United Nations Educational, Scientific and Cultural Organization

National Science Foundation (NSF), Sri Lanka - UNESCO

Workshop
on
Greenstone Digital Library Software

COURSE MATERIAL

at

University of Colombo, Sri Lanka

March 21-24, 2007

National Science Foundation
47/5, Maitland place
Colombo -07

Message from the Director

Information & knowledge are the key resources for the development of a nation. The digital revolution has introduced many changes & challenges in the information field resulting in the evolution of libraries into digital libraries. According to Witten & Bainbridge (2003) Digital collections have enormous potential for changing the way that information is used & for developing new ways of preserving collecting, organizing, propagating & accessing knowledge.

The digitization initiative has picked up momentum across the world although a wide disparity is observed in the development among different countries. The domain of digital library use & application is very wide spanning academia, government, industry & other sectors. It is extremely essential that librarians & information professionals today are well equipped with necessary knowledge & skills to maintain their institutional digital collections and to become an integral part of the global information society.

However digitization projects involve high investments. The software used are often very costly and unaffordable by developing nations. The *Green Stone Digital Library* Open source Software from the University of Waikato, New Zealand offers a cost effective solution in this context. The Greenstone provides a flexible digital library framework that offers plenty of freedom & advantage to work within multilingual & multimedia information. It offers an ideal cost effective software solution to Sri Lanka.

I am extremely happy that the National Science Foundation is organizing this workshop to empower library & information community in Sri Lanka towards building digital collections. I wish to extend my sincere thanks to the UNESCO for their sponsorship and I am grateful to Dr. Sreekumar & his associates for their valuable contribution in conducting the workshop.

13/03/07

Dr. M. C. N. Jayasuriya
Director, National Science Foundation

United Nations Educational, Scientific and Cultural Organization

National Science Foundation (NSF) - UNESCO

Training Workshop on

GREENSTONE DIGITAL LIBRARY SOFTWARE

at
University of Colombo

March 21-24, 2007

Session Plan

Day 1: March 21, 2007

Time	Topic	Faculty
9.00-9.30	Registration	
9.30-10.00	Inauguration	
10.00-10.15	Tea/Coffee Break	
10.15-11.15	Greenstone Digital Library Software: Usage, Interoperability and the Future	MGS
11.15-12.00	Workshop 1: Installing, Browsing & Building	MGS
12.00-1.00	Demo + Hands on Lab 1: Installing, Browsing & Building	MGS/AP/RB
1.00-1.45	Lunch Break	
1.45-3.15	Demo + Hands on Lab 1 Continued	MGS/AP/RB
3.15-3.30	Tea/Coffee Break	
3.30-4.15	Ice Breaking	MGS/AP/RB

Day 2: March 22, 2007

Time	Topic	Faculty
8.30-9.30	Digital Objects Creation and Management: Tools and Techniques	AP
9.30-10.30	Workshop 2: Adding Metadata and Using it	MGS
10.30-10.45	Tea/Coffee break	
10.45-1.00	Demo + Hands on Lab 2: Adding Metadata and Using it	MGS/AP/RB
1.00-1.45	Lunch Break	

1.45-2.45	Workshop 3: Advanced Collection Configuration	MGS
2.45-3.00	Tea/Coffee break	
3.00-4.15	Demo + Hands on Lab3: Advanced Collection Configuration	MGS/AP/RB

Day 3: March 23, 2007

Time	Topic	Faculty
8.30-9.30	Workshop 4: Multimedia and Scanned Images	MGS
9.30-11.00	Demo + Hands on Lab3 Continued	MGS/AP/RB
11.00-11.15	Tea/Coffee break	
11.15-1.00	Hands on Lab 4: Multimedia and Scanned Images	MGS/AP/RB
1.00-1.45	Lunch Break	
1.45-2.45	Workshop 5: CDS/ISIS to Greenstone	MGS
2.45-3.00	Tea/Coffee Break	
3.00-4.15	Hands on Lab 5	MGS/AP/RB

Day 4: March 24, 2007

Time	Topic	Faculty
8.30-9.30	IT Infrastructure and Management Strategies	AP
9.30-10.30	Workshop 6: Metadata and Interoperability	MGS
10.30-10.45	Tea/Coffee break	
10.45-1.00	Hands on Lab 6: Metadata and Interoperability	MGS/AP/RB
1.00-1.45	Lunch Break	
1.45-3.00	Information Management Strategies in the New Information Environment	MGS
3.00-3.15	Tea/Coffee Break	
3.15-4.15	Feedback & Valedictory Session	

Resource Persons

MGS : Dr. M.G. Sreekumar
AP : Mr. Ashok Pathak
RB : Mr. R. Biju

TABLE OF CONTENTS

LAB 1:

Greenstone: Installing, browsing, building

1.1. Installing Greenstone

Installing Greenstone on a Windows system

You need to install three associated softwares in addition to Greenstone for getting the best out of the software. They are : 1. Java Run-time Environment (JRE), 2. ImageMagick, and 3. Ghostscript.

It is advised that you install JRE, followed by ImageMagick and Ghostscript and finally, Greenstone.

I. Installing Java Run-time Environment (JRE) Version 1.5.0-05

1. Locate The file **jre-1_5_0_05-windows-i586-p.exe** on the CD
2. Start installation by double clicking on this **"setup" icon**
3. Click on **<Yes>** to accept license agreement
4. Click on **<next>** to select default setup type, which is **'Typical'**
5. JRE gets installed to default directory, **c:\program files\java**
6. Test The installation by executing the following command in MS_DOS prompt

   C:\Java –version

The above command generates three lines of output indicating JRE and its version.

II. Installing Imagemagick (Version - 6.2.3-Q16) Software on Windows

1. Locate The file **ImageMagick-6.2.3-2-Q16-windows-dll** on the CD
2. Start installation by double clicking on this **"Next" icon** in the Install Wizard
3. Click on **<Yes>** to accept license agreement
4. Imagemagick gets installed to default directory, **c:\program files\ ImageMagick-6.2.3-Q16**
5. You have now installed ImageMagick. To test the installation select Command Prompt from the Windows Start menu. Within the window type:

   convert logo: logo.miff
   imdisplay logo.miff

and the ImageMagick logo should be displayed in a window.

If you selected to create a desktop icon, an icon labeled **ImageMagick Display** will appear on your desktop. Double clicking brings up ImageMagick's image display program.

For your convenience, the full content of the ImageMagick web site has been installed on your computer. This is available from the Windows start menu via "ImageMagick 6.2.0" -> "ImageMagick Web Pages".

III. Installing Greenstone Version 2.72 on Windows

Insert the programme CD to the Drive and select "**resources**" button in the popping up page, and click on the "**Install Greenstone**" link. You will now get a new folder opened up.

Do the following steps now:

A. Click on "gsdl-2.72-win32.exe". The Install Shield Wizard will begin the installation of GSDL software [Note : A graphical tool for collection building called the Greenstone Librarian Interface (GLI) which requires version 1.5.0 of the Java Runtime Environment (JRE 1.5.0) is already included in the software]
Click <next>

B. Accept all the term of license agreement by clicking on **<Yes>** button.

C. Click on **<next>** to install GSDL in the default folder, which is **C:\program files\greenstone**

D. Choose the type **Local Library**. By default, **Local Library** is highlighted.
Click **<next>**

E. Set the Admin Password as **"admin"** (you can later change it).
Click **<next>**
{Installation wizard starts copying the required files from the CD}

F. Click on **Finish** button to finish GSDL installation.

To check whether your installation is proper, Click on
Start → Programs→Greenstone Digital Library→Greenstone Digital Library

Click on **Enter Library** in the 'Dialog Box' and Your Browser should display The GSDL Homepage

Participants are encouraged to read the following GSDL Guides from the Resources Section of the Programme CD:

- **Greenstone Digital Library Installer's Guide**
- **Greenstone Digital Library User's Guide**
- **Greenstone Digital Library Developer's Guide**

- **Greenstone Digital Library: From paper to collection**
- **MGPP: A search engine for XML documents User Guide**
- **Teaching Material**

N.B: CD-ROMs with Greenstone version 2.62 or earlier also include the ***Greenstone Language Pack****, which gives reader's interfaces in many languages (currently about 40). This has its own installer which you have to invoke separately, after you have installed Greenstone. CD-ROMs with version 2.70 or later now come with reader's interfaces in all available languages. Textual images have been removed from the interface; they are now done using CSS (Cascading Style Sheets). The Greenstone Language Pack is no longer needed. Instead, these CD-ROMs come with the* ***Classic Interface Pack****, which contains the old text images for use with a backwards compatibility macro file.*

1. You can also get Greenstone and its associated software from http://www.greenstone.org as well as at http://greenstonesupport.iimk.ac.in.

   Most people download the Windows distribution from http://www.greenstone.org, which contains the latest version of Greenstone. There are several optional modules that must be downloaded separately (to avoid a single massive download): **documented example collections**, the **Export to CD-ROM** package (Greenstone 2.70 and earlier), the **Language Pack** (Greenstone 2.62 and earlier) and **Classic Interface Pack** (Greenstone 2.63 and later). There is also the set of **sample files** used in these exercises. (To reduce the download size the documented example collections are distributed in unbuilt form and need to be built.)

   You need **Java** to run Greenstone. You might already have it; otherwise download it from http://java.sun.com. To work with image collections, you need **ImageMagick** (from http://www.imagemagick.org).

If Greenstone has been installed on your computer before, you should completely remove the old version before installing a new one. (However, you need not remove any pre-packaged collections that you may have installed.) To do this, see **Updating a Greenstone installation**. Here is what you need to do to install Greenstone. Older versions of the installer follow much the same sequence but use slightly different wording.

- Select the language for this installation. We choose **English**
- Welcome to the InstallShield Wizard for the Greenstone Digital Library Software. Click **<Next>**
- License Agreement. Accept the agreement and then click **<Next>**
- Choose location to install Greenstone. Leave at the default and click **<Next>**
- Setup Type. Leave at the default (Local Library) and click **<Next>**
- (For older installers you must now select collections. Leave at the default, Documented Example Collections, and click **<Next>**)
- Set admin password. Choose a suitable password and click **<Next>** (If your computer will not be serving collections online, the password doesn't matter)
- Click **<Install>** to complete the installation
- Files are copied across

- Installation is complete. If you are installing from a CD-ROM, the installer will offer to install ImageMagick (see below), and Java, if necessary.

To invoke the Greenstone Reader's interface, go to the *Greenstone Digital Library Software* item under *Programs* on the Windows *Start* menu and select *Greenstone Digital Library*. To invoke the Librarian interface, go to the same item and select *Greenstone Librarian Interface*.

IV. Building Collection using the GLI tool

The Greenstone Librarian Interface (GLI) is an easy-to-use front-end to Greenstone's collection-building functionality. It provides a graphical, point-and-click interface that allows you to gather files for your digital library collection, assign metadata to them, and then design, customize and build your collection. The Librarian Interface comes as standard in all. It is installed in a subdirectory of your Greenstone installation called "\gli", and requires a recent version of Java (JRE – Java Run-Time Environment) to run.

Accessing the Greenstone Librarian Interface (GLI)

Start→ Programs→GreenstoneDigital Library→Greenstone Librarian Interface

1. Wait for a while – it takes a few seconds to get the module ready.
2. From **GLI**, select **File→New**

   A window will pop up. Give appropriate values

   Collection title

   Description of Content

   Leave the settings for ***Base this collection on:*** at its default *New collection*

   And click ***<OK>***

3. Another window will pop up, from which you select metadata set to use.
   Select ***Dublin Core*** And click ***<OK>***.

4. You need to now **gather file/s** that will constitute the **collection**. The test files are available at C:\.

5. **Drag and Drop the required file/s OR folder/s** from the **Workspace** to the **Collection Area.**

6. You can see the file contents by double clicking on the file in the **Collection Area**.

7. Now go to Create Panel by clicking the **Create Tab**.

8. To start building the collection, simply click **<Build Collection>** at the bottom of the panel.

9. Once the collection has built successfully, a window pops up. To confirm this, Click **<OK>**

10. Click on **Preview Collection** button to look at the end result.

Features of the GLI (The 'Gather', 'Enrich', 'Design' , 'Create' and Format Panels)

The '**Gather**' Panel facilitates putting the relevant files from the 'workspace' to the 'collection building' area. The '**Enrich**' Panel explains how metadata is created, edited, assigned and retrieved, and how to use external metadata sources. Help for this is provided in the GLI Interface. The '**Design**' Panel facilitates customising your interface, once your files are marked up with metadata. Using the Gather Panel, you can specify the fields that are searchable, allow browsing through the document, facilitate the languages that are supported, and provide the buttons that are to appear on the page. Help for this is provided in the GLI Interface. The '**Create**' Panel facilitates creation of your collection.

1.2. Updating a Greenstone installation

These tutorial exercises assume that you are using Greenstone 2.60 or above.

Before updating to a new version of Greenstone, ensure that the computer is not running the Greenstone Librarian Interface or the Greenstone local library server. Normally, quitting your web browser, or quitting the Librarian Interface, also quits the server.

Removing Greenstone from a Windows system

Completely remove the existing version before you install a new version of Greenstone.

1. Ensure that you are not running Greenstone.

2. Remove the old version by going to the Windows Control Panel (from the *Settings* item on the *Start* menu). Click **Add or Remove Programs**, select **Greenstone Digital Library Software**, and **Remove** it. (To do this you may need Windows "Administrator" privileges.)

3. At the end of this procedure you will be asked whether you would like all your Greenstone collections to be removed: you should probably say *No* if you wish to preserve your work.

Occasionally, problems are encountered if older Greenstone installations are not fully removed. To clean up your system, move your Greenstone collect folder, which contains all your collections, to the desktop. Then check for the folder C:\Program Files\gsdl or C:\Program Files\Greenstone, which is where Greenstone is usually installed, and remove it completely if it exists.

Reinstalling Greenstone on a Windows system

4. The reinstallation procedure is exactly the same as the original installation procedure, described in **Installing Greenstone**. If you already have ImageMagick, you do not need to install it again.

There have been some superficial changes to the installation procedure in moving to Greenstone Version 2.60, because it uses a different installer program.

There is another important difference that you should be aware of: Versions 2.60 and above are installed in the folder Program Files\Greenstone, whereas prior versions were placed in the folder Program Files\gsdl (these are both default locations that you could have changed during installation.) When upgrading to Version 2.60, if you want to save existing collections you must explicitly move the contents of your collect folder from the old place to the new one. Future Greenstone versions will be installed in the new place, Program Files\Greenstone, so this problem will not happen again.

