archive from the table
Then click on a verb from the list below to test that function (entering parameters as necesary)

URL : http://rocky.dlib.vt.edu/~classex/cgi-bin/user00/OAI-XMLFile/

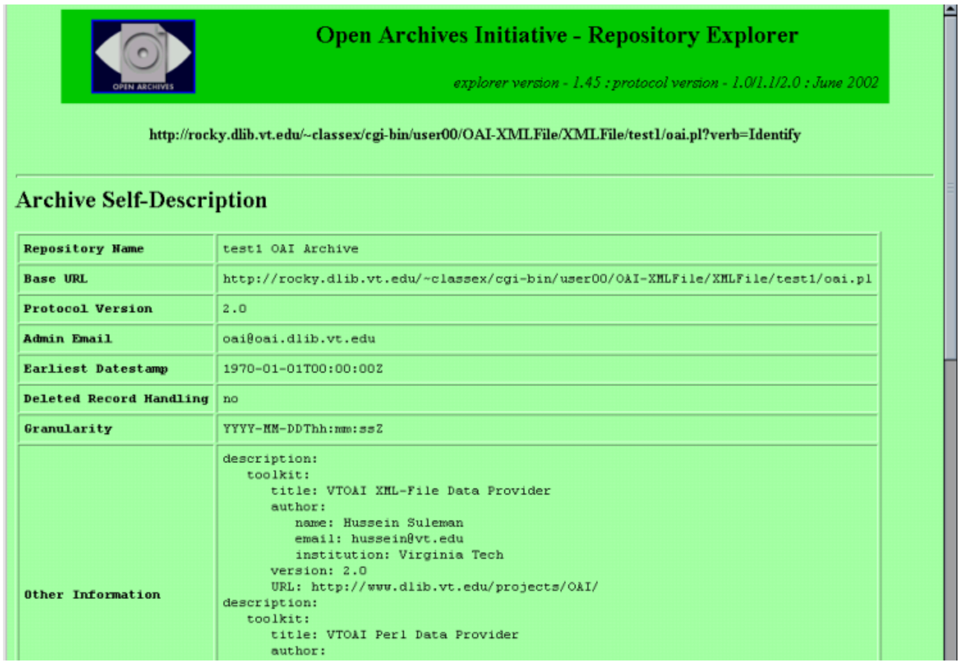
AIM25 - Archives in London
American Memory [LoC]
arXiv
Auburn University - Transforming America

[View Archive Website][Test and Add an archive to this list]

Verbs	Parameters

Identify
List Metadata Formats
List Sets
List Identifiers
	from (eg., YYYY-MM-DD) :
	until (eg., YYYY-MM-DD) :
	metadataPrefix :
	identifier :

You should then see a formatted version of the Identify response as shown below:



Open Archives Initiative - Repository Explorer

explorer version - 1.45 : protocol version - 1.0/1.1/2.0 : June 2002

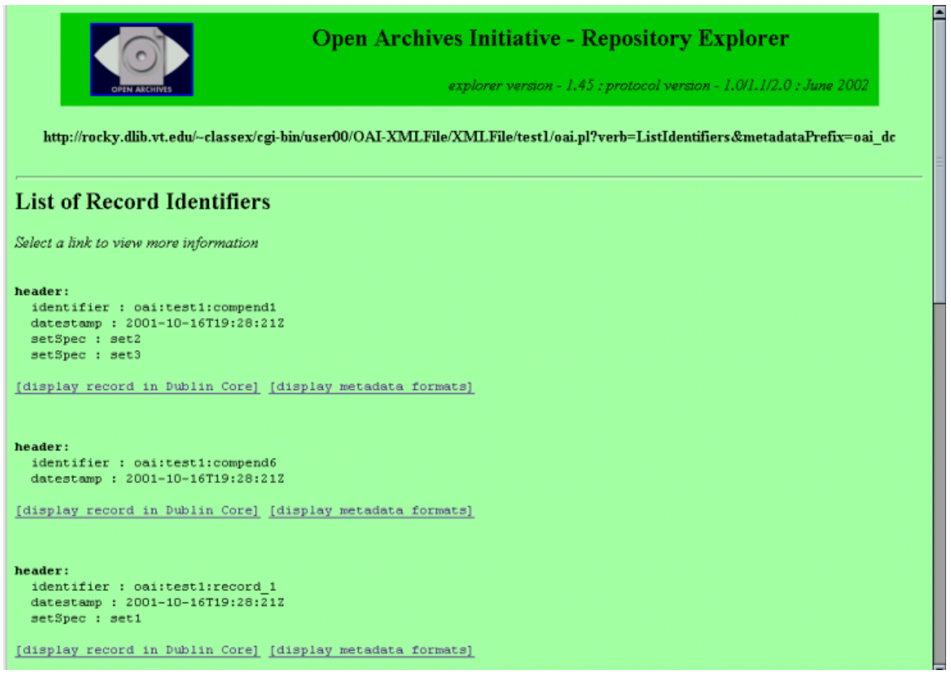
http://rocky.dlib.vt.edu/~classex/cgi-bin/user00/OAI-XMLFile/XMLFile/test1/oai.pl?verb=Identify

