

What's in collection configuration files?

One of the more difficult parts of using Greenstone is to come up with an appropriate configuration file for your new collection. It seems like a black art! To help learn how to do it, this document explains the configuration files for a few actual Greenstone collections:

- greenstone demo
- aircraft images
- msword and pdf demonstration
- bibliographic collection (using MG++)
- OAI collection
- collection using MARC records???

1. The Greenstone demo collection

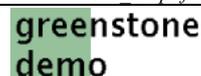
The Greenstone demo collection is supplied with the software, and is used extensively as an example in the documentation.

The configuration file (Section 1.2 below) that is distributed is unnecessarily complex: it contains some redundancy. Some of the plugins are not used, and some phrases are included to support non-English languages. We first examine the stripped-down version in Section 1.1, which defines an English-only collection which is essentially the same as the demo collection distributed with Greenstone. We call it *demo_simplified*.

Collection-level metadata. The first two blocks of lines in Section 1.1 (*creator*, *maintainer*, *public*; and four *collectionmeta* lines) are standard in all Greenstone collections. They give general information about the collection. The *collectionmeta* lines define the collection name, a brief description that appears on the collection's home page, and two versions of the collection's icon. The brief description (in *collectionextra*) can be seen on the collection's home page in Figure 1.

The *iconcollection* item gives the image proclaiming "greenstone demo" that appears on at the upper left of Figure 1: if it is absent, the collection's name appears instead. This image is placed in the *images* subdirectory of the collection's directory (typically, on Windows configurations, in *C:\Program files\gsdl\collect\demo_simplified\images*). The *iconcollectionsmall* is a slightly smaller version of the icon that is used on the Greenstone home page.

`collect\demo_simplified\images\demo.gif`

A rectangular icon with a black border. The word "greenstone" is written in a bold, lowercase, sans-serif font. Below it, the word "demo" is written in a bold, lowercase, sans-serif font. The text is white on a dark green background.

`collect\demo_simplified\images\demo.gif`

A rectangular icon with a black border. The word "greenstone" is written in a bold, lowercase, sans-serif font. Below it, the word "demo" is written in a bold, lowercase, sans-serif font. The text is white on a dark green background.

Plugins. The third block of lines gives the plugins used by the collection. *Documents* in the demo collection are in HTML, so *HTMLPlug* is included first. The *description_tags* option processes tags in the text that define sections and section titles as described below. The *input_encoding* option specifies the expected encoding of the input files: in this case it is *iso_8859_1* which indicates ????. The *cover_image* flag specifies that there is a *.jpg* cover image associated with each document: its name is ??? (I don't have the import

directory to look).

The other plugins *GAPug*, *ArcPlug*, and *RecPlug*, are used by Greenstone for internal purposes and are standard in almost all collections. The *use_metadata_files* flag directs Greenstone to look for *metadata.xml* files that specify metadata for the collection in XML format (see below).

Searchable indexes. The fourth block of lines (starting with *indexes*) specify the searchable indexes that are available in the demo collection. There are three: you can see them in Figure 1 because the pull-down menu has been clicked. The first index is called “chapters,” the second “section titles,” and the third “entire documents.” The first is the default one, and is shown on the page when the menu is not pulled down. The names of these three indexes are given by three *collectionmeta* statements.

The contents of the indexes—that is, the specification of what it is that will be searched—are defined by the *indexes* line that begins the fourth block. This specifies three indexes, two at the section level (beginning with *section:*) and one at the document level (beginning with *document:*). The difference is that a multi-word query will only match a section-level index if all the words appear within a single section, whereas it will match a document-level index if all words appear within a document, which typically comprises several sections. The first and third indexes are *section:text* and *document:text*, and the *:text* indicates that the full text of sections and documents respectively will be searched. The second is *section:Title*, which indicates that *Title* metadata will be searched—in this case, section titles.

