

Lab 5: Interoperability

5.1. Customization: macro files and stylesheets

The appearance of all pages produced by Greenstone is governed by macro files, which reside in the folder *Greenstone macros*, images, and CSS stylesheets, both of which reside in *Greenstone images*.

A macro takes the form `_macro_name_ {macro value}`. Macro names start and end with underscores (`_`), and the macro value is enclosed in curly brackets (`{}`). Macro values can be text or HTML, and can include other macros.

Macros are grouped into packages, and different packages control the appearance of different pages. For example, the **home**, **help**, **preferences**, **query**, **document** packages control the home, help, preferences, query, and document pages, respectively. Some macro files contain macros for just one package, for example, *home.dm*, *query.dm*, *document.dm*, while others contain macros for many packages. *base.dm* contains macros used globally, *style.dm* controls the common style of each page, *english.dm*, *french.dm* and other language files contain the text fragments for the entire interface, in that specific language.

The output of the library program is a page of HTML which is viewed in a web browser. CSS (Cascading Style Sheets) are often used alongside HTML pages to control the formatting, such as layout, colour, font etc. The default Greenstone stylesheet is *Greenstone images style.css*.

In this exercise, we customize the macros, images and stylesheets to change the appearance of our library.

Changing the background and header images

1. Three new images for this exercise can be found in *sample_files custom*. Copy *chalk-blue.gif*, *gsdlhead-blue.gif* and *divb-blue.gif* from the *custom* folder into the *Greenstone images* folder.
2. Open the file *Greenstone macros home.dm* in a text editor, e.g. WordPad. Find each occurrence of `gsdlhead.gif` in this file (there are two) and replace with `gsdlhead-blue.gif`. (If you are using WordPad, you can use **Edit Find** to search for the text.)

Save *home.dm* and close the file.

3. Open the file *Greenstone macros style.dm* with the same program. Locate the following part of the file (this is part of the `_cssheader_` macro):

```
<style type="text/css">
body.bgimage \{ background-image: url("_httpimg_/chalk.gif"); \}
```

Use copy and paste on the `body.bgimage` line to make it look like this:

```
<style type="text/css">
/*body.bgimage \{ background-image: url("_httpimg_/chalk.gif"); \}*/
body.bgimage \{ background-image: url("_httpimg_/chalk-blue.gif"); \}
```

`/*...*/` around a line signals a comment, and this style element will be ignored. We use this to "comment out" the original line and replace it with a modified line. This way it is easy to revert back to the original if necessary. Here we are changing the background image for the `bgimage` section of the `body` of the page to `chalk-blue.gif`.

Save *style.dm* and close the file.

4. Preview the home page in a web browser. (On Windows, restart the Greenstone library server.) The page header and background should now use the new graphics, and be blue.

The final part of this exercise looks at how we determined which images needed replacing, and which

macro files should be edited.

Changing the colour of the navigation bar, page title and page text

Now that the background image is a nice blue colour, lets format the page so that some other parts are blue too. Preview the collection after each change to make sure that it has worked properly. On Windows, macro file changes require a restart of the Greenstone library server. Stylesheet changes may require a force reload in the web browser.

5. First, we'll change the colour of the navigation bar and green divider bars. These use an image as a background, specified in the same macro as the page background.

Open *Greenstone macros style.dm* in a text editor, and find the `_cssheader_` macro that you modified previously. Change the `div.navbar` and `div.divbar` parts to use `divb-blue.gif` instead of `bg_green.png`:

```
/*div.navbar \{ background-image: url("_httpimg_/bg_green.png"); \}*/
div.navbar \{ background-image: url("_httpimg_/divb-blue.gif"); \}
/*div.divbar \{ background-image: url("_httpimg_/bg_green.png"); \}*/
div.divbar \{ background-image: url("_httpimg_/divb-blue.gif"); \}
```

6. The selected item on the navigation bar uses the same background, so change that too:

```
/*a.navlink_sel \{ background-image: url("_httpimg_/bg_green.png"); \}*/
a.navlink_sel \{ background-image: url("_httpimg_/divb-blue.gif"); \}
```

7. Next, we get rid of the background green image on the page and collection titles. Comment out the `p.bannertitle` and `p.collectiontitle` parts:

```
/*p.bannertitle \{background-image: url("_httpimg_/banner_bg.png"); \}*/
/*p.collectiontitle \{background-image: url("_httpimg_/banner_bg.png"); \}*/
```

The above style definitions were included in the macro file so that image paths could be dynamically generated. The majority of the style definitions reside in an external style file, *Greenstone images style.css*, and most style changes involve modifying that file.

