

OAI Implementation in Greenstone

Greenstone has been made OAI compliant now and it is pretty easy to accomplish also. From version 2.52, Greenstone comes with a built-in OAI data provider. This runs as a CGI program called "oaiserver", and is installed in the Greenstone *cgi-bin* directory. It can be accessed via the same URL as the Greenstone library (replacing "library" with "oaiserver"). If you are using the Windows local library server, you must install a web server (such as Apache) to run the OAI server.

1. Setting up OAI Server

Configuration of the server is done via the *oai.cfg* file in the Greenstone *etc* directory. This file specifies general information about the repository, and lists collections to be made accessible to OAI clients. By default, collections are not accessible. To enable a collection, add its name to the *oaicollection* list. Collections built with versions of Greenstone earlier than 2.52 must be rebuilt before they can be served.

Greenstone's OAI server only supports Dublin Core metadata at present. For collections that use other metadata sets, metadata mapping rules should be provided to map the existing metadata to Dublin Core. See the *oai.cfg* file for details.

Following are the steps :

1. Build a sample collection comprising around 15 to 20 documents;
2. Add DC values for all the documents in the Enrich Panel;
3. Edit the "oai.cfg" file at \Greenstone\etc and add
 - a. `baseURL "your-own-baseURL-goes-here";`
e.g: `http://greenstonesupport.iimk.ac.in/gsd1/cgi-bin/oaiserver.cgi`
 - b. `baseDocRoot "your-own-base-document-root-goes-here";`
e.g: `http://greenstonesupport.iimk.ac.in/gsd1`
 - c. the collection name in the *oaicollection*;
e.g: `oaicollection oaidemo` (Note: 'oaidemo' is the collection name)
4. Now the collection is OAI harvestable.

2. Downloading over OAI

Downloading using the Librarian Interface

1. In the Librarian Interface, switch to the **Download** panel. Select **OAI** from the list of download types on the left hand side.
2. In the **url** box, type in the following URL (this is the OAI Base URL):

e.g: <http://greenstonesupport.iimk.ac.in/gsdll/cgi-bin/oaiserver.cgi>

3. We want to download the documents as well as the metadata, so tick the **Get document** checkbox.
4. If your computer is behind a firewall or proxy server, you will need to edit the proxy settings in the Librarian Interface. Click the **<Preferences...>** button. Switch on the **Use proxy connection?** checkbox. Enter the proxy server address and port number in the **Proxy Host:** and **Proxy Port:** boxes. Click **<OK>**.
5. Now click **<Download>**. If you have set proxy information in **Preferences...**, a popup will ask for your user name and password. Once the download has started, a progress bar appears in the lower half of the panel that reports on how the downloading process is doing.
6. Downloaded files are stored in a top-level folder called **Downloaded Files** that appears on the left-hand side of the **Gather** panel. These can files can then be added to a collection.

3. Open Archives Initiative (OAI) collection

The downloaded OAI records get stored in the Downloaded Files Folder in the Gather Panel's Workspace. We need to now build a collection with the downloaded OAI records.

1. Start a new collection called **OAI Test Harvester**. Fill out the fields with appropriate information.
2. In the **Gather** panel, locate the folder '**oaiserver.cgi**' at **Downloaded Files** → **Domain Name** → **gsdl** → **cgi-bin** → **.** Drag this folder into the collection and drop it there.
3. During the copy operation, a popup window appears asking whether to add **OAIPlug** to the list of plug-ins used in the collection, because the Librarian Interface has not found an existing plug-in that can handle this file type. Press the **<Add Plugin>** button to include it.
The files for this collection consist of a set of newsclippings (in IIMK News DL) and a set of OAI records (in IIMK News DL) which contain metadata for the clippings in pdf format.

When files are copied across like this, the Librarian Interface studies each one and uses its filename extension to check whether the collection contains a corresponding plug-in. No plug-in in the list is capable of processing the OAI file records that are copied across (they have the file extension .oai), so the Librarian Interface prompts you to add the appropriate plug-in.