Classifiers. The next block of lines, starting with *classify*, define the browsing indexes. There are four classifiers, corresponding (in order) to the four buttons (excluding *search*) on the navigation bar in Figure 1: *subjects*, *titles a–z*, *organisations*, and *how to*. The first allows access by subject. It is a *Hierarchy* classifier whose hierarchy is defined in *sub.txt* (the *hfile* argument); this file is given below. This classifier is based on *Subject* metadata, and when several books appear at a leaf of the hierarchy they are sorted by *Title* metadata (as you can see in Figure 2). The second allows access by title: it is an *AZList* classifier based on *Title* metadata. The third allows access by organization: it is a *Hierarchy* classifier based on *Organization* metadata whose hierarchy is defined in *org.txt*; this file is given below. Again, the leaves of the hierarchy are sorted by *Title* metadata. The fourth allows access by *Keyword* metadata: it is a plain *List* classifier based on this metadata.

The final statement in this block is a *format* statement. This applies to the fourth classifier (*CL4*), the *How to* information, and contains the specification *VList*, which refers to “vertical” lists produced by that classifier. The *[link]* ... *[/link]* structure puts in a hyperlink to the document; the *[Keyword]* puts in the text of the *Keyword* metadata.



Figure 1. The greenstone demo collection

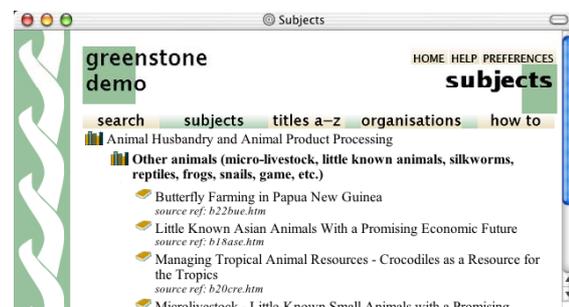


Figure 2. The *subjects* hierarchy browser

Format statements. The final block contains six format statements. The first (*searchVList*) applies to the search results page. The second applies to any other *Vlists*, and therefore to ...???. The third shows how the document text is formatted, with *Title* metadata ([*Title*]) in HTML *h3* format, followed by the text of the document [*Text*]. The fourth indicates that cover images are to be shown with each document. The fifth calls for the *Expand Text*, *Expand Contents*, *Detach* and *Highlight* buttons to be shown with each document. Finally, ???What is *HelpBookDocs* ???

Description tags

The description tags recognized by *HTMLPlug* are inserted into the HTML source text of the documents to define where sections begin and end, and to specify section titles. They look like this:

```
<!--  
<Section>  
  <Description>  
    <Metadata name="Title"> Realizing human rights for  
      poor people: Strategies for achieving the  
      international development targets </Metadata>  
  </Description>  
-->  
(text of section goes here)  
<!--  
</Section>  
-->
```

The `<!-- ... -->` markers are used because they indicate comments in HTML; thus these section tags will not affect document formatting. In the *Description* part other kinds of metadata can be specified, but this is not done for the style of collection we are describing here.

Metadata files

.metadata.xml files ???

The subject hierarchy file

The *sub.txt* file is shown below. The actual file is much larger, but most of it is not needed because it involves subjects that don't occur in the demo collection. The beginning and ending text strings on each line are the same; the number between defines the position in the hierarchy. The first string is matched against the metadata that occurs in the *.metadata.xml* file; the last one is the string that describes that node of the hierarchy on the web pages generated by Greenstone.

collect/demo_simplified/etc/sub.txt

```
"Society, Culture, Community, Woman, Youth, Population" 10 "Society, Culture,
Community, Woman, Youth, Population"
"Social sciences, sociology (works comprising several subgroups) incl. participatory
research and training" 10.6 "Social sciences, sociology (works comprising several
subgroups) incl. participatory research and training"
"Communication, Information and Documentation" 12 "Communication, Information and
Documentation"
"Communication, telecommunication, mass communication, mass media, film-making" 12.2
"Communication, telecommunication, mass communication, mass media, film-making"
"Agriculture and Food Processing" 13 "Agriculture and Food Processing"
"Better Farming series of FAO and INADES - 46 booklets" 13.8 "Better Farming series of
FAO and INADES - 46 booklets"
"Animal Husbandry and Animal Product Processing" 14 "Animal Husbandry and Animal
Product Processing"
"Cattle" 14.5 "Cattle"
"Other animals (micro-livestock, little known animals, silkworms, reptiles, frogs,
snails, game, etc.)" 14.6 "Other animals (micro-livestock, little known animals,
silkworms, reptiles, frogs, snails, game, etc.)"
"Settlements, Housing, Building - Infrastructure Construction (Roads etc)" 15
"Settlements, Housing, Building - Infrastructure Construction (Roads etc)"
"Settlements and housing: general works incl. low- cost housing, planning techniques,
surveying, etc." 15.3 "Settlements and housing: general works incl. low- cost
housing, planning techniques, surveying, etc."
"Development Periodicals and Magazines" 16 "Development Periodicals and Magazines"
"The Courier ACP 1990 - 1996 Africa-Caribbean-Pacific - European Union" 16.2 "The
Courier ACP 1990 - 1996 Africa-Caribbean-Pacific - European Union"
```