8. Open *Greenstone images style.css* in a text editor. Make the following modifications. You might want to preview after each one to see the effect.

Change some of the colours:

- Find the `body` style instructions:

```
body {
background: #ffffff;
}
```

Set color to teal.

- For `a.collectiontitle`, set color to blue.
- For `p.collectiontitle`, add `color: blue;`

9. For fun, lets switch the positions of the home, help and preferences buttons and the collection name or image.

- For `div.pageinfo`, set both `float` and `text-align` to left.
- For `div.collectimage`, set `float` and `text-align` to right.

The look of your library should now be substantially different.

Adding a footer

10. Next we add a footer to each page. *Greenstone macros style.dm* defines a header and footer for each page, and macro files for the different pages define the page content. Open the file *Greenstone*

`macros style.dm` in a text editor.

11. Locate the `_footer_` macro:

```
_footer_ {
<!-- page footer (\_style:footer\) -->
_pagefooterextra__endspacer__htmlfooter_
}
```

After `_pagefooterextra_` add some text or HTML. For example `<center><small>Copyright 2006 My Awesome Digital Library</small></center>`. The resulting macro will look something like:

```
_footer_ {
<!-- page footer (\_style:footer\) -->
_pagefooterextra_ <center><small>Copyright 2006 My Awesome Digital Library</small></center>
_endspacer__htmlfooter_
}
```

The `<center>` and `<small>` HTML tags center the text, and make it a smaller size than the rest of the page.

Save `style.dm` and close the file.

- Preview the changes in a web browser. (On Windows, restart the Greenstone library server.) Each page should now have the new text at the bottom.
- Adding text into the main `_footer_` macro adds it to all pages. To add a footer just to a particular page, use `_pagefooterextra_` in the appropriate macro file. For example, lets add some more text to the footer, this time just on the home page.

Open the file `Greenstone macros home.dm` in a text editor. After the line `package home`, add the following text:

```
_pagefooterextra_ {Collections generated by Me.}
```

Save `home.dm` and close the file.

Preview the home page in a web browser. (On Windows, restart the Greenstone library server.) The home page should now display the new text, while the other pages won't.

Make your own Greenstone home page

You can make radical changes to a page by changing the macro file completely. For example, here we use a predefined alternative to the home page.

- Open the file `Greenstone etc main.cfg` in a text editor. Locate the `macrofiles` list:

```
# The list of display macro files used by this receptionist
macrofiles tip.dm style.dm base.dm query.dm help.dm pref.dm about.dm \
document.dm browse.dm status.dm authen.dm users.dm html.dm \
extlink.dm gsdl.dm extra.dm home.dm collect.dm docs.dm \
bsummary.dm gti.dm gli.dm nav_css.dm usability.dm \
...
```

Change the text `home.dm` to `yourhome.dm`. Save and close the file.

- Preview the newly structured home page in a web browser. (On Windows, restart the Greenstone library server.)
- Reverse this last change by changing `yourhome.dm` back to `home.dm` in the file `Greenstone etc main.cfg`. You may also like to reverse the other changes you have made.

Collection specific customisation

Macros can also be used to customize single collections. They should be added to a file called `extra.dm` in the

macros directory of a collection. This part of the exercise can be done using the Librarian Interface.

We use the Word and PDF collection (from exercise **A collection of Word and PDF files**) as the example for this exercise, but it can be done with any collection. Open up this collection (reports) in the Librarian Interface.

17. Go to the **Format** panel, and select **Collection Specific Macros** from the left hand list. This section allows you to edit the collection's *extra.dm* macro file.
18. First, we change the title of the **About this collection** section of the about page. Add the following text in the edit box:

```
package about

_textabout_ {
<div class="section">
<h3>Very Interesting Reports Collection.</h3>
_Global:collectionextra_
</div>
}
```

Preview the collection. (On Windows, restart the Greenstone library server.) The about page will have a new title underneath the search form.