Amalgamating different Greenstone collections

5. If you have previously installed the Greenstone Digital Library software in a non-standard place, you should amalgamate your collections by moving them from the *collect* folder in the old place into the folder *Program Files\Greenstone\collect*.

6. If you have installed collections from pre-packaged Greenstone CD-ROMs, they reside in a different place: *C:\GSDL\collect*. To amalgamate these with your main Greenstone installation, move them into the folder *Program Files\Greenstone\collect*. The mini version of Greenstone that is associated with the pre-packaged collections is no longer necessary. To uninstall it, select *Uninstall* on the Greenstone menu of the Windows *Start* menu.

Installing the Greenstone language pack (2.62 and earlier)

*If you go to the Preferences page of any Greenstone collection, and look at the **Interface language** menu, you will probably find that only English, Spanish, French and Russian interfaces are installed.*

7. Locate the Greenstone Language Pack (glp-x.xx.exe/glp-x.xx-linux.bin/gli-x.xx-macOSx.command). This may be on the CD-ROM from which you installed Greenstone, or you may have to download it from http://www.greenstone.org.

8. Run the executable file (double click it on Windows); this will start the installer. Accept all the defaults

9. Restart the Greenstone Digital Library and look at the interface language menu again. Now you should see about 40 different languages.

Enabling other languages (2.63 and later)

If you have downloaded Greenstone from the web, then all the languages will be enabled by default. However, if you have installed Greenstone from a UNESCO CD-ROM, then only English, French, Spanish and Russian will be enabled.

10. To enable a new language, edit the file *greenstone →etc →main.cfg*. Look for the appropriate "Language" line, and uncomment it (i.e. remove the # from the start). Check that the required encoding is also enabled.

For example, suppose that we want to enable Turkish. The "Language" line for Turkish looks like:

```
#Language shortname=tr longname=Turkish default_encoding=windows-1254
```

To enable it, we remove the #, i.e. make it look like:

```
Language shortname=tr longname=Turkish default_encoding=windows-1254
```

The default encoding for Turkish is windows-1254. So we look for the windows-1254 Encoding line:

```
Encoding shortname=windows-1254 "longname=Turkish (Windows-1254)"
map=win1254.ump
```

This is already enabled (no # at the start) so we don't need to do anything else.

Installing the Classic Interface Pack (2.63 and later)

Greenstone now comes with all languages enabled. The generated HTML uses text + CSS rather than images for navigation bar, home, help, preferences buttons etc. The classic interface pack is not needed if you want to use Greenstone in another language. It is only needed if you want to revert back to the old style HTML with text images. This may be useful if you have customized your Greenstone, or if you require compatibility with Netscape 4.

11. Locate the Classic Interface Pack (gcip-x.xx.zip). This may be on the CD-ROM from which you installed Greenstone, or you may have to download it from http://www.greenstone.org.

12. The classic interface pack is a zip file containing the old text images, such as classifier buttons. Unzip the zip file into the images directory of your Greenstone installation.

13. Enable the use of the old-style macros by editing *greenstone* → *etc* → *main.cfg*: replace "nav_css.dm" with "nav_ns4.dm" in the "macrofiles" list.

14. Restart the Greenstone Digital Library. It should now be using the old text images.

1.3. Building a small collection of HTML files

You will need some HTML files, such as those in the hobbits folder in sample_files.

Running the Greenstone Librarian Interface

1. Start the Greenstone Librarian Interface:

   Start → All Programs → Greenstone Digital Library Software v2.71 → Greenstone Librarian Interface

After a short pause a startup screen appears, and then after a slightly longer pause the main Greenstone Librarian Interface appears. (A command prompt is also opened in the background.)

Starting a new collection

2. Start a new collection within the Librarian Interface:

   File → New...

3. You will create a collection based on a few HTML web pages that describe some Hobbits in *Lord of the Rings*.

   A window pops up. Fill it out with appropriate values—for example,

   Collection title: About Hobbits
   Description of content: A collection about hobbits.

   Leave the setting for **Base this collection on:** at its default: **-- New Collection --**, and click **<OK>**.

4. Next you must gather together the files that will constitute the collection. A suitable set has been prepared ahead of time in *sample_files → hobbits*. Using the left-hand side of the Librarian Interface's **Gather** panel, interactively navigate to the *sample_files* folder.

Adding documents to the collection

5. Now drag the *hobbits* folder from the left-hand side and drop it on the right. The progress bar at the bottom shows some activity. Gradually, duplicates of all the files will appear in the collection panel.

   You can inspect the files that have been copied by double-clicking on the folder in the right-hand side.

6. Since this is our first collection, we won't complicate matters by manually assigning metadata or altering the collection's design. Instead we rely on default behaviour. So pass directly to the **Create** panel by clicking its tab.

Building the collection

7. To start building the collection, click the **<Build Collection>** button.

8. Once the collection has built successfully, a window pops up to confirm this. Click **<OK>**.

9. Click the **<Preview Collection>** button to look at the end result. This loads the relevant page into your web browser (starting it up if necessary). Look around the collection and learn about Hobbits!

Viewing the extracted metadata

10. Back in the Librarian Interface, click the **Enrich** tab to view the metadata associated with the documents in the collection.

11. Presently there is no manually assigned metadata, but the act of building the collection has extracted metadata from the documents. Double click the *hobbits* folder to expand its content. Then single-click *bilbo.html* to display all its metadata in the right-hand side of the panel. The initial fields, starting "dc.", are empty. These are Dublin Core metadata fields for manually entered data.

12. Use the scroll bar on the extreme right to view the bottom part of the list. There you will see fields starting "ex." that express the extracted metadata: for example **ex.Title**, based on the text within the HTML Title tags, and **ex.Language**, the document's language (represented using the ISO standard 2-letter mnemonic) which Greenstone determines by analyzing the document's text.

13. Close the collection by clicking **File → Close**. This automatically saves the collection to disk.

Setting up a shortcut in the Librarian interface

14. To set up a shortcut to the source files, in the **Gather** panel navigate to the folder in your local file space that contains the files you want to use—in our case, the *sample_files* folder. Select this folder and then right-click it, and choose **Create Shortcut** from the menu. In the **Name** field, enter the name you want the shortcut to have, or accept the default *sample_files*. Click **<OK>**. Close all the folders in the file tree in the left-hand pane, and you will see the shortcut to your source files.

1.4. A collection of Word and PDF files—Part A

You will need some source files like those in the sample_files → Word_and_PDF folder.

1. Start a new collection called **reports** (**File → New...**) and base it on -- **New Collection** --.

2. Copy all the files from *sample_files → Word_and_PDF → Documents* into the collection. You can select multiple files by clicking on the first one and shift-clicking on the last one, and drag them all across together. (This is the normal technique of multiple selection.)

3. Switch to the **Create** panel, and **build** and **preview** the collection.

Viewing the extracted metadata

4. Again, this collection contains no manually assigned metadata. All the information that appears—title and filename—is extracted automatically from the documents themselves. Because of this the quality of some of the title metadata is suspect.

5. Back in the Librarian Interface, click the **Enrich** tab to view the automatically extracted metadata. You will need to scroll down to see the extracted metadata, which begins with "ex.".

6. Check whether the **ex.Title** metadata is correct for some of the documents by opening them. You can open a document from the Librarian Interface by double clicking on it.

7. The extracted Title metadata for some documents is incorrect. For example, the Titles for *pdf01.pdf* and *word03.doc* (the same document in different formats) have missed out the second line. The Title for *pdf03.pdf* has the wrong text altogether. The PostScript documents (*cluster.ps* and *langmodl.ps* do not have extracted titles: what appears in the *Titles* list is just the first few characters of the document).

In exercise 2.1 we correct some of this incorrect metadata by manually adding Dublin Core Title metadata.

1.5. A large collection of HTML files—Tudor

1. Invoke the Greenstone Librarian Interface (from the Windows *Start* menu) and start a new collection called **tudor** (use the **File** menu), based on the default **-- New Collection --**.

2. In the **Gather** panel, open the *tudor* folder in *sample_files*.

3. Drag *englishhistory.net* from the left-hand side to the right to include it in your **tudor** collection.

4. Switch to the **Create** panel and click **<Build Collection>**.

5. When building has finished, **preview** the collection.

Extracting more metadata from the HTML

6. The browsing facilities in this collection (*Titles* and *Filenames*) are based entirely on extracted metadata. Return to the **Enrich** panel in the Librarian Interface and examine the metadata that has been extracted for some of the files.

7. Many HTML documents contain metadata in `<meta>` tags in the `<head>` of the page. Open up the *englishhistory.net* → *tudor* → *monarchs* → *boleyn.html* file by navigating to it in the tree on the left hand side, and double clicking it. This will open it in a web browser. View the HTML source of the page (**View** → **Source** in Internet Explorer, **View** → **Page Source** in Mozilla). You will notice that this page has *page_topic, content* and *author* metadata.

8. By default, **HTMLPlug** only looks for Title metadata. Configure the plugin so that it looks for the other metadata too. Switch to the **Design** panel and select the **Document Plugins** section. Select the **plugin HTMLPlug** line and click **<Configure Plugin...>**. A popup window appears. Switch on the **metadata_fields** option, and set the value to

```
Title,Author,Page_topic,Content
```

Make sure that you have copied this exactly, with no spaces. Click **<OK>**.

9. Switch to the **Create** panel and **rebuild** the collection. Go back to the **Enrich** panel and look at the extracted metadata for some of the HTML files in *englishhistory.net* → *tudor* → *monarchs*. The new metadata should now be visible.

Blocking the stray images

You've probably noticed that the collection contains a few stray image files, as well as the HTML documents. This is a mistake. The issue is that many of the HTML documents include images, and although Greenstone attempts to determine which images belong to HTML pages and only considers other images for inclusion in the collection, in this case it hasn't been completely successful. (This is because the web site from which these files were downloaded occasionally departs from the usual convention of hierarchical structuring.)

10. Switch back to the **Document Plugins** section of the **Design** panel. Beside **plugin HTMLPlug** you will see **-smart_block**. This is the option that attempts to identify images in the HTML pages and block them from inclusion—in this case, it's not smart enough! Configure **plugin HTMLPlug** again, scroll down the page to locate the **smart_block** option, and switch it off.

11. **Rebuild** and **preview** the collection. The collection is exactly as before except that these stray images are suppressed. What is happening is that plug-ins operate as a pipeline: files are passed to each one in turn until one is found that can process it. By default (i.e. without **smart_block**) the HTML plug-in blocks *all* images, which is appropriate for this collection.

Looking at different views of the files in the Gather and Enrich panels

12. Switch to the **Gather** panel and in the right-hand side open *englishhistory.net* → *tudor*.

13. Change the **Show Files** menu for the right-hand side from **All Files** to **HTM & HTML**. Notice the files displayed above are filtered accordingly, to show only files of this type.

14. Change the **Show Files** menu to **Images**. Again, the files shown above alter.

15. Now return the **Show Files** setting back to **All Files**, otherwise you may get confused later. Remember, if the **Gather** or **Enrich** panels do not seem to be showing all your files, this could be the problem.

1.6. Enhanced Word document handling

The standard way Greenstone processes Word documents is to convert them to HTML format using a third-party program, wvWare. This sometimes doesn't do a very good job of conversion. If you are using Windows, and have Microsoft Word installed, you can take advantage of Windows native scripting to do a better job of conversion. If the original document was hierarchically

structured using Word styles, these can be used to structure the resulting HTML. Word document properties can also be extracted as metadata.

1. In your digital library, preview the **reports** collection. Look at the HTML versions of the Word documents and notice how they have no structure-they have been converted to flat documents.

Using Windows native scripting

2. In the Librarian Interface, open up the **reports** collection. Switch to the **Design** panel and select the **Document Plugins** section on the left-hand side. Double click the **WordPlug** plugin and switch on the **windows_scripting** option.

   In the **Search Indexes** section, check the **section** checkbox to build the indexes on section level as well as document level.

3. **Build** the collection. You will notice that the Microsoft Word program is started up for each Word document—the document is saved as HTML from Word itself, to get a better conversion. **Preview** the collection. In the **Titles** list, notice that *word03.doc* and *word06.doc* now have a book icon, rather than a page icon. These now appear with hierarchical structure. But these two are the only ones.

   The default behaviour for **WordPlug** with **windows_scripting** is to section the document based on "Heading 1", "Heading 2", "Heading 3" styles. If you open up the *word03.doc* or *word06.doc* documents in Word, you will see that the sections use these Heading styles.

   Note, to view style information in Word, you can select **Format → Styles and Formatting** from the menu, and a side bar will appear on the right hand side. Click on a section heading and the formatting information will be displayed in this side bar.

4. Some of the documents do not use styles (e.g. *word01.doc*) and no structure can be extracted from them. Some documents use user-defined styles. **WordPlug** can be configured to use these styles instead of Heading 1, Heading 2 etc. Next we will configure **WordPlug** to use the styles found in *word05.doc*.

Modes in the Librarian Interface

5. The Librarian Interface can operate in four modes. Go to **File → Preferences... → Mode** and see the four modes and what functionality they provide access to. **Librarian** is the default mode.

6. Change the mode to **Library Systems Specialist** because you will need to use regular expressions to set up the style options in the next part of the exercise.

Defining styles

7. Open up *word05.doc* in Word (by double-clicking on it in the **Gather** pane), and examine the title and section heading styles. You will see that various user-defined header styles are set such as:
   - *PaperTitle*: Title of the paper
   - *SammaryHeader* (probably mistyped): Summary section
   - *Chapter Title*: Level 1 section heading
   - *SectionHeading*: Level 2 section heading
   - *Reference Heading*: Reference section

8. In the **Document Plugins** section of the **Design** panel, select **WordPlug** and click **<Configure Plugin...>**. Four types of header can be set which are:
   - `level1_header (level1Header1|level1Header2|...)`
   - `level2_header (level2Header1|level2Header2|...)`
   - `level3_header (level3Header1|level3Header2|...)`
   - `title_header (titleHeader1|titleHeader2|...)`

   These header options define which styles should be considered as title, level 1, level 2 and level 3 styles.

   Set the options as follows (spaces in the Word styles are removed when converting to HTML styles, and these options must match the HTML styles):

   ```
   level1_header:(SammaryHeader|ChapterTitle|ReferenceHeading)
   level2_header: SectionHeading
   title_header: PaperTitle
   ```

   *If you can't see these options in the **WordPlug** configuration pane, check that you are in **Library Systems Specialist** mode as described above.*

   Once these are set, click **<OK>**.