The organization hierarchy file

The *org.txt* file is shown below. Again, the actual file is much larger, but most of it is not needed for the demo collection. Again, the beginning and ending text strings on each line are the same because the metadata values in *metadata.xml* are exactly what should be shown on the Greenstone web pages. The number between defines the position in the hierarchy: in this case the hierarchy is flat and what is given is just an integer that determines the order of the list.

collect/demo_simplified/etc/org.txt

```
"BOSTID" 4 "BOSTID"
"EC Courier" 7 "EC Courier"
"FAO Better Farming series" 9 "FAO Better Farming series"
"World Bank" 16 "World Bank"
```

The full collection configuration file

The collection configuration file that is distributed with the greenstone demo collection, shown in Section 1.2, differs from what we have described in several small ways.

First, the lines appear in a different order. The ordering of lines in a collection configuration file is immaterial. They have been re-ordered in the stripped-down configuration file for convenience of presentation.

Second, the *collectionextra* text contains a newline symbol `\n`. This is the right way to put the newline character into format strings, but it does not in fact cause a new line to appear in the description on the collection's home page in Figure 1, because the output is expressed in HTML and newlines are ignored. To do that, the text should contain "`
`" instead.

Third, it includes several extra plugins: *ZIPPlug* (for zipped archives of files), *TEXTPlug* (for plain text files), *EMAILPlug* (for email files), *PDFPlug* (for PDF files), *RTFPlug* (for RTF files), *WordPlug* (for Microsoft Word files), *PSPlug* (for PostScript files). These are included in case the same collection configuration file is used for collections with different document types.

Fourth, two of the classifiers are slightly different: *Title* and *How to*. Figures 3 and 4 show the visible differences, with the stripped-down configuration file on the left and the full one on the right. In the the stripped-down collection configuration file *Title* is an AZList classifier, while in the full version it is a hierarchy classifier based on *AZList* metadata and using the *AZList.txt* file, while in. The difference is that the AZList classifier chooses how to split the alphabet into buckets of appropriate size, whereas the hierarchy classifier uses the AZList metadata to determine the bucket for each title, and this is assigned manually. In fact, the distinction in this case is minimal because the AZList classifier finds that there are so few documents that only one bucket is needed (and suppresses the A-Z selection), and the *AZList.txt* file contains the single line

```
collect/demo/etc/AZList.txt
T.1 1 "A-B-C-D-E-F-G-H-I-J-K-L-M-N-O-P-Q-R-S-T-U-V-W-X-Y-Z"
```

(visible in Figure 3(b)), which means that there is only one bucket. ???But what's the T.1??? The reason why the demo collection's configuration file is like this is to make it the same as the Development Library Subset's configuration file, where this metadata is used to determine the buckets.

The *How to* classifier is a List classifier in the stripped-down version and a hierarchy classifier in the full version, both based on *Keyword* metadata. This is to allow for the possibility that two different documents have the same *Keyword*. In this case the List classifier can only point to one of the documents, whereas each different *Keyword* in the hierarchy classifier opens up a bookshelf that can contain several documents. In fact, however, this does not happen with the documents in the demo collection. Figure 4 shows the difference.