Archive Self-Description

Repository Name	test1 OAI Archive
Base URL	http://rocky.dlib.vt.edu/~classex/cgi-bin/user00/OAI-XMLFile/XMLFile/test1/oai.pl
Protocol Version	2.0
Admin Email	oai@oai.dlib.vt.edu
Earliest Datestamp	1970-01-01T00:00:00Z
Deleted Record Handling	no
Granularity	YYYY-MM-DDThh:mm:ssZ
Other Information	

```
description:
   toolkit:
      title: VTOAI XML-File Data Provider
      author:
         name: Hussein Suleman
         email: hussein@vt.edu
         institution: Virginia Tech
      version: 2.0
      URL: http://www.dlib.vt.edu/projects/OAI/
description:
   toolkit:
      title: VTOAI Perl Data Provider
      author:
```

Enter “oai_dc” as the metadataPrefix parameter and click on ListIdentifiers. The result is a list of identifiers for all items in the Open Archive, as shown below:


Open Archives Initiative - Repository Explorer
explorer version - 1.45 : protocol version - 1.0/1.1/2.0 : June 2002

http://rocky.dlib.vt.edu/~classex/cgi-bin/user00/OAI-XMLFile/XMLFile/test1/oai.pl?verb=ListIdentifiers&metadataPrefix=oai_dc

List of Record Identifiers

Select a link to view more information

```
header:
  identifier : oai:test1:compend1
  datestamp : 2001-10-16T19:28:21Z
  setSpec : set2
  setSpec : set3
```

[display record in Dublin Core] [display metadata formats]

```
header:
  identifier : oai:test1:compend6
  datestamp : 2001-10-16T19:28:21Z
```

[display record in Dublin Core] [display metadata formats]

```
header:
  identifier : oai:test1:record_1
  datestamp : 2001-10-16T19:28:21Z
  setSpec : set1
```

[display record in Dublin Core] [display metadata formats]

Now we will run the full suite of automatic tests.

- Click on “home” at the bottom of the page.
- Click on “Test and Add an archive”.
- Enter the baseURL on the next page and click “Test the archive”.

This will perform a set of tests to verify that the OAI interface works and is somewhat robust. Do not click on “Submit Archive to List” when the tests are completed!

1.3. Add more data

Switch to your shell session.
Change to the “~/public_html/cgi-bin/<userid>/OAI-XMLFile/XMLFile/test1/data” directory.

```
cd ~/public_html/cgi-bin/<userid>/OAI-XMLFile/XMLFile/test1/data
```

Make a duplicate copy of one of the files there (e.g., compend1.xml) – choose any name with a “.xml” extension

```
cp compend1.xml testcopy.xml
```

Edit at least one of the fields in the file using a text editor (joe, pico, vi, vim, and emacs are installed). For example, if you use pico:

```
pico testcopy.xml
```

Go back to the browser, click home, enter the baseURL, enter “oai_dc” for metadataPrefix, and click on ListIdentifiers. You should have one more entry than before. You have just added a new item to the Open Archive !

2. Harvesting from an Open Archive

Harvester is a service provider module that implements an algorithm to get periodic updates from an Open Archive. This module is usually integrated into other modules such as IRDB. For demonstration purposes, the supplied sample code outputs records to the screen.

2.1. Installation

Change to "~/public_html/cgi-bin/<userid>" where <userid> is your userid e.g., user01

```
cd ~/public_html/cgi-bin/<userid>
```

Download the component from the ODL website if you don't already have it

```
wget http://oai.dlib.vt.edu/odl/software/harvest/Harvest-
2.0.tar.gz
```

Decompress the file

```
gzip –cd H* | tar –xf -
```

2.2. Configuration

Change to "ODL-Harvest-2.0/Harvest"

```
cd ODL-Harvest-2.0/Harvest
```

Run

```
./configure.pl test
```

Press "a" to add the archive just installed. Answer the questions asked by the configuration script as listed below:

Archive identifier: test
baseURL of the archive: as used in RE previously
Harvesting interval: 86400 (default)
Harvesting overlap: 1 (default)
Harvesting granularity: second (default)
metadataPrefix: oai_dc
set (leave empty): (default)

Lastly, press ENTER or "d" to finish configuration of the harvester.

