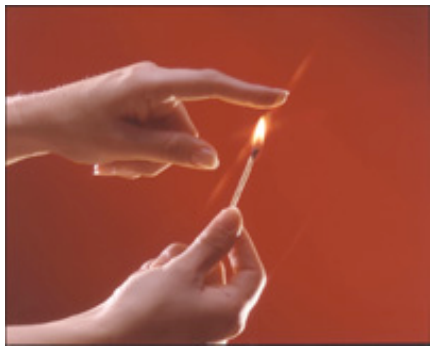


Customizing your collection; creating a CD-ROM



Course material prepared by

Greenstone Digital Library Project
University of Waikato, New Zealand

and

National Centre for Science Information,
Indian Institute of Science, Bangalore



Agenda

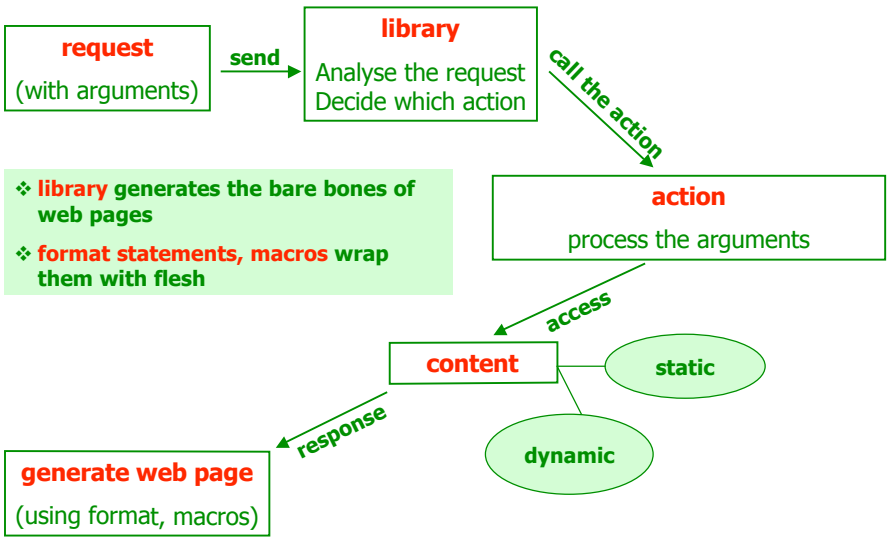
- How Greenstone generates pages
- What are macros?
- Personalizing your home page
- More on macros
- Customizing parts of a page
- Demo collections:
  - garish
  - authentication and formatting demo
- Creating a Greenstone CD-ROM

Customizing the interface

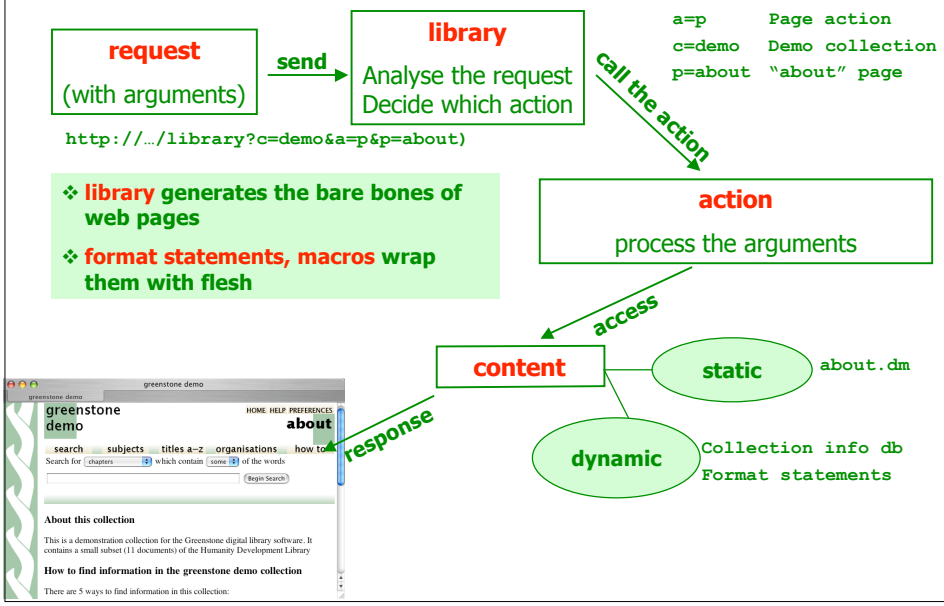
You may need to alter:

