

## Lab 5 (2 hours): Advanced formatting; creating a CD-ROM

### Part I – Macro files: Customizing the library

---

*In this part of the lab we use the Windows file manager (double click on **My Computer** to initiate) to move, copy and rename files. We also use **WordPad** to edit text-based configuration files, though other applications such as **NotePad** are equally suitable.*

The appearance of all pages produced by Greenstone is governed by macro files, which reside in the folder **C:\Program Files\Greenstone\macros**. In this part of the lab, we customize the macros to change the appearance of our library.

#### *Changing the background and header images*

1. Using the Windows file manager, open:

My Computer → GSDL Fiji 05 → workshop\_files

If the CD-ROM is self-installing, you need to right click on the CD-ROM icon in My Computer, and choose “Explore” or “Open” to open it.

Open another window that shows the files in:

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

Drag *usp\_background.gif* and *usp\_header.gif* from the *workshop\_files* folder and drop them into *images*. These are the two images that we will use as the home page header image and background image.

[Note: The Windows operating system normally suppresses the content of the *Program Files* folder. The first time you enter this folder a system alert may ask whether you wish to proceed.]

2. Edit the relevant macro files. Use the file manager to open the folder:

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

3. **Open** the file *base.dm* with **WordPad**. In WordPad, locate this part of the *base.dm* file:

```
_httpiconchalk_ {_httpimg_/chalk.gif}  
_widthchalk_ {2000}  
_heightchalk_ {10}.
```

It's about 2/3rds of the way through. (If it helps, use **Edit→Find** to search for 'chalk'.) Use copy and paste on these three lines to make this part of the file look like the following:

```
# Original statements
#_httpiconchalk_ {_httpimg_/chalk.gif}
#_widthchalk_ {2000}
#_heightchalk_ {10}

_httpiconchalk_ {_httpimg_/usp_background.gif}
_widthchalk_ {1600}
_heightchalk_ {70}
```

A hash (#) at the start of line signals a comment, and Greenstone will ignore this line. We use this to ‘comment out’ the original three statements and replace them with modified lines. It is useful to retain the original version in case we need to restore the original file at a later date. These three lines relate to the background image used.

4. Save *base.dm* (File→Save or Ctrl-S) and close WordPad.
5. **Open** the file *home.dm* with **WordPad**. Use **Edit→Find** to find each occurrence of *gsdlhead.gif* in this file (there are three) and replace each one with *usp\_header.gif*.
6. Save *home.dm* and close WordPad.
7. Quit Greenstone; then re-launch it. Preview the home page. The page header and background should now use the new graphics.

### *Adding a footer*

Greenstone uses a style system to keep the format of each page the same. *Style.dm* defines a header and footer for each page, and macro files for the different pages define the page content.

8. Next we add a footer to each page. **Open** the file *style.dm* with **WordPad**.
9. Locate the `_footer_` macro:

```
_footer_ {
<!-- page footer (\_style:footer\_) -->
_pagefooterextra_
_endspacer__htmlfooter_
}
```

10. After `_pagefooterextra_` add some text. For example, “Produced at the Greenstone workshop, University of the South Pacific, Fiji, 2005”. The resulting macro will look something like:

```
_footer_ {
<!-- page footer (\_style:footer\_) -->
_pagefooterextra_
Produced at the Greenstone workshop, University of the South
Pacific, Fiji, 2005
_endspacer__htmlfooter_
}
```

11. Save *style.dm* and close WordPad. Quit Greenstone; then re-launch it. Each page should now have the new text at the bottom.
12. Adding text into the footer adds it for all pages. To add a specific footer for a particular page, use `_pagefooterextra_`. For example, let's add another piece of text just to the home page footer.
13. **Open** the file *home.dm* with **WordPad**. Add a new macro to the file. The first line in the file is "package home". After this line, add the following text:

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

14. Save *home.dm* and close WordPad. Quit Greenstone; then re-launch it. The home page will display the new text, while the other pages won't.

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

*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.