9. Close any documents that are still open in Word, as this can prevent the build process from completing correctly.

10. **Build** the collection and **preview** it. Look in particular at *word05.doc*. You will see that this document is now also hierarchically structured.

    If you have documents with different formatting styles, you can use `(...|...)` to specify all of the different styles.

Removing pre-defined table of contents

11. If you look at *word06.doc* you will see that it now has two tables of contents. One is generated by Greenstone based on the document's styles, the other was already defined in the Word document. WordPlug can be configured to remove predefined tables of contents and tables of figures. The tables must be defined with Word styles in order for this to work.

12. To remove the tables of contents and figures from *word06.doc*, switch on the **delete_toc** option in **WordPlug**. Set the **toc_header** option to

(MsoToc1|MsoToc2|MsoToc3|MsoTof). In this document, the table of contents and list of figures use these four style names. Click <**OK**>.

13. **Build** and **preview** the collection. *word06.doc* should now have only one table of contents.

14. Switch the Librarian Interface back to **Librarian** mode (**File → Preferences... → Mode**).

Extracting document properties as metadata

15. Word document properties can be extracted as metadata. By default, only the Title will be extracted. Other properties can be extracted using the **metadata_fields** option.

16. In the **Enrich** panel, look at the metadata that has been extracted for *word05.doc* and *word06.doc*. Now open the documents in Word and look at what properties have been set (**File → Properties**). They have Title, Author, Subject, and Keywords properties. **WordPlug** can be configured to look for these properties and extract them.

17. In the **Design** panel, under **Document Plugins**, configure **WordPlug** once again. Switch on the configuration option **metadata_fields**. Set the value to

    ```
    Title,Author<Creator>,Subject,Keywords<Subject>
    ```

    This will make **WordPlug** try to extract Title, Author, Subject and Keywords metadata. Title and Subject will be saved with the same name, while Author will be saved as Creator metadata; and Keywords as Subject metadata.

18. Make sure you have closed all the documents that were opened, then **rebuild** the collection.

19. Look at the metadata for the two documents again in the **Enrich** panel. You should now see ex.Creator and ex.Subject metadata items. This metadata can now be used in display or browsing classifiers etc.

****

LAB 2:

Greenstone: Adding metadata—and using it

2.1. A collection of Word and PDF files—Part B

In the Librarian Interface, open up the reports collection that you created in exercise 1.4. Remember that the extracted Title metadata for some documents was incorrect.

Manually adding metadata to documents in a collection

8. In the **Enrich** panel, manually add Dublin Core **dc.Title** metadata to those documents which have incorrect **ex.Title** metadata. Select *word03.doc* and double-click to open it. Copy the title of this document ("Greenstone: A comprehensive open-source digital library software system") and return to the Librarian Interface. Scroll up or down in the metadata table until you can see **dc.Title**. Click in the value box and paste in the metadata.

9. Now add **dc.Creator** information for the same document. You can add more than one value for the same field: when you press **Enter** in a metadata value field, a new empty field of the same type will be generated. Add each author separately as **dc.Creator** metadata.

10. Close the document (in Microsoft Word) when you have finished copying metadata from it. External programs opened when viewing documents must be closed before building the collection, otherwise errors can occur.

11. Next add **dc.Title** and **dc.Creator** metadata for a few of the other documents.

12. You will notice as you add more values, they appear in the **Existing values for ...** box below the metadata table. If you are adding the same metadata value to more than one document, you can select it from this list. For example, *pdf01.pdf* and *word03.doc* share the same Title; and many documents have common authors.

*If you build and preview your collection at this point, you will see that the Titles list now shows your new Titles. However, the **dc.Creator** metadata is not displayed. You need to alter the collection design to use this metadata.*

Document Plugins

13. In the Librarian Interface, look at the **Document Plugins** section of the **Design** panel, by clicking on this in the list to the left. Here you can add, configure or remove plugins to be used in the collection. There is no need to remove any plugins, but it will speed up processing a little. In this case we have only Word, PDF, RTF, and PostScript documents, and can remove the **ZIPPlug**, **TEXTPlug**, **HTMLPlug**, **EMAILPlug**, **ImagePlug**, **ISISPlug** and **NULPlug** plugins. To delete a plugin, select it and click **<Remove**

Plugin>. **GAPlug** is required for any type of source collection and should not be removed.

Search indexes

14. The next step in the **Design** panel is **Search Indexes**. These specify what parts of the collection are searchable (e.g. searching by title and author). Delete the **ex.Source** index, which is not particularly useful, by selecting it and clicking **<Remove Index>**.

15. Modify the **ex.Title** index to include **dc.Title** by selecting the index in the **Assigned Indexes** box and clicking **<Edit Index>**. Select **dc.Title** from the list of metadata, and click **<Replace Index>**. Searching this index will search both **dc.Title** and **ex.Title** metadata. If you want to restrict searching to just the manually added **dc.Title** metadata, edit the index again and deselect **ex.Title** from the list of metadata.

16. You can add indexes based on any metadata. Add a new index based on **dc.Creator** by clicking **<New Index>**. Select **dc.Creator** in the list of metadata, and click **<Add Index>**.

*The next section is **Partition Indexes**. In this exercise, we will not make any changes to this.*

Browsing classifiers

17. The **Browsing Classifiers** section adds "classifiers," which provide the collection with browsing functions. Go to this section and observe that Greenstone has provided two classifiers, *AZLists* based on **ex.Title** and **ex.Source** metadata. These correspond to the *Titles* and *Filenames* buttons on the collection's access bar.

    Remove the **ex.Source** classifier by selecting it and clicking **<Remove Classifier>**.

18. Modify the **ex.Title** classifier to use **dc.Title** instead. Select the classifier and click **<Configure Classifier...>**. In the **metadata** box, select **dc.Title** instead of **ex.Title**. Click **<OK>**.

19. Now add an **AZCompactList** classifier for **dc.Creator**. Select **AZCompactList** from the **Select classifier to add:** drop-down list and click **<Add Classifier...>**. A popup window **Configuring Arguments** appears. Select **dc.Creator** from the **metadata** drop-down list and click **<OK>**.

    AZCompactList is like **AZList**, except that values that appear multiple times in the hierarchy are automatically grouped together and a new node, shown as a bookshelf icon, is formed.

20. Switch to the **Create** panel, and **build** and **preview** the collection.

21. Check that all the facilities work properly. There should be three full-text indexes, called *text*, *dc.Title,ex.Title*, and *dc.Creator*. The *Titles* list should display all the documents to which you have assigned **dc.Title** metadata (and only those documents). The *Creators*

list should show one bookshelf for each author you have assigned as **dc.Creator**, and clicking on that bookshelf should take you to all the documents they authored.

Renaming the search indexes

22. The default display text for the indexes in the drop-down list on the search page contains the content of the index. Now we will change this display text to make it nicer. Go to the **Format** panel by clicking its tab. This panel is split into several sections, each controlling some aspect of collection presentation.

23. Select **Search** in the left hand list. This section allows you to modify what text is displayed for the drop-down lists in the search form (indexes, subcollections, levels etc). Set the **Display text** for the **dc.Title,Title** index to be "titles", and that for the **dc.Creator** index to be "creators". Preview the collection by clicking the **Preview Collection**. The search form should display the new text.

Classifying on multiple metadata

24. The new *Titles* list shows only those documents which have been assigned **dc.Title** metadata. For many documents, extracted Titles may be fine, and it is impractical to add the same metadata again as **dc.Title**. Fortunately there is a way we can use both metadata types in one classifier: specify a list of metadata names in the classifier.

25. In the **Browsing Classifiers** section of the **Design** panel, select the **AZList** for **dc.Title** in the **Assigned Classifiers** box and click **<Configure Classifier...>**. Note you can achieve the same result by double clicking on the classifier.

26. In the **metadata** field, type ",ex.Title" after the "dc.Title"—i.e. make it read

    ```
    dc.Title,ex.Title
    ```

27. If you have already done the **Enhanced Word document handling** exercise, some of the documents will have extracted ex.Creator metadata, and some will have dc.Creator. To use both of these in the Creators classifier, make a similar change to the **AZCompactList**: make the **metadata** field read `dc.Creator,ex.Creator`.

    You may notice that **AZCompactList** has two options after the **metadata** option: **firstvalueonly** and **allvalues**. Manually added metadata can be used to replace or enhance automatically extracted metadata, and these options control exactly which pieces of metadata a document is classified by.

    For example, say we have two documents. Document 1 has four Creators specified (dc.Creator = dcA, dc.Creator = dcB, ex.Creator = exA, ex.Creator = exB), while document 2 has three (ex.Creator = exA, ex.Creator = exB, ex.Creator = exC). The following table shows which metadata values each document is classified by, for the different classifier options:

AZCompactList options	Document 1	Document 2

-metadata dc.Creator,ex.Creator	dcA, dcB	exA, exB, exC
-metadata dc.Creator,ex.Creator -firstvalueonly	dcA	exA
-metadata dc.Creator,ex.Creator -allvalues	dcA, dcB, exA, exB	exA, exB, exC

28. **Build** the collection again and **preview** it. Now all of the documents should appear in the *Titles* list (and extracted Creators should appear in the *Creators* list).

    Extracted metadata is unreliable. But it is very cheap! On the other hand, manually assigned metadata is reliable, but expensive. The previous section of this exercise has shown how to aim for the best of both worlds by using extracted metadata but correcting it when it is wrong. While this may not satisfy the professional librarian, it could provide a useful compromise for the music teacher who wants to get their collection together with a minimum of effort.

Branding a collection with an image

29. Switch back to the **Format** panel. The first section **General** appears. This allows you to modify the values you provided when defining the collection, if desired. You can also brand the collection using a suitable image.

30. Click on the **<Browse...>** button associated with **URL to 'about page' image:**, and browse to the image *sample_files → Word_and_PDF → wrdpdf.gif* on your computer. When you select this image, Greenstone automatically generates an appropriate URL for the image. **Preview** the collection: you should see the new image at the top left of the page.

2.2. A simple image collection

1. In the Librarian Interface, start a new collection (**File → New...**) called **backdrop**. Fill out the fields with appropriate information. For **Base this collection on:**, select the item **Simple image collection (image-e)** from the pull-down menu.

   When you base a collection on an existing one, it inherits all the settings of the old one, including which metadata sets (if any) the collection uses.

2. Copy the images provided in *sample_files → images* into your newly-formed collection.

3. Change to the **Create** panel and **build** the collection.

4. **Preview** the result.

5. Click on **Browse** in the navigation bar to view a list of the photos ordered by filename and presented as a thumbnail accompanied by some basic data about the image. The structure of this collection is the same as **Simple image collection (image-e)**, but the content is different.

6. Back in the Librarian Interface, change to the **Enrich** panel and view the extracted metadata for *Bear.jpg*.

Adding a metadata set to the collection

We now add our own metadata and use it to give users a new way to browse the collection. We use the Dublin Core metadata set.

7. The collection (image-e) on which **backdrop** is based uses only extracted metadata. To add another metadata set, go to the **Enrich** panel of the Librarian Interface and click the **<Manage Metadata Sets...>** button underneath the file tree.

8. The window that pops up shows the metadata sets currently used by the collection. To add a new one, click **<Add...>**.

   In the window that pops up, select the Dublin Core metadata set from the list of available sets, and click **<Add>**. Close the **Manage Metadata Sets** dialog by clicking **<Close>**.

9. In the **Enrich** panel, the metadata for each file now shows the (empty) Dublin Core **dc.** fields as well as the extracted **ex.** fields.

Adding Title and Description metadata

10. We work with just the first three files (*Bear.jpg*, *Cat.jpg* and *Cheetah.jpg*) to get a flavour of what is possible. First, set each file's **dc.Title** field to be the same as its filename but without the filename extension:

    Click on *Bear.jpg* so its metadata fields are available, then click on its **dc.Title** field on the right-hand side. Type in **Bear**.

    Repeat the process for *Cat.jpg* and *Cheetah.jpg*.

11. Add a description for each image as **dc.Description** metadata.

    What description should you enter? To remind yourself of a file's content, the Librarian Interface lets you open files by double-clicking them. It launches the appropriate application based on the filename extension, Word for .doc files, Acrobat for .pdf files and so on.

    Double-click *Bear.jpg*: on Windows, the image will normally be displayed by Microsoft's Photo Editor (although this depends on how your computer has been set up).

    Back in the **Enrich** pane, make sure that *Bear.jpg* is selected in the collection tree on the left hand side. Enter the text **Bear in the Rocky Mountains** as the value for the **dc.Description** field.

    Repeat this process for *Cat.jpg* and *Cheetah.jpg*, adding a suitable description for each.

12. Go to the **Create** panel and click **<Build Collection>**. Once it has finished building, **preview** the collection. You will not notice anything new. That's because we haven't changed the design of the collection to take advantage of the new metadata.

Change Format Features to display new metadata

13. Now we customize the collection's appearance. Go to the **Format** panel and select **Format Features** from the left-hand list. Leave the feature selection controls at their default values, so that **All Features** is selected for **Choose Feature**, and **VList** is selected as the **Affected Component**. In the **HTML Format String**, edit the text as follows:
    - Change `_ImageName_:` to `Title:`
    - Change `[Image]` to `[dc.Title]`
    - After `[dc.Title]<br>` add `Description: [dc.Description]<br>`

    Metadata names are case-sensitive in Greenstone: it is important that you capitalize "Title" and "Description" (and don't capitalize "dc").

14. The new format statement is displayed in the list of assigned format statements. The first substitution alters the fragment of text that appears to the right of the thumbnail image, the second alters the item of metadata that follows it. The addition displays the description after the Title.

15. Preview the collection by clicking the **<Preview Collection>** button. When you click on **Browse** in the navigation bar the presentation has changed to "Title: Bear" and so on. Each image's description should appear beside the thumbnail, following the title.

After the first three items, the Title and Description become blank because we have only assigned Dublin Core metadata to these first three. To get a full listing, enter all the metadata.

*Changes in the **Format** panel take place immediately and you can see the result straightaway by clicking **reload (or refresh)** in the web browser. If you modify anything in the **Gather**, **Enrich** or **Design** panels, you will need to rebuild the collection.*

Changing the size of image thumbnails

16. Lets change the size of the thumbnail image and make it smaller. Thumbnail images are created by the **ImagePlug** plug-in, so we need to access its configuration settings. To do this, switch to the **Design** panel and select **Document Plugins** from the list on the left. Double-click **ImagePlug** to pop up a window that shows its settings. (Alternatively, select **ImagePlug** with a single click and then click **<Configure Plugin...>** further down the screen). Currently all options are off, so standard defaults are used. Select **thumbnailsize**, set it to **50**, and click **<OK>**.