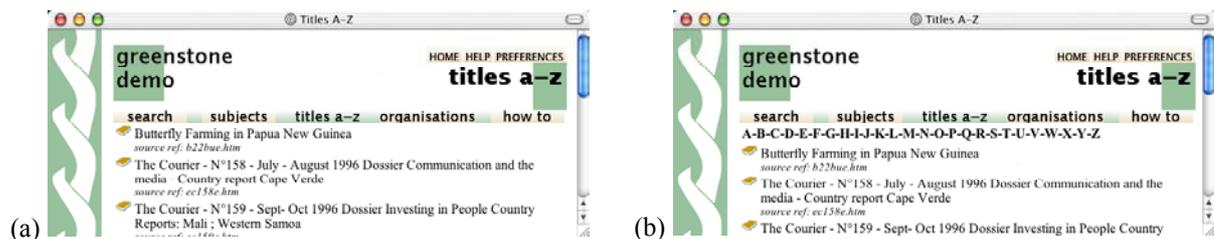


Figure 3. *Title* browser (a) using AZList classifier, (b) using hierarchy classifier

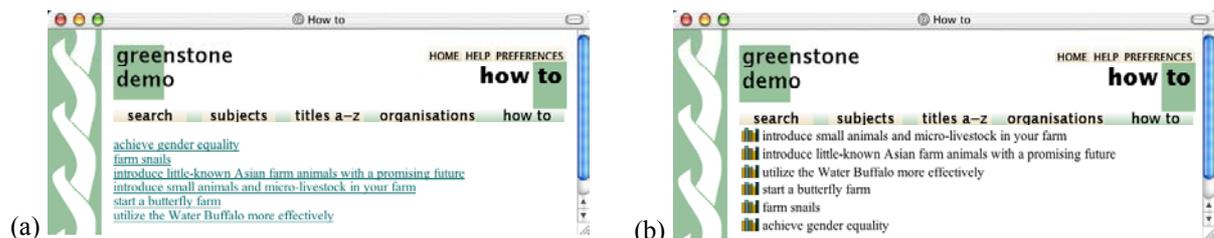


Figure 4. *How to* browser (a) using List classifier, (b) using hierarchy classifier

collect/demo/etc/keyword.txt

```
"introduce small animals and micro-livestock in your farm" 1 "introduce small animals
and micro-livestock in your farm"
"introduce little-known Asian farm animals with a promising future" 2 "introduce
little-known Asian farm animals with a promising future"
"utilize the Water Buffalo more effectively" 3 "utilize the Water Buffalo more
effectively"
"start a butterfly farm" 4 "start a butterfly farm"
"farm snails" 5 "farm snails"
"achieve gender equality" 6 "achieve gender equality"
```

1.1 Stripped-down configuration file for the demo collection

collect/demo_simplified/etc/collect.cfg

```
creator      greenstone@cs.waikato.ac.nz
maintainer   greenstone@cs.waikato.ac.nz
public       true

collectionmeta collectionname      "greenstone demo"
collectionmeta collectionextra     "This is a demonstration collection for the
Greenstone digital library software. It contains a small subset (11 documents) of
the Humanity Development Library"
collectionmeta iconcollectionsmall "_httpprefix_/collect/demo/images/demosm.gif"
collectionmeta iconcollection     "_httpprefix_/collect/demo/images/demo.gif"

plugin       HTMLPlug -description_tags -input_encoding iso_8859_1 -cover_image
plugin       GAPLug
plugin       ArcPlug
plugin       RecPlug -use_metadata_files

indexes      section:text section:Title document:text
defaultindex section:text
collectionmeta .section:text      "chapters"
collectionmeta .section:Title     "section titles"
collectionmeta .document:text     "entire documents"

classify     Hierarchy -hfile sub.txt -metadata Subject -sort Title
classify     AZList -metadata Title
classify     Hierarchy -hfile org.txt -metadata Organization -sort Title
classify     List -metadata Keyword -buttonname Howto
format       CL4VList      "<br>[link][Keyword][link]"

format SearchVList \
  "<td valign=top>[link][icon][link]</td><td>{If}{[parent(All':
  '):Title],[parent(All': '):Title]:}[link][Title][link]</td>"
format VList \
  "<td valign=top>[link][icon][link]</td><td
  valign=top>[highlight]{Or}{[Title],Untitled}[highlight]<i><small>{If}{[Date],<br>
  _textdate_[Date]}{If}{[NumPages],<br>_textnumpages_[NumPages]}{If}{[Source],<br>_t
  extsource_[Source]}</small></i></td>"
format       DocumentText  "<h3>[Title]</h3>\n\n<p>[Text]"
format       DocumentImages true
format       DocumentButtons "Expand Text|Expand Contents|Detach|Highlight"
format       HelpBookDocs  true
```