2.3. Harvesting

Run

```
test/harvest.pl
```

This will do an initial harvest of the archive – records will be displayed on screen.

Once again, run

```
test/harvest.pl
```

Since the time interval has not elapsed, nothing will be displayed.

Force an immediate (now) harvest of all records (start) from all defined archives (all) by running:

```
      test/harvest.pl now all start
```

Now you will see the full contents of the archive

3. Connecting an Open Archive to a search engine component (IRDB)

Harvesting is useful to either import data into a system or to create services such as search engines. IRDB is a small-scale search engine that gets its data from an Open Archive and has a simple machine interface to issue queries and return results.

We will use a mySQL database, where the administrator has already created a database and assigned "all privileges" to the "yourloginname" account. Read the README file under ODL-IRDB-?.?/ dir first and make sure DBI and DBD::mySQL Perl modules are installed.

3.1. Installation

Change to "~/public_html/cgi-bin/<userid>" where <userid> is your userid e.g., user01

```
      cd ~/public_html/cgi-bin/<userid>
```

Download the component from the ODL website if you don't already have it

```
      wget http://oai.dlib.vt.edu/odl/software/irdb/IRDB-1.1.tar.gz
```

Decompress the file

```
      gzip -cd I* | tar -xf -
```

3.2. Configuration

Change to "ODL-IRDB-1.1/IRDB"

```
      cd ODL-IRDB-1.1/IRDB
```

Run

```
      ./configure.pl test
```

Answer the questions asked by the configuration script as listed below:

Database Driver : mysql
Database : yourloginname
Username : yourloginname
Password (leave blank):
Database Table: <userid>
Archive Identifier: test
Repository Name, Admin Email: leave at defaults
Archive URL: enter the baseURL for the XML-File archive

Use defaults for everything else. Use "oai_dc" for metadataPrefix.

3.3. Testing

Populate with data from the XMLFile Open Archive data provider by running:

```
      test/harvest.pl
```

Run a test query from the command-line:

```
      test/testsearch.pl “test”
```

Now run the same test query using the machine (ODL) interface by issuing:

```
      QUERY_STRING='verb=ListRecords&metadataPrefix=oai_dc&set=odlsearc
      h1/test/1/10' test/search.pl
```

3.4. Web Server Permissions

The apache web server will not run a script if the directory is group-writable. IRDB uses default permissions so you will need to disable group-writing with:

```
      chmod 755 /home/yourloginname/public_html/cgi-bin/<userid>/ODL-
      IRDB-1.1/IRDB/test
```

4. Adding a simple WWW interface to the search engine component

A search engine is not very useful without a user interface ☺
We can either parse the XML and generate HTML or use some kind of transformation or stylesheet. IRDB has a sample interface that can be installed.

4.1. Installation

Change to “~/public_html/cgi-bin/<userid>” where <userid> is your userid e.g., user01

```
      cd ~/public_html/cgi-bin/<userid>
```

Download the component from the ODL website if you don’t already have it

```
      wget
      http://oai.dlib.vt.edu/odl/software/compute_ui/compute_ui.tar.gz
```

Decompress the file

```
      gzip –cd c* | tar –xf -
```

4.2. Configuration

Edit the “search.pl” file in the UI directory. For example, if you use joe:

```
      joe UI/search.pl
```

Change the baseURL on line 29 to:

```
      http://rocky.dlib.vt.edu/~yourloginanme/cgi-bin/<userid>/ODL-
      IRDB-1.1/IRDB/test/search.pl
```

The rest of the file can be changed to change the interface appearance, but we will ignore it for now! Simply save the file and continue to the next step.

4.3. Testing the interface

Enter the URL into your web browser as:

```
      http://rocky.dlib.vt.edu/~yourloginname/cgi-
      bin/<userid>/UI/search.pl
```

Try a query such as “test”, “art”, or “war” (or any other word that appeared in the metadata). You will get a list of titles of records that match the search query. Note: The links will not work since we did not edit that part of the search.pl script

Wrap up and discussion

That’s all folks !

You have just built a simple digital library out of components to correspond to the architecture shown on the first page.

Please complete the survey about this exercise.

Thank you !