*Sometimes there is more than one plug-in that could process a file—for example, the .xml extension is used for many different XML formats. The popup window, therefore, offers a choice of all possible plug-ins that matched. It is normally easy to determine the correct choice. If you wish, you can ignore the prompt (click **<Don't Add Plugin>**), because plug-ins can be added later, in the **Document Plugins** section of the **Design** panel.*

4. You need to specify which document the OAI metadata values should be attached to. In the **Design** panel, select the **Document Plugins** section, then select the **OAIPlugin** and click **<Configure Plugin...>**. Locate the **document_field** option in the popup window and select **dc.Resource Identifier** from the drop-down list. Click **<OK>**
5. Now switch to the **Create** panel and **build** and **preview** the collection.

Like other collections we have built by relying on Greenstone defaults, the end result is passable but can be improved. The next steps refine the collection using the metadata harvested by OAI-PMH into the .oai files.

6. In the **Browsing Classifiers** section of the **Design** panel, delete the two **List** classifiers (**dc.Title** and **ex.Source**).
Add an **AZCompactList** classifier based on **dc.Subject and Keywords** metadata.
8. Now add an **AZCompactList** classifier based on **dc.Publisher** metadata. Give **buttonname** to **Categories**.
9. In the **Search Indexes** section of the **Design** panel, delete index on filenames and add a new one based on **dc.Subject and Keywords** metadata.
10. **Build** the collection and **preview** it.

Tweaking the presentation with format statements

11. In the **Format** panel, select **Format Features**. First replace the **VList** format statement with the following:

```
<td valign="top">
{If} {[ex.FileFormat] eq 'PDF',
[srclink][srcicon]/[srclink],
[link][icon]/[link]}</td>
<td valign="top">[ex.srclink]{Or}{[ex.thumbicon]}[ex./srclink]</td>
<td valign="top">[highlight]
{Or}{[dc.Title],[exp.Title],[ex.Title],Untitled}
[/highlight]</td>
```

This format statement customizes the appearance of vertical lists such as the search results, Keywords and categories lists.

12. Next, select **DocumentHeading** from the **Choose Feature** pull-down list and change its format statement to:

```
{Or}{[parent(Top):Title],[dc.Subject],untitled}<br>
```

The document heading appears above the DETACH and NO HIGHLIGHTING buttons when you get to a document in the collection. By default

DocumentHeading displays the document's **ex.Title** metadata. You can see them in the **Enrich** panel if you select an file in *oaiserver.cgi* and check its **ex.Source** and **ex.Title** metadata. The above format statement displays **dc.Subject** metadata instead.

13. Finally, you will have noticed that where the document itself should appear, you see only the raw OAI Record. To give a meaningful display of the record, select **DocumentText** in the **Choose Feature** pull-down list and use the following as its format statement:

```
<center><table width=_pagewidth_ border=1>
<tr><td>Title:</td><td> [dc.Title]</td></tr>
<tr><td>Keywords:</td><td> [sibling(All'<br/>'):dc.Subject]</td></tr>
<tr><td>Publisher:</td><td> [dc.Publisher]</td></tr>
<tr><td>Newspaper:<td>[dc.Source]</td></tr>
<tr><td>File:<td><a href="[dc.Identifier]">Click Here</a></td></tr>
</table></center>
```

```
{If}{_cgiargshowrecord_,
<p><b>OAI Record:</b><br/><tt>[Text]</tt></p>
<center><a href='_gwcgi_?e=_cgiarge_&a=d&d=_cgiargd_'>Hide OAI
Record</a></center>,
<center><a
href='_gwcgi_?e=_cgiarge_&a=d&d=_cgiargd_&showrecord=1\'>Show OAI
Record</a></center>
}
```

This format statement alters how the document view is presented. It includes a table display of the record that hyperlinks back to the original document available on the web. Factual information extracted from the file is also displayed.

14. Format statements are processed by the runtime system, so the collection does not need to be rebuilt for these changes to take effect. Click **<Preview Collection>** to see the changes.

To expedite building, this collection contains fewer source documents than the pre-built version supplied with the Greenstone installation. However, after these modifications, its functionality is the same.