- Collection configuration file
  - gsdl/collect/mycol/etc/collect.cfg
- Images
  - gsdl/images
- Macro files
  - gsdl/macros/*.dm
- List of macro files used by Greenstone
  - gsdl/etc/main.cfg

Generating web pages



Generating web pages



Macros

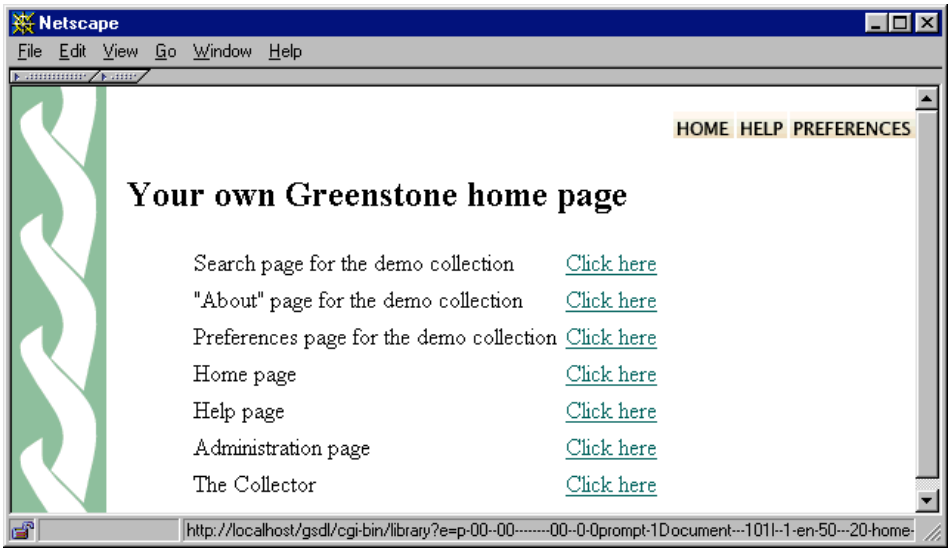
- Macros
  - let you customize presentation
  - present pages in different languages
  - print variables into the page text (e.g. number of search hits)
- Macro files
  - stored in gsdl/macros folder
  - named *.dm
  - each file defines one or more "packages"
  - loaded on startup (note difference between Local and Web Library)
  - listed in etc/main.cfg
- A "package" is a group of macros

Macro packages

Package Name	File Name	Roles
package Global	base.dm english.dm	Contains macros that work for all display screens, if the [c=collection name] is not specified.
Package Style	style.dm	Controls the display style for all pages, if the [c=collection name] is not specified.
Package document	english.dm document.dm	Contains macros for the document display.
Package query	query.dm english.dm	Contains macros for the Search page display.
Package preferences	pref.dm english.dm	Contains macros for the Preferences page display.
Package help	help.dm english.dm	Contains macros for the Help page display.
Package about	about.dm english.dm	Contains macros for the About page display.
Package browse	browse.dm english.dm	Contains macros for the dynamic browsing interface.