1.2 Full configuration file for the demo collection

collect/demo_simplified/etc/collect.cfg

```
creator greenstone@cs.waikato.ac.nz
maintainer greenstone@cs.waikato.ac.nz
public true

indexes section:text section:Title document:text
defaultindex section:text

plugin ZIPPlug
plugin GAPlug
plugin TEXTPlug
plugin HTMLPlug -description_tags -input_encoding iso_8859_1 -cover_image
plugin EMAILPlug
plugin PDFPlug -description_tags
plugin RTFPlug -description_tags
plugin WordPlug -description_tags
plugin PSPlug -description_tags
plugin ArcPlug
plugin RecPlug -use_metadata_files

classify Hierarchy -hfile sub.txt -metadata Subject -sort Title
classify Hierarchy -hfile AZList.txt -metadata AZList -sort Title -buttonname
  Title -hlist_at_top
classify Hierarchy -hfile org.txt -metadata Organization -sort Title
classify Hierarchy -hfile keyword.txt -metadata Keyword -sort Title -buttonname
  Howto

# To build this collection using the List classifier for the Howto
# listing (as shown in the Greenstone Developer's Guide) you should
# uncomment the following two lines and comment out the line above
# before rebuilding.
#classify List -metadata Keyword -buttonname Howto
#format CL4VList "<br>[link][Keyword]/link]"

format SearchVList \
  "<td valign=top>[link][icon]/link</td><td>{If}{[parent(All':
  '):Title],[parent(All': '):Title]:}[link][Title]/link</td>"
format VList \
  "<td valign=top>[link][icon]/link</td><td
  valign=top>[highlight]{Or}{[Title],Untitled}/highlight<i><small>{If}{[Date],<br>
  _textdate_[Date]}{If}{[NumPages],<br>_textnumpages_[NumPages]}{If}{[Source],<br>_t
  extsource_[Source]}</small></i></td>"

format DocumentText "<h3>[Title]</h3>\n\n<p>[Text]"
format DocumentImages true
format DocumentButtons "Expand Text|Expand Contents|Detach|Highlight"
format HelpBookDocs true

collectionmeta collectionname "greenstone demo"
collectionmeta collectionextra "This is a demonstration collection for the
  Greenstone digital library software.\nIt contains a small subset (11 documents) of
  the Humanity Development Library"
collectionmeta collectionextra [l=fr] "C'est une collection pour d\@monstration du
  logiciel Greenstone.\n Elle contient une petite partie du projet de biblioth\@ques
  humanitaires et de d\@veloppement (11 documents)."
collectionmeta collectionextra [l=es] "Esto es una colecci\@n de demostraci\@n para el
  software de biblioteca digital Greenstone. Contiene un peque\@o subconjunto (11
  documentos) de la biblioteca del desarrollo para la humanidad."
collectionmeta iconcollectionsmall "_httpprefix_/collect/demo/images/demosm.gif"
collectionmeta iconcollection "_httpprefix_/collect/demo/images/demo.gif"
collectionmeta .section:Title "section titles"
collectionmeta .section:Title [l=fr] "titres des sections"
collectionmeta .section:Title [l=es] "t\@tulos de las secciones"
collectionmeta .document:text "entire documents"
collectionmeta .document:text [l=fr] "documents entiers"
collectionmeta .document:text [l=es] "documentos enteros"
collectionmeta .section:text "chapters"
collectionmeta .section:text [l=fr] "chapitres"
collectionmeta .section:text [l=es] "cap\@tulos"
```

2. The aircraft images collection

The “kiwi aircraft images” collection is an example of a basic image collection. In this collection, there is no text and no (explicitly-specified) metadata. Several JPEG files are placed in the *import* directory prior to importing and building the collection, that is all.

One index is specified, but it is useless because there is no text. Omitting all index specifications should yield a collection whose navigation bar has no *index* button, which is what should have been done in this case. Unfortunately, Greenstone has a bug: it doesn't suppress the *search* button. ???