15. Use the file manager to open the Greenstone *etc* folder:

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

16. **Open** the file *main.cfg* with **WordPad**.

17. Locate this part of the *main.cfg* file:

```
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 \
               english.dm english2.dm \
               french.dm french2.dm \
               spanish.dm spanish2.dm \
               russian.dm russian2.dm \
               home.dm collect.dm docs.dm bsummary.dm \
               translang.dm garish.dm
```

It's about 1/3rd of the way through. (If it helps, use the Edit→Find feature to search for something like 'macrofiles'.) Change the text **home.dm** to **yourhome.dm**. Save the file and quit WordPad.

18. If you have Greenstone or the Librarian Interface running, quit both.
19. Now launch the Greenstone Digital Library afresh and view the newly structured home page for the collection.

20. Now reverse the changes you have made. Restore the default home page by changing *yourhome.dm* in *etc* → *main.cfg* back to *home.dm*, delete the three lines added in *macros* → *base.dm*, and restore the three lines you commented out.

## Part II – Macro files: Customizing a collection

---

Macros can also be used to customize single collections. There are two ways to add collection specific macros: Add the macros to the general macro files, but include [c=collname] between the macro name and its definition. Alternatively, add the macro files to an extra.dm file stored in the collection.

1. Using the Windows File Manager, copy the file **C:\Program Files\Greenstone\macros\extra.dm** to the **C:\Program Files\Greenstone\collect\reports\macros** folder (you may need to create the macros folder if it doesn't already exist).
2. Open this new file in **WordPad**, and add the following text:

```
package Global

_httpiconchalk_ {_httpimg_/usp_background.gif}
_widthchalk_ {1600}
_heightchalk_ {70}
```

3. Preview the collection. The reports collection will now have the alternative background while the other collections keep the default.
4. Next add a collection-specific macro into one of the main macro files. Open **C:\Program Files\Greenstone\macros\about.dm** in WordPad. Find the following text:

```
_textabout_ {
<h3>_textabcol_</h3>
_Global:collectionextra_
}
```

This macro displays the “About this collection” title, and the collection description underneath the navigation bar on the about page.

5. Use copy and paste on these four lines to make it look like:

```
_textabout_ {
<h3>_textabcol_</h3>
_Global:collectionextra_
}
_textabout_ [c=reports]{
<h3>Reports collection created in Fiji</h3>
_Global:collectionextra_
}
```

**Save** and close the file.

6. **Preview** the reports collection. The about page will have a new title underneath the search form.

Note `_textabcol_` is one of Greenstone's many text macros. These contain fragments of text that need to be translated for interfaces in other languages. They are defined in macro files **english.dm**, **french.dm** and so on. If you open up **english.dm** in **WordPad**, you can see most of the text fragments that are used in the interface.

### **Part III – Creating new Greenstone style images for the library**

---

#### *Creating a new button for the navigation bar*

1. In your digital library, preview the **tudorX** collection. Remember that we created a new browsing structure based on Category metadata, for which there were no nice button images. The link to the classifier therefore appears as text. In this exercise, we create the new images for the navigation bar.
2. In a web browser, go to <http://www.greenstone.org/make-images.html>
3. Fill in the fields in the **Greenstone classifier image generation** section. Keep the language at English, and use the following for the values: "Category", "category", "categories". Click **<Go>**.
4. The resulting page gives you some images and macros. The images should be downloaded to Greenstone's images directory (Right click, then choose "Save image as..." from the menu, navigate to **C:\Program Files\Greenstone\images**, and click **<Save>**.
5. The macro definitions need to be saved to macro file **extra.dm**. Open up **C:\Program Files\Greenstone\macros\extra.dm** in WordPad. Select the macros text on the web page with the mouse and copy it (Edit->Copy, or Ctrl-C). Paste it into the **extra.dm** file (Edit->Paste, or Ctrl-V).
6. Restart the library, and open up the tudorX collection. The new image will appear in the navigation bar.