Personalizing Greenstone

C:\Program Files\gsdl\etc\main.cfg change *home.dm* to *yourhome.dm*



Demo:
customizing the home page

yourhome.dm

```
package home
_content_ {
<h2>Your own Greenstone home page</h2>
<ul>
<table>
<tr valign=top><td>Search page for the demo collection<br></td>
        <td><a href="_httpquery_&c=demo">Click here</a></td></tr>

<tr><td>"About" page for the demo collection</td>
        <td><a href="_httppageabout_&c=demo">Click here</a></td></tr>

<tr><td>Preferences page for the demo collection</td>
        <td><a href="_httppagepref_&c=demo">Click here</a></td></tr>

<tr><td>Home page</td>
        <td><a href="_httppagehome_">Click here</a></td></tr>

<tr><td>Help page</td>
        <td><a href="_httppagehelp_">Click here</a></td></tr>

<tr><td>Administration page</td>
        <td><a href="_httppagestatus_">Click here</a></td></tr>

<tr><td>The Collector</td>
        <td><a href="_httppagecollector_">Click here</a></td></tr>

</table>
</ul>
}
```

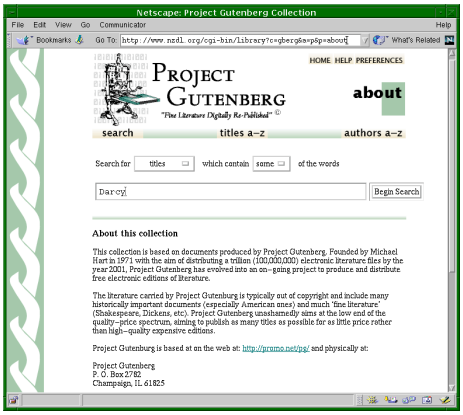
Macros used in home.dm

*_httppagehome_*	name of the home page
*_httppagehelp_*	… the help page
*_httppagestatus_*	… the administration page
*_httppagecollector_*	… the Collector page
httpquery&c=demo	search page for the demo collection
httppageabout&c=demo	about page for the demo collection
httppagepref&c=demo	preferences page for the demo collection
_content_{ … }	defines a macro called *_content_* contains HTML, but '{', '}', '\', and '_' must be escaped with a backslash
_header_{ … }	HTML page header (contains squirly bar)
_footer_{ … }	HTML page footer
main.cfg	contains list of macros, replace *home.dm* by *yourhome.dm* and put it in the *macros* directory

What do macros look like?

- Names begin and end with underscore
- Content is defined using curly brackets
  macroname [Argument] { content of the macro …}
- Content can be
  - plain text
  - HTML
  - links to Java applets and JavaScript
  - other macro names
- can contain conditional statements
  - _If_ …
- can take arguments
  - a=p, l=de etc.

about.dm




```
package about

################################
# about page content
################################

_pagetitle_ {_collectionname_}

_content_ {
<center>
_navigationbar_
</center>
_query:queryform_
<p>_iconblankbar_
<p>_textabout_
_textsubcollections_
<h3>_help:textsimplehelpheading_</h3>
_help:simplehelp_
}

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

Different interface languages

english.dm

```
#----------------------------------------------
# text macros
#----------------------------------------------


_textimageabout_ {About page}
_textimagehelp_ {Help page}
_textimagepref_ {Preferences page}
_textimagegreenstone_ {Greenstone Digital Library Software}

_textimagesearch_ {Search for specific terms}
_textimageCreator_ {Browse alphabetical list of authors}
_textimageTitle_ {Browse alphabetical list of titles}
_textmatches_ {Matches }
_textbeginsearch_ {Begin Search}

_texticonsmalltext_ {View this section of the text}
_texticondetach_ {Open this page in a new window}
_texticonexpandtoc_ {Expand table of contents}
_texticonexpandtext_ {Display all text}
_texticonhighlight_ {Highlight search terms}

_textlanguage_ {Interface language: }
_textsimplemode_ {simple query mode}
_textadvancedmode_ {advanced query mode (allows boolean searching u
```

chinese.dm

```
#---------------------------------------------------
# text macros
#---------------------------------------------------