17. **Build** and **preview** the collection.

18. Once you have seen the result of the change, return to the **Design** panel, select the configuration options for **ImagePlug**, and switch the **thumbnailsize** option off so that the thumbnail reverts to its normal size when the collection is re-built.

Adding a browsing classifier based on Description metadata

19. Now we'll add a new browsing option based on the descriptions. In the **Design** panel, select **Browsing Classifiers** from the left-hand list. Set the menu item for **Select classifier to add:** to **AZList**; then click **<Add Classifier...>**.

20. A window pops up to control the classifier's options. Set the **metadata** option to **dc.Description** and click **<OK>**.

21. **Build** the collection, and **preview** it. Choose the new **Descriptions** link that appears in the navigation bar.

*Only three items are shown, because only items with the relevant metadata (**dc.Description** in this case) appear in the list. The original browse list includes all photos in the collection because it is based on **ex.Image**, extracted metadata that reflects an image's filename, which is set for all images in the collection.*

Creating a searchable index based on Description metadata

22. Now we'll add an index so that the collection can be searched by descriptions. Switch to the **Design** panel and select **Search Indexes** from the left-hand list. Click the **<New Index>** button. Select **dc.Description** from the list of metadata to include in the index, leave **Indexing level:** at its default, "document", and click **<Add Index>**.

23. Switch to the **Create** panel, **build** the collection, then **preview** it. There is now a **Search** button in the navigation bar. As an example, search for the term "bear" in the *document:dc.Description* index (which is the only index at this point).

24. To change the text that is displayed for the index (document:dc.Description), go to the **Format** panel back in the Librarian Interface. Select **Search** from the left-hand list. This panel allows you to change the text that is displayed on the search form. Change the **Display text** for the **document:dc.Description** index to "descriptions" (or other suitable text). Go back to the browser and reload the search page. Your new text will appear in the search form.

2.3. Enhanced collection of HTML files—Tudor

We return to the Tudor collection and add metadata that expresses a subject hierarchy. Then we build a classifier that exploits it by allowing readers to browse the documents about Monarchs, Relatives, Citizens, and Others separately.

Adding hierarchically-structured metadata and a Hierarchy classifier

1. Open up your **tudor** collection (the original version, not the **webtudor** version), switch to the **Enrich** panel and select the *citizens* folder (a subfolder of *englishhistory.net → tudor*). Set its **dc.Subject and Keywords** metadata to **Tudor period|Citizens**. The vertical bar ("|") is a hierarchy marker. Selecting a *folder* and adding metadata has the

effect of setting this metadata value for all files contained in this folder, its subfolders, and so on. A popup alerts you to this fact. Click **<OK>** to close the popup.

2. Repeat for the *monarchs* and *relative* folders, setting their **dc.Subject and Keywords** metadata to **Tudor period|Monarchs** and **Tudor period|Relatives** respectively. Note that the hierarchy appears in the **Existing values for dc.Subject and Keywords** area.

   If you don't want to see the popup each time you add folder level metadata, tick the **Do not show this warning again** checkbox; it won't be displayed again.

3. Finally, select all remaining files—the ones that are not in the *citizens*, *monarchs*, or *relative* folders—by selecting the first and shift-clicking the last. Set their **dc.Subject and Keywords** metadata to **Tudor period|Others**: this is done in a single operation (there is a short delay before it completes).

   When multiple files are selected in the left hand collection tree, all metadata values for all files are shown on the right hand side. Items that are common to all files are displayed in black—e.g. **dc.Subject and Keywords**—while others that pertain to only one or some of the files are displayed in grey—e.g. any extracted metadata.

   Metadata inherited from a parent folder is indicated by a folder icon to the left of the metadata name. Select one of the files in the *relative* folder to see this.

4. Switch to the **Design** panel and select **Browsing Classifiers** from the left-hand list. Set the menu item for **Select classifier to add:** to **Hierarchy**; then click **<Add Classifier...>**.

5. A window pops up to control the classifier's options. Change the **metadata** to **dc.Subject and Keywords** and then click **<OK>**.

6. For tidiness' sake, **remove** the **classifier** for **Source** metadata (included by default) from the list of currently assigned classifiers, because this adds little to the collection.

7. Now switch to the **Create** panel, **build** the collection, and **preview** it. Choose the new **Subjects** link that appears in the navigation bar, and click the bookshelves to navigate around the four-entry hierarchy that you have created.

Adding a hierarchical phrase browser (PHIND)

Next we'll add an interactive hierarchical phrase browsing classifier to this collection.

8. Switch to the **Design** panel and choose the **Browsing Classifiers** item from the left-hand list.

9. Choose **Phind** from the **Select classifier to add:** menu. Click **<Add Classifier...>**. A window pops asking for configuration options: leave the values at their preset defaults (this will base the phrase index on the full text) and click **<OK>**.

10. **Build** the collection again, **preview** it, and try out the new **Phrases** option in the navigation bar. An interesting PHIND search term for this collection is "king". Note that even though it is called a phrase browser, only single terms can be used as the starting point for browsing.

Partitioning the full-text index based on metadata values

Next we partition the full-text index into four separate pieces. To do this we first define four subcollections obtained by "filtering" the documents according to a criterion based on their ***dc.Subject and Keywords*** *metadata. Then an index is assigned to each subcollection. This will enable users to restrict a search to a subset of the documents.*

11. Switch to the **Design** panel, and click **Partition Indexes**. This feature is disabled because you are operating in **Librarian** mode (this is indicated in the title bar at the top of the window).

12. Switch to **Library Systems Specialist** mode by going to **Preferences...** (on the **File** menu) and clicking **<Mode>**. Read about the other modes too.

13. Return to the **Partition Indexes** section of the **Design** panel. Ensure that the **Define Filters** tab is selected (the default). Define a subcollection filter with name **monarchs** that matches against **dc.Subject and Keywords**, and type **Monarchs** as the regular expression to match with. Click **<Add Filter>**. This filter includes any file whose **dc.Subject and Keywords** metadata contains the word *Monarchs*.

14. Define another filter, **relatives**, which matches **dc.Subject and Keywords** against the word **Relatives**. Define a third and fourth, **citizens** and **others**, which matches it against the words **Citizens** and **Others** respectively.

15. Having defined the subcollection filters, we partition the index into corresponding parts. Click the **Assign Partitions** tab. Select the citizens subcollection and click **<Add Partition>**. Next select monarchs, and click **<Add Partition>**. Repeat for the other two subcollections, so that you end up with four partitions, one based on each subcollection filter.

    The order they appear in the **Assigned Subcollection Partitions** list is the order they will appear in the drop down menu on the search page. You can change the order by using the **<Move Up>** and **<Move Down>** buttons.

16. **Build** and **preview** the collection.

17. The search page includes a pulldown menu that allows you to select one of these partitions for searching. For example, try searching the *relatives* partition for *mary* and then search the *monarchs* partition for the same thing.

18. To allow users to search the collection as a whole as well as each subcollection individually, return to the **Partition Indexes** section of the **Design** panel and select the

Assign Partitions tab. Select all four subcollections by checking their boxes and click **<Add Partition>**.

19. To ensure that the combined index appears first in the list on the reader's web page, use the **<Move Up>** button to get it to the top of the list here in the **Design** panel. Then **build** and **preview** the collection.

20. Search for a common term (like *the*) in all five index partitions, and check that the numbers of words (not documents) add up.

21. The text in the drop down box on the search page is based on the filters each partition was built on. To change the text that is displayed, go to the **Search** section of the **Format** panel. The single filter partitions have sensible default text, but the combined partition does not. Set the **Display text** for the combined partition to "all". **Preview** the collection.

22. In the Librarian Interface, return to **Librarian** mode, using **Preferences...** (on the **File** menu).

Controlling the building process

Finally we look at how the building process can be controlled. Developing a new collection usually involves numerous cycles of building, previewing, adjusting some enrich and design features, and so on. While prototyping, it is best to temporarily reduce the number of documents in the collection. This can be accomplished through the ***maxdocs*** *parameter to the building process.*

23. Switch to the **Create** panel and view the options that are displayed in the top portion of the screen. Select **maxdocs** and set its numeric counter to **3**. Now **build**.

24. Preview the newly rebuilt collection's **Titles** page. Previously this listed more than a dozen pages per letter of the alphabet, but now there are just three—the first three files encountered by the building process.

25. Go back to the **Create** panel and turn off the **maxdocs** option. **Rebuild** the collection so that all the documents are included.

****

LAB 3:

GSDL: Advanced collection configuration

3.1. Formatting the Word and PDF collection

In this exercise, we play around with the format statements in the Word and PDF collection.

1. Open the **reports** collection in the Librarian Interface and go to the **Format Features** section of the **Format** panel.

Tidying up the default format statement

2. In this part of the exercise, we make the format statement simpler without changing the resulting display.

   Greenstone's default format statement is complex because it is designed to produce something reasonable under almost any conditions, and also because for practical reasons it needs to be backwards compatible with legacy collections. For this collection, we don't need all of the complexity.

   Make sure that the **VList** format statement is selected in the list of formats.

   The default **VList** format statement looks like the following:

```
<td valign="top">[link][icon][/link]</td>
<td
valign="top">[ex.srclink]{Or}{[ex.thumbicon],[ex.srcicon]}[ex./srclink]</t
d>
<td valign="top">[highlight]
{Or}{[dls.Title],[dc.Title],[ex.Title],Untitled}
[/highlight]{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
```

   This format statement is the default used for any vertical list, such as search results, classifiers, and document table of contents.

   `{Or}{[ex.thumbicon],[ex.srcicon]}` chooses *ex.thumbicon* metadata if its there, otherwise chooses *ex.srcicon* metadata. If neither are present, nothing is displayed. For this collection there is no *ex.thumbicon* metadata so the choice is not needed.

   Replace `{Or}{[ex.thumbicon],[ex.srcicon]}` with `[ex.srcicon]`.

   The resulting format statement looks like the following:

```
<td valign=top>[link][icon][/link]</td>
<td valign=top>[ex.srclink][ex.srcicon][ex./srclink]</td>
```

```
<td valign=top>[highlight]
{Or}{[dc.Title],[ex.Title],Untitled}[/highlight]
{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
```

Preview the collection to make sure the display hasn't changed. You shouldn't notice any difference when looking at search results, classifiers etc.

Linking to Greenstone version or original version of documents

3. For collections with documents that undergo a conversion process during importing (e.g. Word, PDF, PowerPoint documents, but not text, HTML documents), the original file is stored in the collection along with the converted version. The default **VList** format statement links to both versions:

   `[link][icon][/link]` links to the Greenstone HTML version, while `[srclink][srcicon][/srclink]` links to the original.

   Choose **SearchVList** in **Format Features** by selecting **Search** from the **Choose Feature** drop down list, and **VList** from the **Affected Component** list. Click **<Add Format>** to add the **SearchVList** format statement into the list of assigned formats. Experiment with removing either of the two links from the format statement.

   To see the results of your changes, preview the collection and do a search. You are making changes to **SearchVList**, which means the changes will only apply to search results.

   Storing and displaying the original allows users to see the correct format, but requires the user to have the relevant program installed. It also increases the size of the collection. The Greenstone version can be viewed in a browser, but may not look as nice.

Making bookshelves show how many items they contain

4. Next, we'll customize the format for the *Creators* list. Classifier bookshelves have only a few pieces of metadata to display: `[ex.Title]` and `[numleafdocs]`. Whatever metadata the classifier has been built on, the bookshelf label is always stored as `[ex.Title]`. This is why a Creator is printed out for each bookshelf even though `[dc.Creator]` is not specified in the format statement. `[numleafdocs]` is only defined for bookshelves, so this metadata can be used in an `{If}` statement to make bookshelves and documents display differently in the list.

   Make each bookshelf in the Creator classifier show how many entries it contains. In the **Format Features** section of the **Format** panel, select the **CL2 AZCompactList** classifier which is based on **dc.Creator** metadata from the **Choose Feature** drop down list, and **VList** from the **Affected Component** list. Click the **<Add Format>** button to add this format into the list of assigned formats. Note that it gets added as **CL2VList** in this list: it is the **VList** format for the second (**CL2**) classifier.

   Append the following text to the bottom of the format statement:

```
{If}{[numleafdocs],<td><i>([numleafdocs])</i></td>}
```

Preview the collection. Click on the *Creators* list and notice that the bookshelves now display how many documents they contain.

This revised format statement has the effect of specifying in brackets how many items are contained within a bookshelf. Since only bookshelves define `[numleafdocs]`, only they will display this. By modifying **CL2VList** instead of **VList**, the change will only apply to the second classifier (Creators).

Displaying multi-valued metadata

5. Next we modify the document entries in the Creator classifier to display all authors. Back in **Format Features**, select the **CL2VList** format in the list of assigned formats. After `{If}{[ex.Source],<br>` in the format statement, add `[sibling:dc.Creator]`.

   `[ex.Source]` is not defined for bookshelves, so can also be used to differentiate bookshelves and documents.

   The resulting format statement looks like:

```
<td valign=top>[link][icon][/link]</td>
<td valign=top>[ex.srclink][ex.srcicon][ex./srclink]</td>
<td valign=top>[highlight]
{Or}{[dc.Title],[ex.Title],Untitled}[/highlight]
{If}{[ex.Source],<br>[sibling:dc.Creator]
<i>([ex.Source])</i>}</td>
{If}{[numleafdocs],<td><i>([numleafdocs])</i></td>}
```

   This will display the Greenstone link, the link to the original, then the Title. For bookshelves, it will also display how many documents the bookshelf contains. For documents, it will display all the Authors (Creators), and the source document. `[sibling:dc.Creator]` displays all the Creator metadata for the document, separated by a space (" "), while `[dc.Creator]` displays only the first author. Preview the *Creators* list and make sure that all authors are displayed for documents.

6. You can change the separator between the authors. Modify the format statement, and replace `[sibling:dc.Creator]` with `[sibling(All'<br/>'):dc.Creator]`. This will add a new line after each author (`<br/>` specifies a line break in HTML). Preview the *Creators* list.