19. Next we'll do some style customisations for this collection. Add the following text:

```
package Style

_collectionspecificstyle_ {
<style type="text/css">
/*clear the use of a background image */
body.bgimage \{ background-image: none; \}
/* set the background color to pink */
body \{ background: pink; \}
/* clear the background image for the navigation bar, and set its color to red */
div.navbar \{ background-image: none; background-color: red; \}
/* clear the background image for the divider bars, and set their color to red */
div.divbar \{ background-image: none; background-color: red; \}
</style>
}
```

Preview the collection. (On Windows, restart the Greenstone library server.) The reports collection will now have a pink background, and the navigation bar and divider bars will be red. These changes will only affect this collection.

Any macros from the general macro files can be copied into a collection's *extra.dm* file and modified. Remember to include the package declaration to make sure that the macros get applied to the correct page(s).

The style modifications made above were minor. The collection still uses the majority of the standard style file. The style declarations in the `_collectionspecificstyle_` macro get appended to the default ones. To completely change the appearance of a collection, we can use a new style sheet altogether.

20. Add the following to *extra.dm* after the last modifications:

```
_cssheader_ {
<link rel="stylesheet" href="_httpcimages_/style-blue.css" type="text/css"
title="Blue Style" charset="UTF-8">
}
```

Outside of the Librarian Interface, locate the collection folder *Greenstone collect reports*. Create an *images* folder inside this (if not already present), and copy the file *sample_files custom style-blue.css* into this folder.

Preview the collection; it should look radically different.

How to determine which images to replace (advanced)

21. In the first part of this exercise we replaced the default background (**chalk.gif**) and header

(**gsdlhead.gif**) images with new ones. To do this we needed to change the image names in the macro files. How did we know which images we were replacing and which macro files to edit? This exercise shows you how to find out.

22. To find out the names of the images to replace, go to the home page of your digital library in a browser. Right-click on the header image ("Greenstone digital library software") and select "Save picture as". A dialog will pop up and will display the image name: "gsdlhead.gif" (or "gsdlhead-blue.gif" if you are using the new header). Click Cancel to close the dialog—you don't need to save the images. Do the same for the background image by right clicking on the left hand green (or blue) swirly bar. This time choose "Save background as" to find the name: "chalk.gif" (or "new_background.gif"), then click Cancel.
23. These instructions apply to Internet Explorer. Other browsers may have other options in the right-click menu. For example, Mozilla provides "View Image" and "View Background Image" options. Using these options will put the path to the image in the browser address box, and the name can be seen from this.
24. Once you have identified the names of the images to be replaced, you need to find out where they occur in the macro files. To do this, search the macro files for the image names using the **find** program, which is run in a command prompt. Open a command prompt using **Start Programs Accessories Command Prompt** , or **Start Run** and enter `cmd` as the name of the program to run.

You can type `find/?` to see a description of the program and its arguments.

To search the macro files for "gsdlhead.gif" type

```
find "gsdlhead.gif" "C:\Program Files\Greenstone\macros\*.dm"
```

***.dm** means all files ending in **.dm**. A list of all macro files will be displayed, along with any matches. You will see that *home.dm* and *exported_home.dm* both contain **gsdlhead.gif**. *home.dm* is the one you want to edit—*exported_home.dm* is used for the home page when you export a collection to CD-ROM.

Do the same thing for "chalk.gif":

```
find "chalk.gif" "C:\Program Files\Greenstone\macros\*.dm"
```

base.dm and *style.dm* are the only files that mention this image.

Close the command prompt.

5.2. Open Archives Initiative (OAI) collection

This exercise explores service-level interoperability using the Open Archive Initiative Protocol for Metadata Harvesting (OAI-PMH). So that you can do this on a stand-alone computer, we do not actually connect to the external server that is acting as the data provider. Instead we have provided an appropriate set of files that take the form of XML records produced by the OAI-PMH protocol.

One of Greenstone's documented example collections is sourced over OAI. This exercise takes you through the steps necessary to reconstruct it. (Note: this example is a collection of images: you will not be able to build it unless ImageMagick is installed on your computer.) You may wish to take a look at the documented example collection OAI demo now to see what this exercise will build.

1. Start a new collection called **OAI Service Provider**. Fill out the fields with appropriate information.
2. In the **Gather** panel, locate the folder `sample_files oai sample_small oai`. 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 images (in `JCDLPICS srcdocs`) and a set of OAI records (in `JCDLPICS`) which contain metadata for the images.