#### *Creating a new collection image*

7. You can also make a Greenstone style image for a collection. Fill in the fields in the **Greenstone collection image generation** section. For example, for the TudorX collection, use "tudorx" for the collection internal name, and "Tudor\ncollection" for the icon text. (The \n will split the text into two lines.) Click **<Go>**. Save the two images to the images folder in the tudorX collection: **C:\Program Files\Greenstone\collect\tudorx\images**.
8. Open up the tudorX collection in GLI, and switch to the **Design** panel. Use the **Browse** buttons to select these images for the **URL to 'about page' icon** (tudorx.gif) and **URL to 'home page' icon** (tudorx\_sm.gif).
9. **Preview** the collection—you will see the new collection icon.

## Part IV – Format statements: Customizing a collection

---

In this section we look at customizing some of the collections using format statements. You have already done some basic modifications to format statements in the previous labs. Here you will do more extensive formatting.

*Customizing the Word and PDF collection:*

1. Open the **reports** collection in the Librarian Interface and go to the **Format Features** section of the **Design** panel.
2. 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.

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]</td>
<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 search results, classifiers, and document table of contents. First you will tidy this up a bit.

{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].

There is no *dls.Title* metadata, so remove that element from {Or} { [dls.Title] , [dc.Title] , [ex.Title] , Untitled }.

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.

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. Experiment with removing either of the two links from the format statement. 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.

4. Next, we'll customize the format for the *authors a-z* list. Classifier nodes have only a few pieces of metadata to display: `[ex.Title]` and `[numleafdocs]`. Whatever metadata the classifier has been built on, the node label is always stored as `[ex.Title]`. This is why a Creator is printed out for each bookshelf node even though *dc.Creator* is not specified in the format statement. `[numleafdocs]` is only defined for bookshelf nodes, so this metadata can be used in an `{If}` statement to make bookshelf nodes and document nodes display differently.
5. Make each bookshelf node in the **Creator** classifier show how many entries it contains. In the **Format Features** section of the **Design** panel, select the **Creator** classifier from the **Choose Feature** drop down list, and **VList** from the **Affected Component** list. Append the following:

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

Click **<Add Format>**, switch to the **Create** panel, and click **<Preview>** (no need to rebuild).

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

6. Modify the document nodes in the **Creator** classifier to display all authors. After `{If}{[ex.Source],<br>` in the format statement, add `[sibling:ex.Creator]`. `[ex.Source]` is not defined for bookshelf nodes, 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:ex.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 bookshelf nodes, it will also display how many documents the bookshelf contains.

For document nodes, it will display all the Authors (Creators), and the source document. `[sibling:ex.Creator]` displays all the Creator metadata for the document, separated by “,”. **Preview** the *authors a-z* list.

Change the separator between the authors. Modify the format statement, and replace `[sibling:ex.Creator]` with `[sibling(All'<br/>'):ex.Creator]`. This will add a new line after each author. **Preview** the *authors a-z* list.

### *Customizing the Beatles Music collection*

7. Open the **Beatles Music** collection in the Librarian Interface, and **preview** it.
8. In the browsing lists, each file has two icons: one linking to the Greenstone document, the other to the file itself. None of the files in this collection have any extractable text, and so the link to the Greenstone document is useless.
9. Images and MP3 files have nice icons – either a thumbnail of the image, or an MP3 icon. MIDI file icons currently use the unknown icon. We will change that to a MIDI icon.

Modify the **VList** format statement in the **Design** panel (**Format Features** section) so that each document (and bookshelf) has only one link. Change the first two lines to the following:

```
<td valign=top>
  {If}{[numleafdocs],[link][icon][link]}
  {If}{[ex.FileFormat] eq MIDI,[srclink]_iconmidi_[/srclink]}
  {If}{[ex.FileFormat] eq MP3,[srclink][srcicon][/srclink]}
  {If}{[ex.FileFormat] eq JPEG,[srclink][thumbicon][/srclink]}
</td>
```

`_iconmidi_` is a macro representing a MIDI icon.