   If you have done exercise **Enhanced Word document handling**, the collection will have both dc.Creator and ex.Creator metadata. To display both, you can use

```
[sibling:dc.Creator] [sibling:ex.Creator]
```

   To display dc.Creator if it is present, otherwise display ex.Creator, use

```
{Or}{[sibling:dc.Creator],[sibling:ex.Creator]}
```

Opening PDF files with query terms highlighted

7. Next we'll customize the **SearchVList** format statement to highlight the query terms in a PDF file when it is opened from the search result list. This requires Acrobat Reader 7.0 version or higher, and currently only works on a Microsoft Windows platform.

8. The search terms are kept in the macro variable **_cgiargq_**, and we append **#search="_cgiargq_"** to the end of a PDF file link to pass the query terms to the PDF file.

   PDFPlug renames each PDF file as **doc.pdf** and saves it in a unique directory for that document, so we use

   ```
   _httpcollection_/index/assoc/[archivedir]/doc.pdf
   ```

   to refer to the PDF source file. (However, if you used the **-keep_original_filename** option to **PDFPlug** when building the collection, the original name of the PDF file is kept, and we use

   ```
   _httpcollection_/index/assoc/[archivedir]/[Source]
   ```

   instead to locate the PDF source file.)

9. Select **SearchVList** from the list of assigned formats. We need to test whether the file is a PDF file before linking to doc.pdf, using `[If]{[ex.FileFormat] eq 'PDF',,}`. For PDF files, we use the above format instead of the `[ex.srclink]` and `[ex./srclink]` variables to link to the file.

   The resulting format statement is:

   ```
   <td valign="top">[link][icon][/link]</td>
   <td valign="top">{If}{[ex.FileFormat] eq 'PDF', <a
   href=\"_httpcollection_/index/assoc/[archivedir]/doc.pdf#search=&quot;_cgi
   argq_&quot;\">[ex.srcicon]</a>,
   [ex.srclink][ex.srcicon][ex./srclink]}</td>
   <td valign="top">[highlight]
   {Or}{[dc.Title],[ex.Title],Untitled}
   [/highlight]{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
   ```

   When the PDF icons are clicked in the search results, Acrobat will open the file with the search window open, and the query terms highlighted.

3.2. Formatting the HTML collection—Tudor

1. Open up your **tudor** collection, go to the **Format** panel (by clicking on its tab) and select **Format Features** from the left-hand list. Leave the editing controls at their default value, so

that **Choose Feature** displays *All Features* and **VList** is selected as the **Affected Component**. The text in the **HTML Format String** box reads as follows:

```
<td valign=top>[link][icon][/link]</td>
<td valign=top>[ex.srclink](Or){[ex.thumbicon],[ex.srcicon]}
[ex./srclink]</td>
<td valign=top>[highlight]
(Or){[dls.Title],[dc.Title],[ex.Title],Untitled}
[/highlight](If){[ex.Source],<br><i>([ex.Source])</i>}</td>
```

This displays something that looks like this:


A discussion of question five from Tudor Quiz: Henry VIII
(quizstuff.html)

for a particular document whose *Title* metadata is **A discussion of question five from Tudor Quiz: Henry VIII** and whose *Source* metadata is **quizstuff.html**.

This format appears in the search results list, in the **Titles** list, and also when you get down to individual documents in the **Subjects** hierarchy. This is Greenstone's default format statement.

Greenstone's default format statement is complex because it is designed to produce something reasonable under almost any conditions, and also because for practical reasons it needs to be backwards compatible with legacy collections.

2. Delete the contents of the **HTML Format String** box and replace it with this simpler version:

```
<td>[link][icon][/link]</td>
<td>[ex.Title]<br>
    <i>([ex.Source])</i>
</td>
```

Preview the result (you don't need to build the collection, because changes to format statements take effect immediately). Look at some search results and at the **Titles** list. They are just the same as before! Under most circumstances this far simpler format statement is entirely equivalent to Greenstone's more complex default.

*But there's a problem. Beside the bookshelves in the **Subjects** browser, beneath the subject appears a mysterious "()". What is printed for these bookshelves is governed by the same format statement, and though bookshelf nodes of the hierarchy have associated* Title *metadata—their title is the name of the metadata value associated with that bookshelf—they do not have **ex.Source** metadata, so it comes out blank.*

3. In the **Format Features** section of the **Format** panel, the **Choose Feature** menu (just above **Affected Component** menu) displays *All Features*. That implies that the same format is used

for the search results, titles, and all nodes in the subject hierarchy—including internal nodes (that is, bookshelves). The **Choose Feature** menu can be used to restrict a format statement to a specific one of these lists. We will override this format statement for the hierarchical *subject* classifier. In the **Choose Feature** menu, scroll down to the item that says

CL2: Hierarchy -metadata dc.Subject and Keywords

and select it. This is the format statement that affects the second classifier (i.e., "CL2"), which is a **Hierarchy** classifier based on **dc.Subject and Keywords** metadata.

Click **<Add Format>** to add this format statement to the collection.

Edit the **HTML Format String** box below to read

```
<td>[link][icon][/link]</td>
<td>[ex.Title]</td>
```

4. **Preview** the **Subjects** list in the collection. First, the offending "()" has disappeared from the bookshelves. Second, when you get down to a list of documents in the subject hierarchy, the filename does not appear beside the title, because **ex.Source** is not specified in the format statement and this format statement applies to all nodes in the *subject* classifier. Note that the search results and titles lists have not changed: they still display the filename underneath the title.

5. Let's change the search results format so that **dc.Subject and Keywords** metadata is displayed here instead of the filename. In the **Choose Feature** menu (under **Format Features** on the **Format** panel), scroll down to the item **Search** and select it. Click **<Add Format>** to add this format statement to the collection. Change the **HTML Format String** box below to read

```
<td>[link][icon][/link]</td>
<td>[ex.Title]<br>
    [dc.Subject]
</td>
```

6. To insert the **[dc.Subject]**, position the cursor at the appropriate point and either type it in, or select it from the **Insert Variable...** drop down menu. This menu shows many of the things that you can put in square brackets in the format statement.

7. **Preview** the collection. Documents in the search results list will be displayed like this:


A discussion of question five from Tudor Quiz: Henry VIII
Tudor period|Others

8. (The vertical bar appears because this **dc.Subject and Keywords** metadata is hierarchical metadata. Unfortunately there is no way to get at individual components of the hierarchy. For most metadata, such as title and author, this isn't a problem.)

8. Finally, let's return to the *Subjects* hierarchy and learn how to do different things to the bookshelves and to the documents themselves. In the **Choose Feature** menu, re-select the item

   CL2: Hierarchy -metadata dc.Subject and Keywords

   Edit the **HTML Format String** box below to read

```
<td>[link][icon][/link]</td>
<td>{If}{[numleafdocs],<b>Bookshelf title:</b> [ex.Title],
        Title:</b> [ex.Title]}
</td>
```

   Again, you can insert the items in square brackets by selecting them from the **Insert Variable...** drop down box.

   The If statement tests the value of the variable ***numleafdocs****. This variable is only set for internal nodes of the hierarchy, i.e. bookshelves, and gives the number of documents below that node. If it is set we take the first branch, otherwise we take the second. Commas are used to separate the branches. The curly brackets serve to indicate that the If is special—otherwise the word "If" itself would be output.*

9. **Preview** the collection and examine the subject hierarchy again to see the effect of your changes. Bookshelves should say **Bookshelf title:** and then the title, while documents will display **Title:** and the title. Note that the number of documents in the bookshelf is not displayed: we are using `[numleafdocs]` to test what kind of item in the list we are at, but we are not displaying it.

3.3. Section tagging for HTML documents

1. In a browser, take a look at the Greenstone demo collection. Browse to one of the documents. This collection is based on HTML files, but they appear structured in the collection. This is because these HTML files were tagged by hand into sections.

2. Using a text editor (e.g. WordPad) open up one of the HTML files from the demo collection: *Greenstone → collect → demo → import → fb33fe →fb33fe.htm*. You will see some HTML comments which contain section information for Greenstone. They look like:

```
<!--
<Section>
  <Description>
    <Metadata name="Title">Farming snails 1: Learning about snails;
```

```
    Building a pen; Food and shelter plants</Metadata>
  </Description>
-->

<!--
</Section>
<Section>
  <Description>
    <Metadata name="Title">Dew and rain</Metadata>
  </Description>
-->
```

When Greenstone encounters a `<Section>` tag in one of these comments, it will start a new subsection of the document. This will be closed when a `</Section>` tag is encountered. Metadata can also be added for each section—in this case, **Title** metadata has been added for each section. In the browser, find the **Farming snails 1** document in the demo collection (through the *Titles* browser). Look at its table of contents and compare it to the `<Section>` tags in the HTML document.

3. Add a new Section into this document. For example, lets add a new subsection into the **Introduction** chapter. In the text editor, add the following just after the Section tag for the **Introduction** section:

```
<!--
<Section>
  <Description>
    <Metadata name="Title">Snails are good to eat.</Metadata>
  </Description>
-->
```

Then just before the next section tag (**What do you need to start?**), add the following:

```
<!--
</Section>
-->
```

The effect of these changes is to make a new subsection inside the **Introduction** chapter.

4. Open the Greenstone demo collection in the Librarian Interface. In the **Document Plugins** section of the **Design** panel, note that **HTMLPlug** has the **description_tags** option set. This option is needed when `<Section>` tags are used in the source documents.

5. In the **Search Indexes** section, check the **section** checkbox to build the indexes on section level as well as document level.

6. **Build** and **preview** the collection. Look at the **Farming snails 1** document again and check that your new section has been added.

3.4. Exporting a collection to CD-ROM/DVD

*To publish a collection on CD-ROM or DVD, Greenstone's Export to CD-ROM export module must be installed. This is included with CD-ROM distributions, and all distributions 2.70w and later. It must be installed separately for non-CD-ROM versions of Greenstone, version 2.70 and earlier (see **Installing Greenstone**).*

1. Launch the Greenstone Librarian Interface if it is not already running.

2. Choose **File → Write CD/DVD image...**. In the resulting popup window, select the collection or collections that you wish to export by ticking their check boxes. You can optionally enter a name for the CD-ROM: this is the name that will appear in the menu when the CD-ROM is run. If a name is not entered, the default **Greenstone Collections** will be used. You can also specify whether the resulting CD-ROM will install files onto the host machine when used or not. Click **<Write CD/DVD image>** to start the export process.

   The necessary files for export are written to:

   Greenstone → tmp → exported_xxx

   where xxx will be similar to the name you have entered. If you didn't specify a name for the CD-ROM, then the folder name will be *exported_collections*.

   You need to use your own computer's software to write these on to CD-ROM. On *Windows XP* this ability is built into the operating system: assuming you have a CD-ROM or DVD writer insert a blank disk into the drive and drag the *contents* of *exported_xxx* into the folder that represents the disk.

   The result will be a self-installing Windows Greenstone CD-ROM or DVD, which starts the installation process as soon as it is placed in the drive.

3.5. Enhanced PDF handling

Greenstone converts PDF files to HTML using third-party software: *pdftohtml.pl*. This lets users view these documents even if they don't have the PDF software installed. Unfortunately, sometimes the formatting of the resulting HTML files is not so good.
This exercise explores some extra options to the PDF plugin which may produce a nicer version for display. Some of these options use the standard pdftohtml program, others use ImageMagick and Ghostscript to convert the file to a series of images. Ghostscript is a program that can convert Postscript and PDF files to other formats. You can download it from http://www.cs.wisc.edu/~ghost/ (follow the link to the current stable release).

1. In the Librarian Interface, start a new collection called "PDF collection" and base it on -- **New Collection** --.

   In the **Gather** panel, drag just the PDF documents from *sample_files → Word_and_PDF → Documents* into the new collection. Also drag in the PDF documents from *sample_files → Word_and_PDF → difficult_pdf*.

Go to the **Create** panel and build the collection. Examine the output from the build process. You will notice that one of the documents could not be processed. The following messages are shown: "The file pdf05-notext.pdf was recognised but could not be processed by any plugin.", and "5 documents were processed and included in the collection. 1 was rejected".

2. Preview the collection and view the documents. *pdf05-notext.pdf* does not appear as it could not be processed. *pdf06-weirdchars.pdf* was processed but looks very strange. The other PDF documents appear as one long document, with no sections.

Modes in the Librarian Interface

The Librarian Interface can operate in different modes. The default mode is ***Librarian*** *mode. We can use* ***Expert*** *mode to work out why the pdf file could not be processed.*

3. Use the **Preferences...** item on the **File** menu to switch to **Expert** mode and then build the collection again. The **Create** panel looks different in **Expert** mode because it gives more options: locate the **<Build Collection>** button, near the bottom of the window, and click it. Now a message appears saying that the file could not be processed, and why. Amongst all the output, we get the following message: "Error: PDF contains no extractable text. Could not convert pdf05notext.pdf to HTML format". pdftohtml.pl cannot convert a PDF file to HTML if the PDF file has no extractable text.

4. We recommend that you switch back to **Librarian** mode for subsequent exercises, to avoid confusion.

Splitting PDFs into sections

5. In the **Document Plugins** section of the **Design** panel, configure **PDFPlug**. Switch on the **use_sections** option.

   In the **Search Indexes** section, check the **section** checkbox to build the indexes on section level as well as document level.

   Build and **preview** the collection. View the text versions of some of the PDF documents. Note that these are now split into a series of pages, and a "go to page" box is provided. The format is still a bit ugly though, and pdf05-notext.pdf is still not processed.

Using image format

6. If conversion to HTML doesn't produce the result you like, PDF documents can be converted to a series of images, one per page. This requires ImageMagick and Ghostscript to be installed.

7. In the **Document Plugins** section, configure **PDFPlug**. Set the **convert_to** option to one of the image types, e.g. **pagedimg_jpg**. Switch off the **use_sections** option, as it is not used with image conversion.

8. **Build** the collection and **preview**. All PDF documents (including pdf05-notext.pdf) have been processed and divided into sections, but each section displays "This document has no text.". For the conversion to images for PDF documents, no text is extracted.

9. In order to view the documents properly, you will need to modify the format statement. In the **Format Features** section on the **Format** panel, select the **DocumentText** format statement. Replace

   ```
   [Text]
   ```

   with

   ```
   [srcicon]
   ```

10. Preview the collection. Images from the document are now displayed instead of the extracted text. Both *pdf05-notext.pdf* and *pdf06-weirdchars.pdf* display nicely now.