_textimageabout_ [l=zh] {主页}
_textimagehelp_ [l=zh] {帮助页}
_textimagepref_ [l=zh] {选项页}
_textimagegreenstone_ [l=zh] {绿石数字化图书馆软件}

_textimagesearch_ [l=zh] {特定范围检索}
_textimageCreator_ [l=zh] {按作者名字的阿拉伯字母顺序浏览 }
_textimageTitle_ [l=zh] {按题名的阿拉伯字母顺序浏览}
_textmatches_ [l=zh] {匹配 }
_textbeginsearch_ [l=zh] {开始查询}

_texticonsmalltext_ [l=zh] {  浏览该部份文本}
_texticondetach_ [l=zh] {  打开一个新的窗口显示本页}
_texticonexpandtoc_ [l=zh] {  扩展书录表}
_texticonexpandtext_ [l=zh] {  显示所有文本}
_texticonhighlight_ [l=zh] {  查询项以粗体显示}

_textlanguage_ [l=zh] {界面语言：}
_textsimplemode_ [l=zh] {简单查询方式}
_textadvancedmode_ [l=zh] {高级查询方式(可以使用 !、&、| 和括号来布尔查询)}
```

Examples of macros

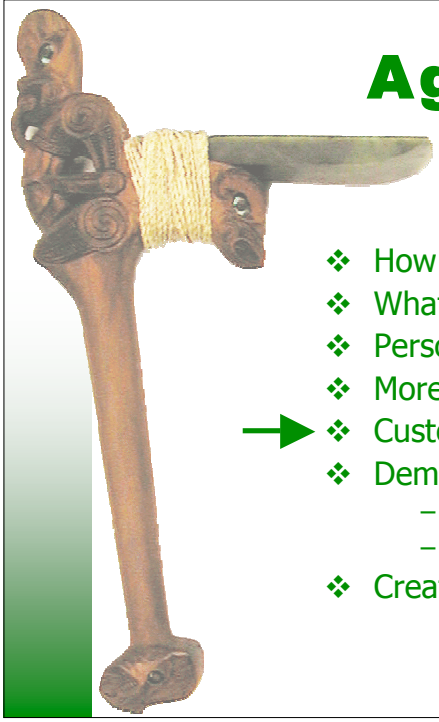
- _content_
  Defines the way the page looks
- _httppagehome_
  Generates the home page
- _httpquery_&c_=colname
  Generates search page of collection called "colname"

Creating collection-specific macros

- Create new macro package for collection
- Add collection specific macros to this file
  - Package line in the beginning is mandatory
- Can override any macro by passing collection argument in front of each macro
  - _content_ [c=collectionname] { }
- Put new macro file into etc/main.cfg
- See "garish" collection for an example

Customization hints

- To change the look and feel of Greenstone
  - edit the *base* and *style* packages
- To change the query page
  - edit *query.dm*
- To change the Preferences page
  - edit *pref.dm*



Agenda

- How Greenstone generates pages
- What are macros?
- Personalizing your home page
- More on macros
- Customizing parts of a page
- Demo collections:
  - garish
  - authentication and formatting demo
- Creating a Greenstone CD-ROM

Customizing parts of a page



❶ **In collect.cfg, define iconcollection**

iconcollection "_httpprefix_/path to your image/image file name"

e.g.

"_httpprefix_/collect/auhist/images/auhist.gif"

(can be gif, jpg, ...)

Customizing parts of a page



2 **Macro called _javalinks_ in package global in the base.dm file**

```
_javalinks_ {_imagehome__imagehelp__imagepref_}
```

Position of the buttons is in macro _pagebanner_ in the style.dm file. You can change the position of the buttons, add new ones, delete some, or display none at all

Customizing parts of a page



3 **Macro called _imagethispage_, e.g. in package help**

```
package help
_imagethispage_ [c=auhist] {_iconthispage_}
_iconthispage_ [c=auhist] {<img src="_httpiconhhelp_" alt="_textHelp_"
                              width=_widthhhelp_ height=_heighthhelp_>}
```

the icon is language-specific, so it's in english.dm