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 configure the image plug-in. In the **Design** panel, select the **Document Plugins** section, then select the **ImagePlug** line and click **<Configure Plugin...>**. In the resulting popup window locate the **screenviewsize** option, switch it on, and type the number **300** in the box beside it to create a screen-view image of 300 pixels. Click **<OK>**.
5. Now switch to the **Create** panel and **build** and **preview** the collection.

OAIPlug will process the OAI records, and assign metadata to the images, which are processed by **ImagePlug**.

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 **AZList** classifiers (**ex.Title** and **ex.Source**).
7. Add an **AZCompactList** classifier based on **ex.Subject** metadata.
8. Now add an **AZCompactList** classifier based on **ex.Description** metadata. In its configuration panel set **mingroup** to **2**, **mincompact** to **1**, **maxcompact** to **10** and **buttonname** to **Captions**.

Setting **mingroup** to 2 will mean that two or more documents with the same description will be

grouped into a bookshelf; the default **mingroup** of 1 means that every document will get a bookshelf. **mincompact** and **maxcompact** control how many documents are grouped into each section of the horizontal A-Z list. In this case, each group can have as few as one document, and no more than ten.

9. In the **Search Indexes** section of the **Design** panel, delete all indexes and add a new one based on **ex.Description** 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 (which can be copied from the file `vlist_tweak.txt` in the `sample_files oai format_tweaks` folder).

```
<td>
  {If}{[numleafdocs],[link][icon][link],[link][thumbicon][link]}
</td>
<td valign=middle>
  {If}{[numleafdocs],[Title],<i>[Description]</i>}
</td>
```

This format statement customizes the appearance of vertical lists such as the search results and captions lists to show a thumbnail icon followed by Description metadata. Greenstone's default is to use extracted metadata, so `[Description]` is the same as `[ex.Description]`.

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

```
<h3>[Subject]</h3>
```

*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. In this particular set of OAI exported records, titles are filenames of JPEG images, and the filenames are particularly uninformative (for example, `01dla14`). You can see them in the **Enrich** panel if you select an image in `oai JCDLPICS srcdocs` and check its **ex.Source** and **ex.Title** metadata. The above format statement displays **ex.Subject** metadata instead.*

13. Finally, you will have noticed that where the document itself should appear, you see only "This document has no text.". To rectify this, select **DocumentText** in the **Choose Feature** pull-down list and use the following as its format statement (this text is in `doctxt_tweak.txt` in the `format_tweaks` folder mentioned earlier):

```
<center><table width=_pagewidth_ border=1>
<tr><td colspan=2 align=center>
  <a href=[OrigURL]>[screenicon]</a></td></tr>
<tr><td>Caption:</td><td> <i>[Description]</i> <br>
  (<a href=[OrigURL]>original [ImageWidth]x[ImageHeight] [ImageType] available</a>)
</td></tr>
<tr><td>Subject:</td><td> [Subject]</td></tr>
<tr><td>Publisher:</td><td> [Publisher]</td></tr>
<tr><td>Rights:<td> [Rights]</td></tr>
</table></center>
```

This format statement alters how the document view is presented. It includes a screen-sized version of the image that hyperlinks back to the original larger version available on the web. Factual information extracted from the image, such as width, height and type, 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.

5.3. Downloading over OAI

The previous exercise did not obtain the data from an external OAI-PMH server. This missing step is accomplished either by running a command-line program or by using the **Download** panel in the Librarian Interface. This exercise shows you how to do this using both methods.

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:

<http://rocky.dlib.vt.edu/~jcdlpix/cgi-bin/OAI/jcdlpix.pl>
3. We want to download the documents as well as the metadata, so tick the **get_doc** 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 files can then be added to a collection.

Downloading using the command line

For command line downloading to work, your computer must have a direct connection to the Internet—being behind a firewall may interfere with the ability to download the information. You will need to use the Librarian Interface for downloading if you are behind a firewall.

7. Close the Librarian Interface.

We will work with the OAI collection used in exercise **Open Archives Initiative (OAI) collection**. You may have noticed that its internal name is **oiservi**.