There is only one plugin: *ImagePlug*, aside from the three that are always included (*GAPlug*, *ArcPlug*, *RecPlug*). This automatically creates a thumbnail and generates the following metadata for each image in the collection:

<i>Image</i>	Name of file containing the image
<i>ImageWidth</i>	Width of image (in pixels)
<i>ImageHeight</i>	Height of image (in pixels)
<i>ImageSize</i>	Size of image (in bytes) EXCEPT THAT IT SEEMS SLIGHTLY OUT, and even unknown in Figs 5 & 6???
<i>Thumb</i>	Name of gif file containing thumbnail of image
<i>ThumbWidth</i>	Width of thumbnail image (in pixels)
<i>ThumbHeight</i>	Height of thumbnail image (in pixels)
<i>assocfilepath</i>	Pathname of image directory in the collection's <i>assoc</i> directory

The image is stored as an “associated file” in the *assoc* subdirectory of the collection's *index* directory. (*Index* is where all files necessary to serve the collection are placed, to make it self-contained.) The pathname *_httpcollimg_*, which is the same as *_httpcollection_/index/assoc*, refers to this directory. For any given document both the thumbnail and image occupy the same subdirectory of it, whose name is given by *assocfilepath*.

The second format statement in the configuration file, *DocumentText*, dictates how the document will appear, and Figure 5 shows the result. There is no document text (if there were, it would be producible by *[text]*). What is shown is the image itself, along with some metadata.

The configuration file defines one classifier, an *AZList* based on *Image*



Figure 5. Document in a simple image collection

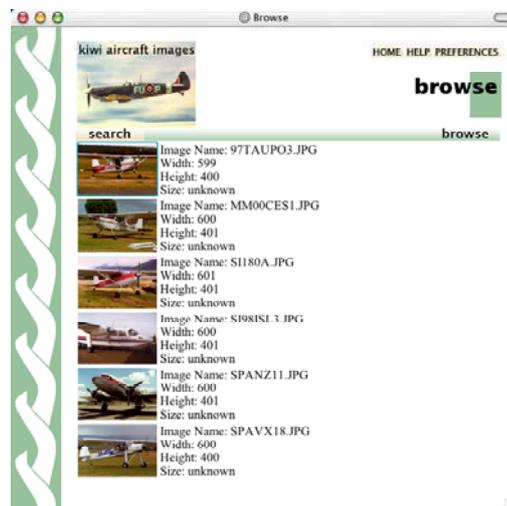


Figure 6. Browsing the collection

metadata, shown in Figure 6 (Greenstone has suppressed the alphabetic selector because this collection has only six images). The format statement shows the thumbnail image along with some metadata. (Any other classifiers would all have this format, since this statement does not name the classifier.)

You may wonder why the thumbnail image is generated and stored explicitly, when the same effect would be obtained by using the original image and scaling it:

```
<td>[link]<img src='_httpcollimg_/[assocfilepath]
/[Thumb]' width=[ThumbWidth] height=[ThumbHeight]>
[/link]</td><td valign=middle><i>[Title]</i></td>
```

The reason is to save communication bandwidth by not sending large images when small ones would do.

2.1 Configuration file for the aircraft images collection

collect\aircraft\etc\collect.cfg

```
creator          greenstone@cs.waikato.ac.nz
maintainer       greenstone@cs.waikato.ac.nz
public           true
beta             true

indexes          document:text
defaultindex     document:text

classify         AZList -metadata Image -buttonname Browse

collectionmeta  collectionname      "aircraft"
collectionmeta  .document:text      "documents"
collectionmeta  iconcollection      _httpprefix_/collect/aircraft/images/logo.gif

plugin           GAPlug
plugin           ImagePlug
plugin           ArcPlug
plugin           RecPlug

format VList ' <td valign="top">[link][/link]</td><td valign="top">Image
Name: [Image]<br>Width: [ImageWidth]<br>Height: [ImageHeight]<br>Size:
[ImageSize]</td>'

format DocumentText ' <center><table width=" pagewidth "><tr><td><br>Image Name: [Image]<br>Width:
[ImageWidth]<br>Height: [ImageHeight]<br>Size:
[ImageSize]</td></tr></table></center>'

format DocumentHeading ''
format DocumentButtons ''
```

The msword and pdf demonstration

The msword and pdf demonstration collection is a small collection that includes a few Word, RTF, PDF, and PostScript documents. It contains these four plugins (along with the standard three). All these plugins extract *Title* and *Filename* metadata. Third-party modules are used to extract text from these formats. The Greenstone team does not maintain these, although we do include the latest versions with each Greenstone release. Bugs arise with unusual Word documents (e.g. from older Macintosh systems), and sometimes the text is badly extracted. Some PDF files do not contain machine-readable text at all, containing instead a sequence of page *images* from which text could only be extracted by optical character recognition (OCR), which Greenstone does not attempt.