```
_httpiconhhelp_ { httpimg_/h\_help.gif}
```

Customizing parts of a page



4 **These buttons are created by classifiers, e.g. in collect.cfg**

```
classify AZList -metadata Title
```

buttons are defined in package Global in base.dm

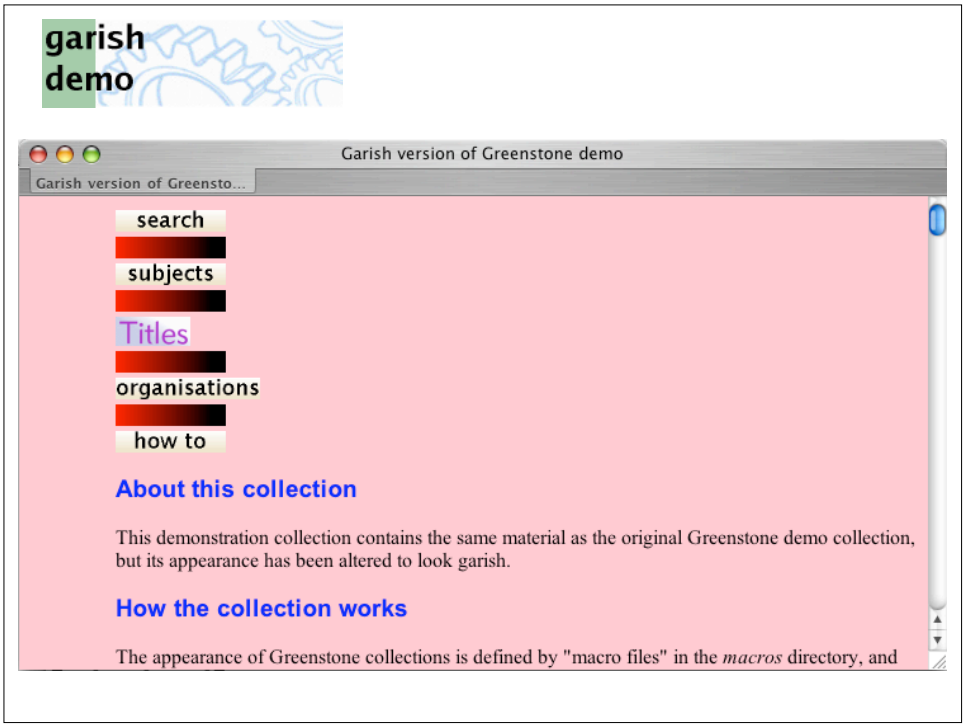
```
_imageTitle_ {_gsimage_(_httpbrowseTitle_, _httpiconttitlof_, _
                _httpiconttitlon_,titles,_textimageTitle_)}
```

images (_httpiconttitlof_, _httpiconttitlon_) are in english.dm

Customizing parts of a page



5 **The classifier list display is defined by a format string in collect.cfg**



Garish: package "Global"

```
#####################################################
package Global
#####################################################
_starthighlight_ [c=garish] {<b>}
_endhighlight_   [c=garish] {</b>}
_imagespacer_ [c=garish]
   { </nobr> <br/>
     <img src="_httpimg_/garish/horizontal_line.gif"
          width="87" height="17">
     <br/><nobr> }
_httpiconttitlgr_ [c=garish]
   {_httpimg_/garish/title_green.gif}
_httpiconttitlon_ [c=garish]
{_httpiconttitlof_} # make it the same as off
_httpiconttitlof_ [c=garish]
   {_httpimg_/garish/title_button.gif}
_widthttitlx_ [c=garish] {59}
```

Highlight query terms in bold

Separates search and classifier buttons

Special Titles button: 3 gifs
- **gr for selected**
- **on for on (mouse hovers over it)**
- **of for off (not selected)**

Width is used when calculating gap between buttons

Garish: package "Style"