**Preview** the collection.

### *Customizing the demo collection*

Cover images can be supplied with documents in a collection. They need to be jpeg files (with extension *.jpg*), and have the same name as the document. For example, the cover image for a file called *greenstone.doc* must be called *greenstone.jpg*. If cover images are present, the build process will associate them with the documents, and they can be used for document display. The demo collection is one collection that has cover images.

10. In a browser, look at the documents in the demo collection. Notice that they have an image at the top left of the document, underneath the navigation bar.
11. Open the demo collection in the Librarian Interface, and go to **Format Features** in the **Design** panel.



12. Scroll down the list of Currently Assigned Format Commands, and select the **DocumentImages** format statement. This is an on/off type format command. Switch it off by unchecking the **Enabled** checkbox, and click **<Replace Format>**. **Preview** the collection—the cover images are no longer displayed.
13. Cover images can be used in a format statement by adding `[DocImage]`. Modify the **SearchVList** format statement, replacing `[icon]` with `<img src=' [DocImage]' height='40' />`. **Preview** the collection, and do a search. The cover images should show up in the search results.
14. Reverse the changes you have made to the demo collection format statements.

## **Part V – Exporting to CD-ROM**

---

We now create a CD-ROM of your custom collection for you to take away. In this exercise, you should go through the process of creating the contents for the CD-ROM, but don't write it to disk. On the last day, once you have finished all the labs, you can do this again and create a CD-ROM to take home with you.

1. First decide which collections you want to **export** to CD-ROM. You should choose the collection of your own documents, but you might also want to include some of the collections you have worked on in the labs.
2. Launch the Greenstone Librarian Interface (GLI) if it is not already running.
3. Choose **File→Write CD/DVD Image**. In the resulting popup window, enter a name for your CD. This is the name that will appear in the menu when the CD is run. Next, select the collections you wish to export, then click **<Write CD/DVD image>**.

The necessary files for export are written to:

My Computer → Local Disk (C:) → Program  
Files→Greenstone→tmp→exported\_xxx

where xxx will be similar to the name you have entered.

4. Using the Windows CD-ROM writing facility, copy the contents of the `exported_xxx` folder into the CD-ROM and write it to a disk.

## **Part VI – Your custom collection**

---

1. Brand your Greenstone Digital Library by designing your own background image that incorporates your institution's logo. You may need to use an image editor such as PhotoShop to compose the necessary graphic.
2. Enhance the look of your collection by playing with the format statements.

## **Part VII – Extra work**

---

1. In Part I of this lab, 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.
2. To find out the file 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 popup and will display the image name: gsdlhead.gif. Click Cancel to close the dialog—you don’t need to save the image. Do the same for the background image by right clicking on the left hand green squirly bar. This time choose “Save background as” to find out the name: chalk.gif, then click Cancel.

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.

3. 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 up a **command prompt** (**Start→Run**, and enter “cmd” as the name of the program to run, or **Start→Programs→Accessories→Command Prompt**).

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 the macro files will be displayed, along with any matches. You will see that *home.dm* and *exported\_home.dm* both contain gsdlhead.gif. *home.dm* is the one you want to edit—*exported\_home.dm* is used for the home page when you export a collection to CD-ROM.

Do the same thing for chalk.gif:  
*find “chalk.gif” “C:\Program Files\Greenstone\macros\\*.dm”*

*base.dm* is the only file that mentions this image.

Close the command prompt.