8. In a text editor (e.g. WordPad), open the collection's configuration file, which is in *Greenstone collect oiservi etc collect.cfg*. Add the following line (all on one line):

```
acquire OAI -src http://rocky.dlib.vt.edu/~jcdlpix/cgi-bin/OAI/jcdlpix.pl -getdoc
```

Although the position of this line is not critical, we recommend that you place it near the beginning of the file, after the public and creator lines but before the index line. Save the file and quit the editor.

9. Delete the contents of the collection's *import* folder. This contains the canned version of the collection files, put there during the previous exercise. Now we want to witness the data arriving anew from the external OAI server.
10. Open a DOS window to access the command-line prompt. This facility should be located somewhere within your **Start Programs** menu, but details vary between different Windows systems. If you cannot locate it, select **Start Run** and enter *cmd* in the popup window that appears.
11. In the DOS window, move to the home directory where you installed Greenstone. This is accomplished by something like:

```
cd C:\Program Files\Greenstone
```

12. Type:

```
setup.bat
```

to set up the ability to run Greenstone command-line programs.

13. Change directory into the folder containing the OAI Services Provider collection you built in the last exercise.

```
cd collect\oaiservi
```

Even though the collection name used capital letters the directory generated by the Librarian Interface is all lowercase.

14. Run:

```
perl -S importfrom.pl oaiservi
```

Greenstone will immediately set to work and generate a stream of diagnostic output. The importfrom.pl program connects to the OAI data provider specified in collection configuration file (it does this for each "acquire" line in the file) and exports all the records on that site.

15. The downloaded files are saved in the collection's import folder. Once the command is finished, everything is in place and the collection is ready to be built. Confirm you have successfully acquired the OAI records by rebuilding the collection.

5.4. Use METS as Greenstone's Internal Representation

1. In the Greenstone Librarian Interface, open up one of your existing collections, for example the **Small Tudor** collection.

*To be able to substitute **METSPlug** for **GAPug** you need to be in **Expert** mode.*

2. Click **File** **Preferences...** **Mode** and change to **Expert** mode.
3. Switch to the **Design** panel and select **Document Plugins**. Remove **GAPug** from the list of plug-ins and add **METSPlug**, with the default configuration options.
4. Now change to the **Create** panel, locate the options for the import process and set **saveas** to **METS**. Import options are not available unless you are in **Expert** mode.
5. **Rebuild** the collection.
6. In your Windows file browser, locate the *archives* folder for the collection you are working with (in *Greenstone collect <collname> archives*). For each document in the collection, Greenstone has generated two files: *docmets.xml*, the core METS description, and *doctxt.xml*, a supporting file. (Note: unless you are connected to the Internet you will be unable to view *doctxt.xml* in your web browser, because it refers to a remote resource.) Depending on the source documents there may be additional files, such as the images used within a web page. One of METS' many features is the ability to reference information in external XML files. Greenstone uses this to tie the content of the document, which is stored in the external XML file *doctxt.xml*, to its hierarchical structure, which is described in the core METS file *docmets.xml*.

5.5. Moving a collection from DSpace to Greenstone

1. First, change to **Library Systems Specialist** (or **Expert**) mode (using **File Preferences...**), because you will need to change the order of plug-ins in the **Design** panel.
2. Start a **new collection** called **StoneD** and fill out its fields appropriately. Leave the metadata set at Dublin Core, the default.
3. Switch to the **Design** panel and select the **Document Plugins** section on the left-hand side. **Remove ZIPPlug, TEXTPlug, HTMLPlug, EMAILPlug, PSPlug, ImagePlug, ISISPlug and NULPlug.** Strictly speaking we do not need to remove these, however it reduces clutter.
4. Now add **DSpacePlug**. Leave the plugin options at their defaults and press **<OK>**.
5. Using the up arrow, **move** the position of **DSpacePlug** to the top of the list (above **GAPlug**).
6. In the **Gather** panel, locate the folder *sample_files dspace* . It contains five example items exported from a DSpace institutional repository. Copy them into your collection by dragging them over to the right-hand side of the panel.
7. **Build** the collection and **preview** it to see the basic defaults exhibited by a DSpace collection.

If you browse by Titles, you will find 7 documents listed, though only 5 items were exported from DSpace. Two of the original items had alternative forms in their directory folder. The DSpace plug-in options control what happens in such situations: the default is to treat them as separate Greenstone documents.