    In this collection, we only have PDF documents and they have all been converted to images. If we had other document types in the collection, we should use a different format statement, such as:

    ```
    {If}{[parent:FileFormat] eq PDF,[srcicon],[Text]}
    ```

    ***FileFormat** is an extracted metadata item which shows the format of the source document. We can use this to test whether the documents are PDF or not: for PDF documents, display [srcicon], for other documents, display [Text].*

Using process_exp to control document processing (advanced)

11. Processing all of the PDF documents using an image type may not give the best result for your collection. The images will look nice, but as no text is extracted, searching the full text will not be available for these documents. The best solution would be to process most of the PDF files as HTML, and only use the image format where HTML doesn't work.

12. We achieve this by putting the problem files into a separate folder, and adding another **PDFPlug** plugin with different options.

13. Go to the **Gather** panel. Make a new folder called "notext": right click in the collection panel and select **New folder** from the menu. Change the **Folder Name** to "notext", and click **<OK>**.

Move the two pdf files that have problems with html (*pdf05-notext.pdf* and *pdf06-weirdchars*.pdf) into this folder by drag and drop. We will set up the plugins so that PDF files in this *notext* folder are processed differently to the other PDF files.

14. Change to **Library Systems Specialist** mode so that you can add two of the same plugin, and use regular expressions in the plugin options (**File → Preferences... → Mode**).

    For version 2.71, you'll need to close GLI now then restart it to get the list of plugins to update properly.

15. Switch to the **Document Plugins** section of the **Design** panel. Add a second PDF plugin by selecting **PDFPlug** from the **Select plugin to add:** drop-down list, and clicking **<Add Plugin...>**. This plugin will come after the first PDF plugin, so we configure it to process PDF documents as HTML. Set the **convert_to** option to **html**, and switch on the **use_sections** option. Click **<OK>**.

16. Configure the first PDF plugin, and set the **process_exp** option to **'notext.*\.pdf'**.

17. The two PDF plugins should have options like the following:

    ```
    plugin PDFPlug -convert_to pagedimg_jpg -process_exp 'notext.*\.pdf'
    plugin PDFPlug -convert_to html -use_sections
    ```

    The *paged_img* version must come earlier in the list than the *html* version. The **process_exp** for the first **PDFPlug** will process any PDF files in the *notext* directory. The second **PDFPlug** will process any PDF files that are not processed by the first one.

    Note that all plugins have the **process_exp** option, and this can be used to customize which documents are processed by which plugin. This option is only visible in **Library Systems Specialist** and **Expert** modes.

    Change back to **Librarian** mode.

18. Edit the **DocumentText** format statement. PDF files processed as HTML will not have images to display, so we need to make sure they get text displayed instead. Change `[srcicon]` to `(Or){[srcicon],[Text]}`.

19. Build and preview the collection. All PDF documents should look relatively nice. Try searching this collection. You will be able to search for the PDFs that were converted to HTML (try e.g. "bibliography"), but not the ones that were converted to images (try searching for "banana" or "METS").

LAB 4:

Greenstone: Multimedia and Scanned Images

4.1. Looking at a multimedia collection

1. Copy the entire folder

   sample_files → *beatles* → *advbeat_large*

   (with all its contents) into your Greenstone *collect* folder. If you have installed Greenstone in the usual place, this is

   My Computer → *Local Disk (C:)* → *Program Files* → *Greenstone* → *collect*

   Put *advbeat_large* in there.

2. If the Greenstone Digital Library Local Library Server is already running, re-start it by clicking the CD icon on the task bar and then pressing *Restart Library*. If not, start it up by selecting *Greenstone Digital Library* from the *Start* menu.

3. Explore the Beatles collection. Note how the *Browse* button divides the material into seven different types. Within each category, the documents have appropriate icons. Some documents have an audio icon: when you click these you hear the music (assuming your computer is set up with appropriate player software). Others have an image thumbnail: when you click these you see the images.

4. Look at the *Titles* browser. Each title has a bookshelf that may include several related items. For example, *Hey Jude* has a MIDI file, lyrics, and a discography item.

5. Observe the low quality of the metadata. For example, the four items under **A Hard Day's Night** (under "H" in the *Titles* browser) have different variants as their titles. The collection would have been easier to organize had the metadata been cleaned up manually first, but that would be a big job. Only a tiny amount of metadata was added by hand—fewer than ten items. The original metadata was left untouched and Greenstone facilities used to clean it up automatically. (You will find in **Building a multimedia collection** that this is possible but tricky.)

6. In the Windows file browser, take a look at the files that makes up the collection, in the

   sample_files → *beatles* → *advbeat_large* → *import*

   folder. What a mess! There are over 450 files under seven top-level sub-folders. Organization is minimal, reflecting the different times and ways the files were gathered. For example, *html_lyrics* and *discography* are excerpts of web sites, and *images* contains

various images in JPEG format. For each type, drill down through the hierarchy and look at a sample document.

4.2. Building a multimedia collection

We will proceed to reconstruct from scratch the Beatles collection that you have just looked at. We develop the collection using a small subset of the material, purely to speed up the repeated rebuilding that is involved.

1. Start a new collection (**File → New...**) called **small beatles**, basing it on the default -- **New Collection** --. (Basing it on the existing Advanced Beatles collection would make your life far easier, but we want you to learn how to build it from scratch!)

2. Copy the files provided in

   sample_files → beatles → advbeat_small

   into your new collection. Do this by opening up *advbeat_small*, selecting the eight items within it (from *discography* to *beatles_midi.zip*), and dragging them across. Because some of these files are in MP3 and MARC formats you will be asked whether to include **MP3Plug** and **MARCPlug** in your collection. Click **<Add Plugin>**.

3. Change to the **Enrich** panel and browse around the files. There is no metadata—yet. Recall that you can double-click files to view them.

   (There are no MIDI files in the collection: these require more advanced customisation because there is no MIDI plugin. We will deal with them later.)

4. Change to the **Create** panel and **build** the collection.

5. **Preview** the result.

Manually correcting metadata

6. You might want to correct some of the metadata—for example, the atrocious misspelling in the titles "MAGICAL MISTERY TOUR." These documents are in the discography section, with filenames that contain the same misspelling. Locate one of them in the **Enrich** panel. Notice that the extracted metadata element **ex.Title** is now filled in, and misspelt. You cannot correct this element, for it is extracted from the file and will be re-extracted every time the collection is re-built.

7. Instead, add **dc.Title** metadata for these two files: "Magical Mystery Tour." Change to the **Enrich** panel, open the discography folder and drill down to the individual files. Set the **dc.Title** value for the two offending items.

Now there's a twist. The dc.Title metadata won't appear in Titles because the classifier has been instructed to use ex.Title. But changing the classifier to use dc.Title would miss out all the

extracted titles! Fortunately, there's a way of dealing with this by specifying a list of metadata names in the classifier.

8. Change to the **Design** panel and select the **Browsing Classifiers** section. Double-click the **ex.Title** classifier (the first one) to edit its configuration settings.
    - Type `dc.Title,` before the `ex.Title` in the metadata box—i.e. make it read

      ```
      dc.Title,ex.Title
      ```

      and click **<OK>**.

9. **Build** the collection again, and **preview** it.
10. Extracted metadata is unreliable. But it is very cheap! On the other hand, manually assigned metadata is reliable, but expensive. The previous section of this exercise has shown how to aim for the best of both worlds by using extracted metadata but correcting it when it is wrong. While this may not satisfy the professional librarian, it could provide a useful compromise for the music teacher who wants to get their collection together with a minimum of effort.

Browsing by media type

9. First let's remove the **AZList** classifier for filenames, which isn't very useful, and replace it with a browsing structure that groups documents by category (discography, lyrics, audio etc.). Categories are defined by manually assigned metadata.
    - Change to the **Enrich** panel, select the folder *discography* and set its **dc.Format** metadata value to "Discography". Setting this value at the folder level means that all files within the folder inherit it.
    - Repeat the process. Assign "Lyrics" to the *html_lyrics* folder, "Images" to *images*, "MARC" to *marc*, "Audio" to *mp3*, "Tablature" to *tablature_txt*, and "Supplementary" to *wordpdf*.
    - Switch to the **Design** panel and select the **Browsing Classifiers** section.
    - Delete the **ex.Source** classifier (the second one).
    - Add an **AZCompactList** classifier. Select **dc.Format** as the **metadata** field and specify "browse" as the **buttonname**. Click the **sort** checkbox, and select **ex.Title** in the drop-down list: this will make the classifier display documents in alphabetical order of title.

    Build the collection again and **preview** it.

Note how we assigned dc.Format metadata to all documents in the collection with a minimum of labour. We did this by capitalizing on the folder structure of the original information. Even though we complained earlier about how messy this folder structure is, you can still take advantage of it when assigning metadata.

Suppressing dummy text

10. Alongside the Audio files there is an MP3 icon, which plays the audio when you click it, and also a text document that contains some dummy text. Image files also have dummy

documents. These dummy documents aren't supposed to be seen, but to suppress them you have to fiddle with a format statement.

- Change to the **Format** panel and select the **Format Features** section.
- Ensure that **VList** is selected, and make the changes that are highlighted below. You need to insert five lines into the first line, and delete the second line. (Note, the changes are available in a text file, see below.) Change:

```
<td valign=top>[link][icon][/link]</td>
<td
valign=top>[ex.srclink][Or]{[ex.thumbicon],[ex.srcicon]}[ex./srcl
ink]</td>
<td valign=top>[highlight]
{Or}{[dls.Title],[dc.Title],[Title],Untitled}
[/highlight]{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
```

to this:

```
<td valign=top>
{If}{[dc.Format] eq 'Audio',
[srclink][srcicon][/srclink],
{If}{[dc.Format] eq 'Images',
[srclink][thumbicon][/srclink],
[link][icon][/link]}}</td>
<td valign=top>[highlight]
{Or}{[dls.Title],[dc.Title],[Title],Untitled}
[/highlight]{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
```

11. To make this easier for you we have prepared a plain text file that contains the new text. In WordPad open the following file:
12. *sample_files → beatles → format_tweaks → audio_tweak.txt*
13. (Be sure to use WordPad rather than Notepad, because Notepad does not display the line breaks correctly.) Place it in the copy buffer by highlighting the text in WordPad and selecting **Edit → Copy**. Now move back to the Librarian Interface, highlight all the text that makes up the current **VList** format statement, and use **Edit → Paste (ctrl-v)** to transform the old statement to the new one.
14. **Preview** the result. You may need to click the browser's **<Reload>** button to force it to re-load the page.

11. While we're at it, let's remove the source filename from where it appears after each document.
    - In the **VList** format feature, delete the text that is highlighted below:

```
<td valign=top>
{If}{[dc.Format] eq 'Audio',
[srclink][srcicon][/srclink],
{If}{[dc.Format] eq 'Images',
[srclink][thumbicon][/srclink],
[link][icon][/link]}}</td>
<td valign=top>[highlight]
{Or}{[dls.Title],[dc.Title],[Title],Untitled}
[/highlight]{If}{[ex.Source],<br><i>([ex.Source])</i>}</td>
```

12. **Preview** the result (you don't need to rebuild the collection.)

Using AZCompactList rather than AZList

12. There are sometimes several documents with the same title. For example, *All My Loving* appears both as lyrics and tablature (under *ALL MY LOVING*). The **Titles** browser might be improved by grouping these together under a bookshelf icon. This is a job for an **AZCompactList**.
    - Change to the **Design** panel and select the **Browsing Classifiers** section.
    - Remove the **ex.Title** classifier (at the top)
    - Add an **AZCompactList** classifier, and enter **dc.Title,ex.Title** as its metadata.
    - Finish by pressing **<OK>**.
    - Move the new classifier to the top of the list (**<Move Up>** button).

    Build the collection again and **preview** it. Both items for *All My Loving* now appear under the same bookshelf. However, many entries haven't been amalgamated because of non-uniform titles: for example *A Hard Day's Night* appears as four different variants. We will learn below how to amalgamate these.

Making bookshelves show how many items they contain

13. Make the bookshelves show how many documents they contain by inserting a line in the **VList** format statement in the **Format Features** section of the **Format** panel. The added line is shown highlighted below. The complete format statement can be copied from *sample_files → beatles → format_tweaks → show_num_docs.txt*.

```
<td valign=top>
{If}{[dc.Format] eq 'Audio',
[srclink][srcicon][/srclink],
{If}{[dc.Format] eq 'Images',
[srclink][thumbicon][/srclink],
[link][icon][/link]}}</td>
<td>{If}{[numleafdocs],([numleafdocs])}</td>
<td valign=top>[highlight]
{Or}{[dls.Title],[dc.Title],[Title],Untitled}
[/highlight]</td>
```

    Preview the result (you don't need to build the collection.) Bookshelves in the titles and browse classifiers should show how many documents they contain.

Adding a Phind phrase browser

14. In the **Browsing Classifiers** section on the **Design** panel, add a **Phind** classifier. Leave the settings at their defaults: this generates a phrase browsing classifier that sources its phrases from *Title* and *text*.

    Build the collection again and **preview** it. Select the new **Phrases** option from the navigation bar. Enter a single word in the text box, such as **band**. The phrase browser will present you with phrases found in the collection containing the search term. This can

provide a useful way of browsing a very large collection. Note that even though it is called a phrase browser, only single terms can be used as the starting point for browsing.

Branding the collection with an image

15. To complete the collection, lets give it a new image for the top left corner of the page. Go to the **General** section of the **Format** panel. Use the browse button of **URL to 'about page' image:** to select the following image:

    sample_files → beatles → advbeat_large → images → beatlesmm.png

    Preview the collection, and make sure the new image appears.

Using UnknownPlug

In this section we incorporate the MIDI files. Greenstone has no MIDI plugin (yet). But that doesn't mean you can't use MIDI files!

16. **UnknownPlug** is a useful generic plugin. It knows nothing about any given format but can be tailored to process particular document types—like MIDI—based on their filename extension, and set basic metadata.

    In the **Document Plugins** section of the **Design** panel:

    - add **UnknownPlug**;
    - activate its **process_extension** field and set it to "mid" to make it recognize files with extension *.mid*;
    - Set **file_format** to "MIDI" and **mime_type** to "audio/midi".

    In this collection, all MIDI files are contained in the file *beatles_midi.zip*. **ZIPPlug** (already in the list of default plugins) is used to unpack the files and pass them down the list of plugins until they reach **UnknownPlug**.

17. **Build** the collection and **preview** it. Unfortunately the MIDI files don't appear as Audio under the *browse* button. That's because they haven't been assigned **dc.Format** metadata.
    - Back in the **Enrich** panel, click on the file *beatles_midi.zip* and assign its **dc.Format** value to "Audio"—do this by clicking on "Audio" in the **Existing values for dc.Format** list. All files extracted from the Zip file inherit its settings.