```
##############################################
package Style
##############################################
_htmlhead_ [c=garish] {<html>
<head>
    <title>_pagetitle_</title>
    <link rel="stylesheet"
          href="_httpimg_/garish/style.css"
          type="text/css"/>
  </head>
<body>
}

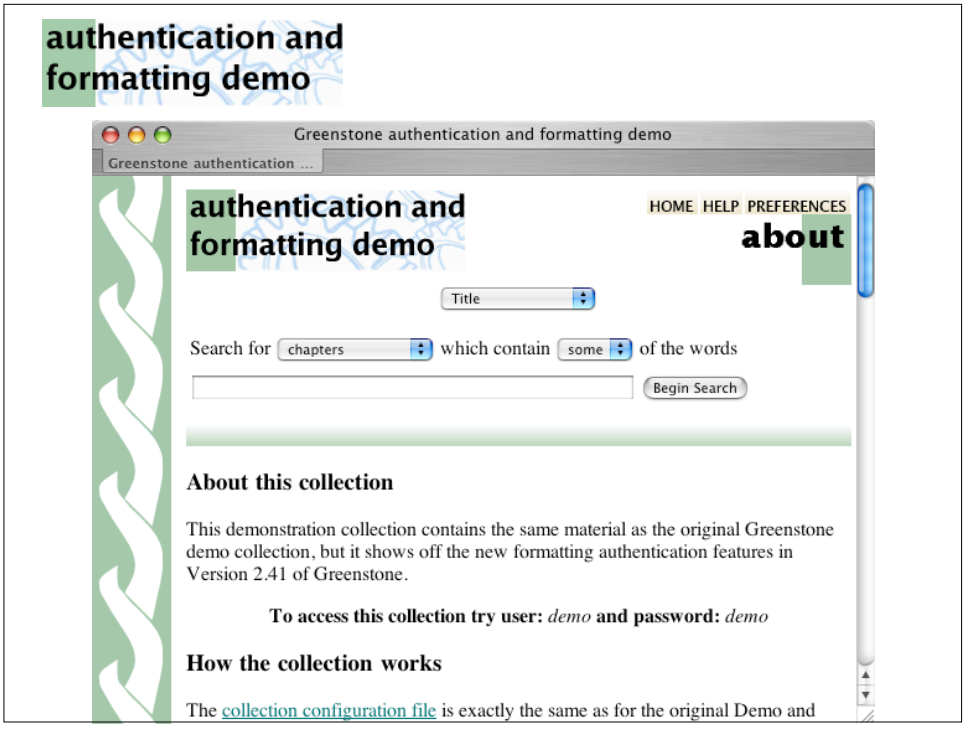
_pagebanner_ [c=garish] {}
```

Page header macro

This overrides it to make all pages use the style sheet

Pagebanner normally includes the collection's logo, home/help/preferences buttons, and the image at top left that identifies the page.

This overrides it to suppress these images





Agenda

- How Greenstone generates pages
- What are macros?
- Personalizing your home page
- More on macros
- Customizing parts of a page
- Demo collections:
  - garish
  - authentication and formatting demo
- Creating a Greenstone CD-ROM

Creating a CD-ROM

- You can export any Greenstone collection as a CD-ROM
- Self-installing
- Windows only (sorry!)
- While installing Greenstone you need
  - Full Installation
    (Compact Installation does not include the "Export to CD-Rom" feature)
- In the Librarian, go to File menu
- Click "Export to CD-ROM"

Note

- CD-ROM's created this way have not been tested extensively under different Windows configurations
- But they should work on all Windows platforms …
- … except 3.1/3.11 (is this a problem?)

Documentation and Help

- *Developer's Guide* (developer.pdf)
- Customizing the Greenstone user interface (http://www.wrlc.org/dcpc/UserInterface/interface.htm)
- Documented example collections:
  - garish
  - authentication and formatting demo

Questions?

Comments?

Discussion?

Feedback form!