The *collectionname* and *collectionextra* collection-level metadata is given in both French and English (???Should be Spanish too).

The configuration file defines one classifier, an *AZList* based on *Title* metadata, shown in Figure 7 (the alphabetic selector is suppressed because this collection has only a few documents). However, no format statement is specified. In the absence of explicit information, Greenstone supplies sensible defaults. In this case, the default format for the classifier gives:

- an icon for the HTML version of the document (the text that is actually indexed, essentially the same as the Greenstone Archive format)
- an icon for the original version of the document (clicking it should opens the document in its original form)
- *Title* metadata, extracted from the document
- *Filename* metadata, extracted from the document.

Here is a format statement that achieves the same effect explicitly. It applies to all *Vlists*, and so controls both search results list and alphabetic title browser.

```
format VList "<td>[link][icon][link]</td>
<td>[srclink][srcicon][srclink]</td>
<td>[Title]<br><i>([Source])</i></td>"
```

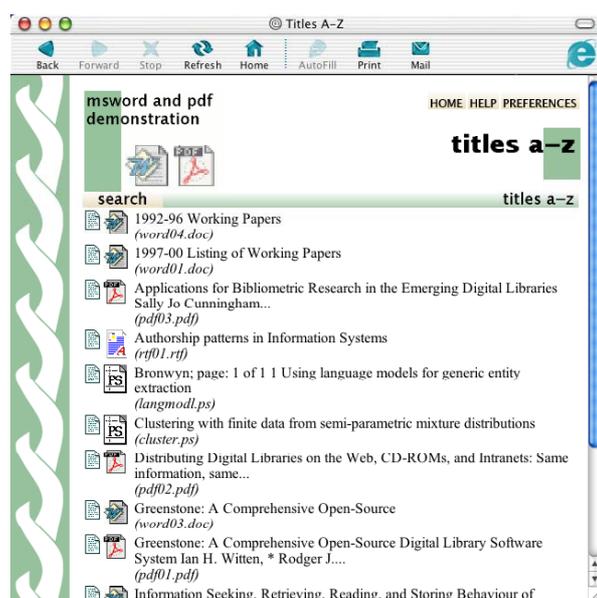


Figure 7. Browsing the msword and pdf demonstration

3.1 Configuration file for the msword and pdf demonstration

collect\wordpdf\etc\collect.cfg

```
creator      greenstone@cs.waikato.ac.nz
maintainer   greenstone@cs.waikato.ac.nz
public       true

indexes      document:text
defaultindex document:text

plugin       GAPlug
plugin       WordPlug
plugin       RTFPlug
plugin       PDFPlug
plugin       PSPlug
plugin       ArcPlug
plugin       RecPlug

classify     AZList -metadata Title

format DocumentHeading ""
format DocumentButtons ""

collectionmeta collectionname      "Word/PDF/RTF/PS demonstration"
collectionmeta collectionname [l=fr] "Word/PDF/RTF/PS d\@monstration"
collectionmeta iconcollection
    "_httpprefix_/collect/wordpdf/images/wordpdf.gif"

collectionmeta collectionextra      "This collection demonstrates Greenstone's
ability to build collections from documents provided in different formats.\n It
contains a number of papers written by various members of the NZDL project in PDF,
MSWord, RTF, and Postscript formats."

collectionmeta collectionextra [l=fr] "Cette collection d\@montre les capacit\@s de
Greenstone pour construire des collections \u00e0 partir de documents existants en
diff\@rents formats.\nElle contient plusieurs articles \u00e9crits par divers membres
du projet NZDL en format PDF, MS WORD, RTF, et Postscript."

collectionmeta .document:text      "documents"
```

❖ Adding node information to CL Hierarchy

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