Cleaning up a title browser using regular expressions

We now clean up the Titles browser.

*To do this we must put the Librarian Interface into a different mode. The interface supports four levels of user: **Library Assistant**, who can add documents and metadata to collections, and create new ones whose structure mirrors that of existing collections; **Librarian**, who can, in addition, design new collections, but cannot use specialist IT features (e.g. regular expressions); **Library Systems Specialist**, who can use all design features, but cannot perform troubleshooting*

tasks (e.g. interpreting debugging output from Perl programs); and ***Expert****, who can perform all functions.*

So far you have mostly been operating in ***Librarian*** *mode. We switch to* ***Library Systems Specialist*** *mode for the next exercise.*

18. To switch modes, click **File → Preferences... → Mode** and change to **Library Systems Specialist**. Note from the description that appears that you need to be able to formulate regular expressions to use this mode fully. That is what we do below.

19. Next we return to our **Titles** browser and clean it up. The aim is to amalgamate variants of titles by stripping away extraneous text. For example, we would like to treat "ANTHOLOGY 1", "ANTHOLOGY 2" and "ANTHOLOGY 3" the same for grouping purposes. To achieve this:
    - Go to the Title **AZCompactList** under **Browsing Classifiers** on the **Design** panel;
    - Activate **removesuffix** and set it to:

      ```
      (?i)(\\s+\\d+)|(\\s+[[:punct:]].*)
      ```

20. **Build** the collection and **preview** the result. Observe how many more times similar titles have been amalgamated under the same bookshelf. Test your understanding of regular expressions by trying to rationalize the amalgamations. (Note: *[[:punct:]]* stands for any punctuation character.) The icons beside the Word and PDF documents are not the correct ones, but that will be fixed in the next format statement.

The previous exercise was done in ***Library Systems Specialist*** *mode because it requires the use of regular expressions, something librarians are not normally trained in.*

One powerful use of regular expressions in the exercise was to clean up the ***Titles*** *browser. Perhaps the best way of doing this would be to have proper title metadata. The metadata extracted from HTML files is messy and inconsistent, and this was reflected in the original Titles browser. Defining proper title metadata would be simple but rather laborious. Instead, we have opted to use regular expressions in the* ***AZCompactList*** *classifier to clean up the title metadata. This is difficult to understand, and a bit fiddly to do, but if you can cope with its idiosyncrasies it provides a quick way to clean up the extracted metadata and avoid having to enter a large amount of metadata.*

Using non-standard macro files

To put finishing touches to our collection, we add some decorative features

20. Close the collection in the Librarian Interface (**File → Close**).

21. Using your Windows file browser outside Greenstone, locate the folder

    sample_files → beatles → advbeat_large

22. Open up another file browser, and locate the *small beatles* collection in your Greenstone installation:

    Greenstone → collect → smallbea

    smallbea is the folder name generated by Greenstone for this collection. You can determine what the folder name is for a collection by looking at the title bar of the Librarian Interface: the folder name is displayed in brackets after the collection name.

23. Using the file browser, copy the *images* and *macros* folders from the *advbeat_large* folder into the *smallbea* folder. (It's OK to overwrite the existing *images* folder: the image in it is included in the folder being copied.) The *images* folder includes some useful icons, and the *macros* folder defines some macro names that use these images.

    To see the macro definitions, open the collection in the Librarian Interface (**File → Open...**) and view the **Collection Specific Macros** section in the **Format** panel.

Using different icons for different media types

24. Re-edit your **VList** format statement to be the following (in **Format Features** on the **Format** panel). You can copy this text from the file *sample_files → beatles → format_tweaks → multi_icons.txt*.

```
<td valign=top>
  {If}{[numleafdocs],[link][icon][/link]}
  {If}{[dc.Format] eq 'Lyrics',[link]_iconlyrics_[/link]}
  {If}{[dc.Format] eq 'Discography',[link]_icondisc_[/link]}
  {If}{[dc.Format] eq 'Tablature',[link]_icontab_[/link]}
  {If}{[dc.Format] eq 'MARC',[link]_iconmarc_[/link]}
  {If}{[dc.Format] eq 'Images',[srclink][thumbicon]{/srclink]}
  {If}{[dc.Format] eq 'Supplementary',{srclink][srcicon]{/srclink]}
  {If}{[dc.Format] eq 'Audio',[srclink]{If}{[FileFormat] eq
'MIDI',_iconmidi_,_iconmp3_}[/srclink]}
</td>
<td>
{If}{[numleafdocs],{[numleafdocs]}}
</td>
<td valign=top>
[highlight]
{Or}{[dc.Title],[Title],Untitled}
{/highlight}
</td>
```

25. **Preview** your collection as before. Now different icons are used for discography, lyrics, tablature, and MARC metadata. Even MP3 and MIDI audio file types are distinguished. If you let the mouse hover over one of these images a "tool tip" appears explaining what file type the icon represents in the current interface language (note: *extra.dm* only defines English and French).

Changing the collection's background image

26. Go to the **Collection Specific Macros** section in the **Format** panel.

27. The content is fairly brief, specifying only what needs to be overridden from the default behaviour for this collection. Near the top you should see:

```
_collectionspecificstyle_ {
<style>
body.bgimage \{ background-image: url("_httpcimages_/beat_margin.gif");
\}
\#page \{ margin-left: 120px; \}
</style>
}
```

Replace the text **beat_margin.gif** with **tile.jpg**.

This line relates to the background image used. The new image *tile.jpg* was in the *images* folder that was copied across previously.

28. **Preview** the collection's home page. The page background is now the new graphic.

Other features can be altered by editing the macros—for example, the headers and footers used on each page, and the highlighting style used for search terms (specify a different colour, use bold etc.).

Building a full-size version of the collection

29. To finish, let's now build a larger version of the collection. To do this:
    - Close the current collection (**File → Close**).
    - Start a new collection called *large beatles* (**File → New...**).
    - Base this new collection on *small beatles*.
    - Copy the content of *sample_files → beatles → advbeat_large → import* into this newly formed collection. Since there are considerably more files in this set of documents the copy will take longer.
    - **Build** the collection and **preview** the result. (If you want the collection to have an icon, you will have to add it from the **Format** panel.)

Adding an image collage browser

30. Switch to the **Design** panel and select the **Browsing Classifiers** section. Pull down the **Select classifier to add:** menu and select **Collage**. Click **<Add Classifier...>**. There is no need to customize the options, so click **<OK>** at the bottom of the resulting popup.

31. Now change to the **Create** panel and **build** and **preview** the collection.

4.3. Scanned image collection

Here we build a small replica of Niupepa, the Maori Newspaper collection, using five newspapers taken from two newspaper series. It allows full text searching and browsing by title

and date. When a newspaper is viewed, a preview image and its corresponding plain text are presented side by side, with a "go to page" navigation feature at the top of the page.

The collection involves a mixture of plugins, classifiers, and format statements. The bulk of the work is done by ***PagedImgPlug****, a plugin designed precisely for the kind of data we have in this example. For each document, an "item" file is prepared that specifies a list of image files that constitute the document, tagged with their page number and (optionally) accompanied by a text file containing the machine-readable version of the image, which is used for full text searching. Three newspapers in our collection (all from the series "Te Whetu o Te Tau") have text representations, and two (from "Te Waka o Te Iwi") have images only. Item files can also specify metadata. In our example the newspaper series is recorded as* ***ex.Title*** *and its date of publication as* ***ex.Date****. Issue* ***ex.Volume*** *and* ***ex.Number*** *metadata is also recorded, where appropriate. This metadata is extracted as part of the building process.*

1. Start a new collection called **Paged Images** and fill out the fields with appropriate information: it is a collection sourced from an excerpt of Niupepa documents.

2. In the **Gather** panel, open the *sample_files* → *niupepa* → *sample_items* folder and drag the two subfolders into your collection on the right-hand side. A popup window asks whether you want to add **PagedImgPlug** to the collection: click **<Add Plugin>**, because this plugin will be needed to process the item files.

3. Some of the files you have just dragged in are the newspaper images; others are text files that contain the text extracted from these images. We want these to be processed by **PagedImgPlug**, not **ImagePlug** or **TEXTPlug**. Switch to the **Document Plugins** section of the **Design** panel and delete **ImagePlug** and **TEXTPlug**. While you are at it, you could tidy things up by deleting **ZIPPlug** and all plugins from **HTMLPlug** to **NULPlug** as well, since they will not be used. **GAPlug** and **PagedImgPlug** remain.

4. Open up the configuration window for **PagedImgPlug** by double-clicking on the plugin. Switch on its **screenview** configuration option by checking the box. The source images we use were scanned at high resolution and are large files for a browser to download. The **screenview** option generates smaller screen-resolution images of each page when the collection is built. Click **<OK>**.

5. In the **Search Indexes** section, check the **section** checkbox to build the indexes on section level as well as document level.

6. Now go to the **Create** panel, **build** the collection and **preview** the result. Search for "waka" and view one of the titles listed (all three appear as *Te Whetu o Te Tau*). Browse by **Titles** and view one of the *Te Waka o Te Iwi* newspapers. Note that only the *Te Whetu o Te Tau* newspapers have text; *Te Waka o Te Iwi* papers don't.

This collection was built with Greenstone's default settings. You can locate items of interest, but the information is less clearly and attractively presented than in the full Niupepa collection.

Grouping documents by series title and displaying dates within each group

Under Titles documents from the same series are repeated without any distinguishing features such as date, volume or number. It would be better to group them by series title and display other information within each group. This can be accomplished using an AZCompactList classifier rather than AZList, and tuning the classifier's format statement.

7. In the **Design** panel, under the **Browsing Classifiers** section, delete the **AZList** classifiers for **ex.Source** and **ex.Title**.

8. Now add an **AZCompactList** classifier, setting its **metadata** option to **ex.Title**, and add a **DateList** classifier, setting its **metadata** option to **ex.Date**.

9. In the **Format Features** section of the **Format** panel, select the **ex.Title** classifier in the **Choose Feature** list, and **VList** in the **Affected Component** list. Click **<Add Format>** to add this format statment to your collection. Delete the contents of the **HTML Format String** box, and add the following text. (This format statement can be copied and pasted from the file *sample_files → niupepa → formats → titles_tweak.txt*.)

```
<td valign="top">[link][icon][/link]</td>
<td valign="top">
{If}{[numleafdocs],[ex.Title] ([numleafdocs]),
{If}{[ex.Volume],Volume [ex.Volume] }
{If}{[ex.Number],Number [ex.Number] }
{If}{[ex.Date], [ex.Date]}}
</td>
```

10. **Build** the collection, and **preview** the new *Titles* list.

    As a consequence of using the **AZCompactList** classifier, bookshelf icons appear when titles are browsed. This revised format statement has the effect of specifying in brackets how many items are contained within a bookshelf. It works by exploiting the fact that only bookshelf icons define `[numleafdocs]` metadata. For document nodes, Title is not displayed. Instead, Volume, Number and Date information are displayed if present.

Displaying scanned images and suppressing dummy text

*When you reach a newspaper, only its associated text is displayed. When either of the Te Waka o Te Iwi newspapers is accessed, the document view presents the message "This document has no text.". No scanned image information (screen-view resolution or otherwise) is shown, even though it has been computed and stored with the document. This can be fixed by a format statement that modifies the default behaviour for **DocumentText**.*

11. In the **Format Features** section of the **Format** panel, select the **DocumentText** format statement. The default format string displays the document's plain text, which, if there is none, is set to "This document has no text.". Change this to the following text. (This format statement can be copied and pasted from the file *sample_files → niupepa → formats → doc_tweak.txt*)

```
<table><tr>
<td valign=top>[srclink][screenicon][/srclink]</td>
```

```
<td valign=top>[Text]</td>
</tr></table>
```

*Including `[screenicon]` has the effect of embedding the screen-sized image generated by switching the **screenview** option on in **PagedImgPlug**. It is hyperlinked to the original image by the construct `[srclink]...[/srclink]`.*

This modification will display screenview image, but does nothing about the dummy text "This document has no text.", which will still be displayed. To get rid of this, edit the **DocumentText** format statement again and replace

```
<td valign=top>[Text]</td>
```

with

```
{If}{[Text] ne "This document has no text. ",<td
valign=top>[Text]</td>}
```

12. **Preview** the collection and view one of the **Te Waka o Te Iwi** documents. The line "This document has nd text." should now be gone. (Note that it important to get the text exactly right for this to work, including the space after the ".".)

Searching at page level

13. The newspaper documents are split into sections, one per page. For large documents, it is useful to be able to search on sections rather than documents. This allows users to more easily locate the relevant information in the document.

14. Go to the **Search Indexes** section of the **Design** panel. Remove the **ex.Source** index. Check the **section** checkbox to build the indexes on section level as well as document level. Make section level the default by selecting its **Default** radio button.

15. Set the display text used for the level drop-down menu by going to the **Search** section on the **Format** panel. Set the document level text to "newspaper", and the section level text to "page".

16. **Build** and **preview** the collection. Compare searching at "newspaper" level compared to "page" level. A useful search term for this collection is "aroha".

17. You will notice that when searching for individual pages, the newspaper image is displayed in the search results. As these images are very large, this is not very useful. Go to **Format Features** section of the **Format** panel in the Librarian Interface and select the **VList** format statement from the list of assigned format statements. Remove the second line from the **HTML Format String**:

```
<td
valign="top">[ex.srclink]{Or}{[ex.thumbicon],[ex.srcicon]}[ex./srclink]
</td>
```

While we are here, lets remove the filename from the display. Remove the following from the last line:

```
{If}{[ex.Source],<br><i>([ex.Source])</i>}
```

Preview the collection—the search results should be back to normal.

18. Now you will notice that page level search results only show the Title of the page (the page number), and not the Title of the newspaper. We'll modify the format statement to show the newspaper title as well as the page number. Also, lets add in Volume and Number information too.

    In the **Format Features** section, select **Search** in **Choose Feature**, and **VList** in **Affected Component**. Click **<Add Format>** to add this format to the collection. The previous changes modified **VList**, so they will apply to all **VLists** that don't have specific format statements. These next changes are made to **SearchVList** so will only apply to search results.

    The extracted Title for the current section is specified as `[ex.Title]` while the Title for the parent section is `[parent:ex.Title]`. Since the same **SearchVList** format statement is used when searching both whole newspapers and newspaper pages, we need to make sure it works in both cases.