*Below we use a plug-in option (**first_inorder_ext**) to fuse the alternative forms together. This option has the effect of treating documents with the same filename but different extensions as a single entity within a collection. One of the files is viewed as the primary document—it is indexed, and metadata is extracted from it if possible—while the others are handled as "associated files."*

*The **first_inorder_ext** option takes as its argument a list of file extensions (separated by commas): the first one in the list that matches becomes the primary document.*

8. Select **DSpacePlug** and click **<Configure Plugin...>**. Switch on its configuration option **first_inorder_ext**. Set its value to "pdf,doc,rtf" in the popup window that appears and press **<OK>**.
9. **Build** and **preview** the collection.

There are now only 5 documents, because only one version of each document has been included—the primary version.

Adding indexing and browsing capabilities to match DSpace's

The DSpace exported files contain Dublin Core metadata for title and author (amongst other things).

10. In the **Design** panel, select **Search Indexes**. Delete the **ex.Title** and **ex.Source** indexes, and add one for **dc.Title** called "titles" and another for **dc.Contributor** called "authors".
11. Staying within the **Design** panel, select **Browsing Classifiers** and **delete** both **AZList** classifiers (**ex.Title** and **ex.Source**). Add an **AZList** classifier for **dc.Title** and an **AZCompactList** classifier for **dc.Contributor**.
12. Now select the **Format Features** section of the **Format** panel, and select the **VList** format statement in the list of assigned format statements. Add the following text before the final `</td>`:

```
{If}{[ex.equivlink],<br>Also available as:[ex.equivlink]}
```

13. Also, let's add a format statement for the classifier based on **dc.Contributor** metadata. In the **Choose Feature** menu (under **Format Features** on the **Format** panel), select the item that says:

CL2: AZCompactList -metadata dc.Contributor

14. Leave **VList** as the **Affected Component** and click **<Add Format>**. Edit the text in the **HTML Format String** box. Replace

```
{Or}{[dls.Title],[dc.Title],[ex.Title],Untitled}
```

with

```
{If}{[numleafdocs],[ex.Title],[dc.Title]}
```

This will display the number of documents for each bookshelf in the *Contributors* classifier.

15. **Build** collection once again and **preview** it.

There are still only 5 documents, but against some of the entries appears the line "Also available as:" followed by icons that link to the alternative representations.

5.6. Moving a collection from Greenstone to DSpace

*In this exercise you export a Greenstone collection in a form suitable for DSpace. It is possible to do this from the Librarian Interface's **File** menu, which contains an item called **Export...**, that allows you to export collections in different forms. However, to gain a deeper understanding of Greenstone, we perform the work by invoking a program from the Windows command-line prompt. This requires some technical skill; if you are not used to working in the command-line environment we recommend that you skip this exercise.*

Using Greenstone from the command line

1. Open a DOS window to access the command-line prompt. This facility should be located somewhere within your **Start Programs** menu, but details vary between different Windows systems. If you cannot locate it, select **Start Run** and enter `cmd` in the popup window that appears.
2. In the DOS window, move to the home directory where you installed Greenstone. This is accomplished by something like:

```
cd C:\Program Files\Greenstone
```

3. Type:

```
setup
```

to set up the ability to run Greenstone command-line programs.

4. Change directory into the folder containing the Stoned collection you built in the last exercise.

```
cd collect\stoned
```

5. Run the following command to export the collection using the DSpace import/export format:

```
perl -S export.pl -saveas DSpace -removeold stoned
```

Exporting in Greenstone is an additive process. If you ran the `export.pl` command once again, the new files exported would be added—with different folder names—to those already in the export folder. For the kind of explorations we are conducting we might re-run the command several times. The `-removeold` option deletes files that have previously been exported.

6. This command has created a new subfolder, `collect stoned export`. Use the file browser to explore it. In it are the files needed to ingest this set of documents into DSpace.

You could equally well run the `export.pl` command on a different Greenstone collection and transfer the output to a DSpace installation by using DSpace's batch-import facility.

Copyright © 2005 2006 2007 by the [New Zealand Digital Library Project](#) at [the University of Waikato](#), New Zealand

Permission is granted to copy, distribute and/or modify this document under the terms of the [GNU Free Documentation License](#), Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled [“GNU Free Documentation License.”](#)