## Appendix: Format statement reference

---

### Format types:

Item	Description
DocumentImages true/false	If true, display a cover image at the top left of the document page Default: false
DocumentTitles true/false	If DocumentImages is false, and this is true, use DocumentHeading to display the title. Default: true
DocumentHeading formatstring	This is used for a document heading at the top left if DocumentImages is false and DocumentTitles is true. Default: {Or} {[parent(Top):Title],[Title],untitled}  [Title]
DocumentContents true/false	Display table of contents (if document is hierarchical), or next/previous section arrows and "page k of n" text (if document is paged) Default: true
DocumentButtons string	Controls the buttons that are displayed on a document page. Valid options are Detach, Highlight, Expand Text, Expand Contents. Should be separated by  . Default: "Detach Highlight"
DocumentText formatstring	Format of the text to be displayed on a document page Default: <center> <table width=537> <tr><td>[Text]</td></tr> </table> </center>
DocumentArrowsTop true/false	Display next/previous section arrows at top of document, underneath the navigation bar, on document page Default: false
DocumentArrowsBottom true/false	Display next/previous section arrows at bottom of document page Default: true
DocumentUseHTML true/false	If true, each document is displayed in a separate frame. The Preferences page will also change slightly, adding options applicable to a collection of HTML documents. Default: false
NavigationBar pulldown	If set, provides a drop down list in place of the usual navigation bar (that contains search and classifier options). This alters the html that appears in the dynamically generated _navigationbar_ macro. Default: not set
VList formatstring	Applies to all vertical lists, unless overridden by a more specific format item. These include search results, classifier lists, and document table of contents
HList formatstring	Applies to all horizontal lists. Horizontal lists are often used in classifiers, particularly AZ[Compact][Section]Lists
DateList formatstring	Applies to all date lists - these are the vertical lists generated by a DateList classifier.
SearchVList formatstring	The vertical list of search results
DocumentVList formatstring	The document table of contents
CL1VList formatstring	Applies only to the vertical list of classifier 1
CL1HList formatstring	Applies only to the horizontal list of classifier 1
CL1DateList formatstring	Applies only to the DateList in classifier 1

### Format statement elements

Item	Description
[link][link]	Link to the document (Greenstone version)
[srclink][srclink]	Link to the original document (only if the original was converted to another form)
[icon]	An appropriate icon for a classifier/document node. E.g. bookshelf, book, chapter, page
[srcicon]	An appropriate icon for the original source document. E.g. Word, PDF, PS icon.
[num]	The document number (position in the search results - useful for debugging)
[numleafdocs]	The number of documents below the current classifier node. This is often used as a test for classifier nodes, as numleafdocs will not be set for documents. This allows different formatting for classifier nodes and document nodes in a hierarchy.
[Text]	The text of the current section
[RelatedDocuments]	Related Documents info (if available). This is a vertical list of Titles (or Subjects if Titles aren't available) that link to the related documents. It is based on "relation" metadata, which is a space separated list of collection,OID pairs.
[highlight][highlight]	These are used for 'highlighting' (actually bolding) the selected section in a hierarchical table of contents, and the selected node in a classifier. Apart from those two cases, this has no effect. If you actually want to highlight/bold/italicise something in a list, and have it apply to all items in the list, then either use actual html tags, like <b></b>, <u></u> and <i></i>, or use the _starthighlight_ and _endhighlight_ macros (defined in <i>macros/base.dm</i> ).
[Summary]	Displays Summary metadata if available, otherwise displays a short summary created on the fly.
[DocOID]	The internal identifier of the document
[DocRank]	The rank of the current document - used in search results
[DocImage]	The URL to the cover image of the document

[collection]	The directory name of the collection this document is from - useful in cross-collection searching. (version 2.61)
[collection:meta-name]	A collection metadata for the collection this document is from - useful in cross-collection searching. E.g. [collection:collectionname]. This will display in the current language if an appropriate version is available. (version 2.61)
[metadata-name]	The value of this metadata element for the document

### Extended metadata names

There are a few options for displaying metadata. The basic way is to specify e.g. [Title] or [dc.Title]: this displays the value of that particular metadata element for the current document/section. Metadata names can be prefixed by parent: or sibling. The following examples all use Title or Subject metadata, but any metadata could be used, including ones with namespaces (e.g. dc.Title). Any metadata name can also be prefixed by "cgisafe:". This results in the value being formatted so that it is safe to put in a URL.