    Set the format statement to the following text (it can be copied and pasted from the file *sample_files → niupepa → formats → search_tweak.txt*):

```
<td valign="top">[link][icon][/link]</td>
<td valign="top">
{If}{[parent:ex.Title],[parent:ex.Title]
{If}{[parent:ex.Volume],Volume [parent:ex.Volume] }
{If}{[parent:ex.Number],Number [parent:ex.Number]}: Page [ex.Title],
[ex.Title] {If}{[ex.Volume], Volume [ex.Volume] }
{If}{[ex.Number], Number [ex.Number] }}
<br/><i>({Or}{[parent:ex.Date],[ex.Date]})</i></td>
</td>
```

    Preview the search results. Items display newspaper title, Volume, Number and Date if available, and pages also display the page number.

In the collection you have just built, newspapers are grouped by series title, and dates are supplied alongside each one to distinguish it from others in the same series. Users can browse chronologically by date, and when a newspaper page is viewed a preview image is shown on the left that displays the original high-resolution version when clicked, accompanied on the right by the plain-text version of that newspaper (if available).

While we are here, lets remove the filename from the display. Remove the following from the last line:

```
{If}{{ex.Source},<br><i>({ex.Source})</i>}
```

Preview the collection—the search results should be back to normal.

18. Now you will notice that page level search results only show the Title of the page (the page number), and not the Title of the newspaper. We'll modify the format statement to show the newspaper title as well as the page number. Also, lets add in Volume and Number information too.

    In the **Format Features** section, select **Search** in **Choose Feature**, and **VList** in **Affected Component**. Click **<Add Format>** to add this format to the collection. The previous changes modified **VList**, so they will apply to all **VLists** that don't have specific format statements. These next changes are made to **SearchVList** so will only apply to search results.

    The extracted Title for the current section is specified as `[ex.Title]` while the Title for the parent section is `[parent:ex.Title]`. Since the same **SearchVList** format statement is used when searching both whole newspapers and newspaper pages, we need to make sure it works in both cases.

    Set the format statement to the following text (it can be copied and pasted from the file *sample_files → niupepa → formats → search_tweak.txt*):

```
<td valign="top">{link}{icon}{/link}</td>
<td valign="top">
{If}{{parent:ex.Title},{parent:ex.Title}
{If}{{parent:ex.Volume},Volume {parent:ex.Volume} }
{If}{{parent:ex.Number},Number {parent:ex.Number}}: Page {ex.Title},
{ex.Title} {If}{{ex.Volume}, Volume {ex.Volume} }
{If}{{ex.Number}, Number {ex.Number} }}
<br/><i>({Or}{{parent:ex.Date},{ex.Date}})</i></td>
</td>
```

    Preview the search results. Items display newspaper title, Volume, Number and Date if available, and pages also display the page number.

In the collection you have just built, newspapers are grouped by series title, and dates are supplied alongside each one to distinguish it from others in the same series. Users can browse chronologically by date, and when a newspaper page is viewed a preview image is shown on the left that displays the original high-resolution version when clicked, accompanied on the right by the plain-text version of that newspaper (if available).

LAB 5:

MARC and CDS/ISIS to Greenstone

5.1. Bibliographic collection—Part A

This exercise looks at using fielded searching in a collection. Fielded searching is best used for metadata rich collections. Here we use bibliographic data in MARC format. We also "explode" the database, enabling editing of the metadata with the Librarian Interface.

1. Start a new collection called **Beatles Bibliography** which will contain a collection of MARC records on the Beatles, from the US Library of Congress. Enter the requested information and base it on -- **New Collection** --.

2. In the **Gather** panel, open the *sample_files* → *marc* folder, drag *locbeatles50.marc* into the right-hand pane and drop it there. A popup window asks whether you want to add **MARCPlug** to the collection to process this file. Click **<Add Plugin>**, because this plugin will be needed to process the MARC records.

3. In the **Document Plugins** section of the **Design** panel, remove the plugins **TextPlug** to **NULPlug** by selecting each one in the **Assigned Plugins** list and clicking **<Remove Plugin>** (**ZIPPlug**, **GAPlug** and **MARCPlug** remain). It is not strictly necessary to remove these redundant plugins, but it is good practice to include only plugins that are needed, to avoid unwanted (and unexpected) side effects.

4. Now select **Browsing Classifiers** within the **Design** panel and **remove** the default classifier for **Source** metadata.

5. In the **Search Indexes** section, **remove** the **ex.Source** index. In this collection all records are from the same file, so **ex.Source** metadata, which is set to the filename, is not particularly interesting or useful.

6. Switch to the **Create** panel, **build** the collection, and **preview** it. Browse through the *Titles* and view a record or two. Try searching—for example, find items that include **rock music**.

7. Back in the Librarian Interface, go to the **Browsing Classifiers** section of the **Design** panel. Select **AZCompactList** from the **Select classifier to add:** drop down menu, and click **<Add Classifier...>**. In the popup window, select **ex.Subject** as the metadata item. Click **<OK>**.

   AZCompactList is like AZList, except that terms that appear multiple times in the hierarchy are automatically grouped together and a new node, shown as a bookshelf icon, is formed.

8. **Build** the collection and **preview** the result.

Using fielded searching

9. Collections built with MGPP (the default indexer) provide the option of fielded searching. In the browser, go to the *PREFERENCES* page. You will notice that there is a **Query style:** option which enables you to switch between "normal" and "fielded" search. Change to fielded search now and click on the **Search** button. The search form has changed to a fielded form.

10. You can specify which search form types are available for a particular collection, and which one is the default, using the **SearchTypes** format statement. In the **Format** panel select **Format Features** from the left-hand list. Select the **SearchTypes** format statement from the list of assigned formats, and set the contents to **form**. This will make only fielded searching available for this collection.

    *Search type options include **form** and **plain**. You can specify one or both separated by a comma. If both are specified, the first one is used as the default: this is the one that the user will see when they first enter the collection.*

11. **Preview** the collection again. Notice that the collection's home page no longer includes a query box. (This is because the search form is too big to fit here nicely.) To search, you have to click **Search** in the navigation bar. Note that the *PREFERENCES* page has changed so that the "normal" query style is no longer offered.

12. Look at the search form in the collection. There are two fields that can be searched: *text* and *Title*. Add some more fields to search on by going back to the Librarian Interface.

13. In the **Design** panel, go to the **Search Indexes** section. Add a new index based on **ex.Subject** by clicking **<New Index>**, selecting **ex.Subject** in the list of metadata, and clicking **<Add Index>**.

14. **Rebuild** the collection and **preview** the results. Notice the extra field in the **... in field** drop-down menus in the search form. You can do quite complicated queries by searching for words in different fields at the same time.

15. To change the text that is displayed in the drop-down menus of the search form, go to the **Search** section of the **Format** panel. Here you can change the display text for the indexes.

5.2. Bibliographic collection—Part B

Exploding the database

16. Go to the **Enrich** panel and try to see the metadata. It doesn't appear! This is because the metadata is associated with records inside the file, not the file itself.

Metadata file types, such as MARC, CDS/ISIS, BibTex etc. can be imported into Greenstone but their metadata cannot be viewed in the Librarian Interface. To edit any metadata you need to go back to the program that created the file.

Greenstone provides a way of *exploding* a metadata database so that each record appears as an individual document, with viewable and editable metadata. This process is irreversible: once this step has been done, the database is deleted and can no longer be used in its original program.

17. In the **Gather** panel, you may notice that the MARC database has a different coloured icon to other files. This green icon indicates that a file is a metadata database that can be exploded. Right-click on the file and choose **Explode Metadata Database** from the menu. A new window opens, containing options for the exploding process. A description of each option can be obtained by hovering the mouse over the option.

    Turn on the **metadata_set** option by checking its box. This option indicates which metadata set to explode the metadata into. The default set is the "Exploded Metadata Set"—a metadata set which initially has no elements in it, but will receive a new element for each metadata field retrieved from the database.

18. Click **<Explode>** to start the exploding process. This may take a short while, depending on the size of the database.

19. Once exploding has finished, the MARC database file will have been deleted, and a folder created in its place. This folder contains an empty file for each record in the original database. The metadata for these records can be viewed and edited by switching to the **Enrich** panel.

20. Because the MARC file is no longer present, and the collection contains empty (.nul) files, we need to change the list of plugins. In the **Document Plugins** section of the **Design** panel, remove **MARCPlug** and add **NULPlug** (use the default configuration).

21. **Rebuild** and **preview** the collection. You will notice that the *Titles* classifier displays the filename not the record title, the *Subjects* classifier is empty, searching no longer returns any results, and the document display is useless.

Reformatting the collection to use the exploded metadata

The collection previously used extracted (ex.) metadata, but now it uses exploded (exp.) metadata. The classifiers and search indexes were built on ex metadata, which is why they no longer work properly.
There is also no longer any text in the documents. Previously, MARCPlug stored the raw record as the "text" of each record. Now that the metadata is in the Librarian Interface, there is no longer the concept of raw record, and so there is no text.
We need to modify the collection design to take note of these changes.

22. In the **Search Indexes** section, change the Title index to use **exp.Title**: select the Title index in the **Assigned Indexes** list and click **<Edit Index>**. Deselect **ex.Title** in the list of metadata, and select **exp.Title**. Click **<Replace Index>**.

23. Remove the **ex.Subject** index by selecting it in the **Assigned Indexes** list and clicking **<Remove Index>**. Add an index on **exp.Subject**: click **<New Index>**, select **exp.Subject** in the metadata list, and click **<Add Index>**.

24. The text index is no longer any use, so remove that index too.

25. To enable combined searching across all indexes at once, click **<New Index>**, tick the **Add combined searching over all assigned indexes (allfields)** checkbox, and click **<Add Index>**. Move this to the top of the list using the **<Move Up>** button, so that it appears first in the drop down list. Click **<Set Default Index>** so that it becomes the default field for searching.

26. In the **Browsing Classifiers** section, change the **ex.Title AZList** to use **exp.Title** metadata. Double click the **ex.Title AZList** in the **Assigned Classifiers** list, and change the **metadata** option to use **exp.Title**. Click **<OK>**. Do the same thing for the Subject **AZCompactList**, changing **ex.Subject** to **exp.Subject**.

27. In the **Format Features** section of the **Format** panel, select **VList** in the list of assigned format statements.
    - There is no dls or dc metadata for this collection, so replace `{Or}{[dls.Title],[dc.Title],[ex.Title],Untitled}` with `{Or}{[exp.Title],[ex.Title],Untitled}`.
    - There are no source or thumb icons, so remove the second line: `<td valign="top">[ex.srclink]{Or}{[ex.thumbicon],[ex.srcicon]}[ex./srclink]</td>`.
    - The ex.Source metadata is set to the nul filename, so remove that from the display: remove `{If}{[ex.Source],<br><i>([ex.Source])</i>}`

    The resulting format statement looks like:

```
<td valign="top">[link][icon][/link]</td>
<td valign="top">[highlight]
{Or}{[exp.Title],[ex.Title],Untitled}
[/highlight]</td>
```

28. Clear the **DocumentHeading** format statement by selecting it in the list of assigned format statements and deleting the contents in the **HTML Format String**. The record Title will be displayed as part of the **DocumentText** format, so we don't need it here.

29. Next, edit the **DocumentText** format statement. Delete the contents and replace it with

```
<table>
<tr><td>Title:</td><td>[exp.Title]</td></tr>
<tr><td>Subject:</td><td>[exp.Subject]</td></tr>
```

```
<tr><td>Publisher:</td><td>[exp.Publisher]</td></tr>
</table>
```

30. The *DETACH* and *NO HIGHLIGHTING* buttons are not very useful for this collection, so lets get rid of them. Edit the **DocumentButtons** format statement to make it empty.

31. **Rebuild** and **preview** the collection. The classifiers should be back to normal, searching should now work, and there should be a nice record display.

5.3. CDS/ISIS collection

*This exercise is similar to the **Bibliographic collection** exercise, except that a CDS/ISIS database is used instead of a MARC database, and we do not explode the database.*

1. Start a new collection called **ISIS Collection**.

2. Drag the files from *sample_files* → *isis* (excluding the *format_tweaks* folder) into the collection.

3. **Build** and **preview** the collection. The default indexes, classifiers and formats are not very useful for this data. There is no metadata searching, and the *Titles* classifier is completely empty. The filenames classifier is useless because all records come from the same file.

4. In the **Search Indexes** section of the **Design** panel, remove the useless Source and Title indexes, and add new indexes for Photographer^all, Country^all and Notes^all metadata.

   CDS/ISIS metadata has subfields, and these are represented using ^.

5. In the **Browsing Classifiers** section, remove the existing (useless) classifiers for **Title** and **Source**, and add a new **AZList** for **Photographer**.

6. **Rebuild** and **preview** the collection.

7. In the **Format Features** section of the **Format** panel, change the **VList** format statement to display **Photographer** and **Notes** metadata. Change it to look like:

   ```
   <td valign=top>[link][icon][/link]</td>
   <td valign=top><b>[ex.Photographer^all]</b><br/>[ex.Notes^all]</td>
   ```

8. Make fielded searching the default by changing the **SearchTypes** format statement to **form,plain** (instead of **plain,form**).

ISISPlug stores a nicely formatted version of the record as the document text, and this is what is displayed when we view a record. Lets tidy it up a little more.

9. Remove the *DETACH* and *NO HIGHLIGHTING* buttons by setting the **DocumentButtons** format statement to empty.

10. Remove the "Untitled" at the top of the document by setting the **DocumentHeading** format statement to empty.

11. Finally, lets link to the raw record, which is stored as **ISISRawRecord** metadata. Edit the **DocumentText** format statement to look like the following. (This format can be copied from *sample_files → isis → format_tweaks → document_text.txt*.)

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

Preview the collection.

LAB 6:

Greenstone: Customization & Interoperability

6.1. 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 `_macroname_ {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;
     color: #000000;
     }
     ```

     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 text "`_pagefooterextra_`" in the `_footer_` macro and give the following:

    ```
    {<center><small><font color='navy'>Copyright 2007 My Digital
     Library</font></small></center>}
    ```

    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.

12. 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.

13. 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.

14. 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.

15. Preview the newly structured home page in a web browser. (On Windows, restart the Greenstone library server.)

16. 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* in 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 is the only file that mentions this image.

Close the command prompt.

6.2. 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.

6.3. 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 can 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 **oaiservi**.

8. In a text editor (e.g. WordPad), open the collection's configuration file, which is in *Greenstone → collect → oaiservi → 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.

6.4. 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],([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.

6.5. 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.