[parent:Title]	The Title of the immediate parent section
[parent(Top):Title]	The Title of the topmost parent section
[parent(All):Title]	All Titles of the parent sections, separated by ", "
[parent(All' '):Title]	All Titles of the parent sections, separated by "' " (or whatever appears inside the '')
[child:Subject]	The Subjects of all child nodes of the current node, separated by '. (child modifier available from version 2.61)
[child(All'xxx'):Subject]	The Subjects of all child nodes of the current node, separated by xxx
[child(2):Subject]	The Subject of the second child of the current node. Child numbering starts from 1.
[child(last):Subject]	The Subject of the last child of the current node. 'first' is also a valid option.
[sibling:Subject]	All Subjects of the current section, separated by ", ". This is used for displaying metadata where there is more than one value. [Subject] will just display the first value.
[sibling(All' '):Subject]	All Subjects of the current section, separated by  .
[sibling(2):Subject]	The second Subject metadata value for the current node. Numbering starts from 1.
[sibling(last):Subject]	The last Subject metadata value for the current node. 'first' is also a valid option.
[parent:sibling:Subject]	sibling can be combined with parent to give all (or specific) values for the parent node(s). All parent and sibling qualifiers can be used. (version 2.61)
[child:sibling:Subject]	sibling can be combined with child to give all (or specific) values for the child node(s). All parent and child qualifiers can be used. (version 2.61)
[cgisafe:parent(Top):Title]	The Title of the topmost parent section, made safe for URLs.
[cgisafe:sibling(All' '):Subject]	All Subjects of the current section, separated by  , made safe for URLs.

### Conditional expressions in formatstrings

{If} {[metadata], action-if-non-null, action-if null}	<p>If there is a value for this metadata element, then output the first clause, otherwise output the second clause. Either clause is optional: if empty, nothing will be done for that case.</p> <p>This is useful for displaying classifier nodes differently to document nodes: {If} {[numleafdocs],format for classifier,format for document}</p>																											
{If} {[metadata] op value, action-if-true, action-if-not-true}	<p>Can do tests on metadata values. These can be string comparisons, or numeric comparisons. Valid operators are:</p> <table><tr><th>String</th><th>Numeric</th><th>Meaning</th></tr><tr><td>eq</td><td>==</td><td>equals</td></tr><tr><td>ne</td><td>!=</td><td>not equals</td></tr><tr><td>gt</td><td>&gt;</td><td>greater than</td></tr><tr><td>ge</td><td>&gt;=</td><td>greater than or equal to</td></tr><tr><td>lt</td><td>&lt;</td><td>less than</td></tr><tr><td>le</td><td>&lt;=</td><td>less than or equal to</td></tr><tr><td>sw</td><td></td><td>starts with</td></tr><tr><td>ew</td><td></td><td>ends with</td></tr></table>	String	Numeric	Meaning	eq	==	equals	ne	!=	not equals	gt	>	greater than	ge	>=	greater than or equal to	lt	<	less than	le	<=	less than or equal to	sw		starts with	ew		ends with
String	Numeric	Meaning																										
eq	==	equals																										
ne	!=	not equals																										
gt	>	greater than																										
ge	>=	greater than or equal to																										
lt	<	less than																										
le	<=	less than or equal to																										
sw		starts with																										
ew		ends with																										

Note that only eq and ne are available for Greenstone versions 2.60 and earlier.

{Or} {[metadata], [metadata2], [metadata3]...}	<p>Each metadata is evaluated in turn, and the first one that exists is output</p> <p>Useful for cases where there are different namespaced version of the same metadata, e.g. {Or} {[dc.Title],[dls.Title],[Title],Untitled}. The last item can be plain text.</p>
nested If/Or	<p>{Or} can have another conditional as its final option, eg {Or} {[BookTitle],[Title],{If} {[XXX],aaa,bbb} }.</p> <p>The following is not valid: {Or} {[BookTitle],[Title]{If} {[XXX],aaa,bbb} }.</p> <p>{If} can have nested conditionals at either true/false option. eg.</p> <p>{If} {[numleafdocs],[Title],[BookTitle]{If} {[Subject],[Subject],unclassified} }</p>