Greenstone Documentation

Web library and Remote Collection Building with GLI Client

Web Library. This enables any computer with an existing webserver to serve pre-built Greenstone collections. As with the Local Library above, the Greenstone Demo collection will automatically be installed. You can also install the other collections on the CD-ROM

The Web Library differs from the Local Library because it is intended for computers that already have webserver software. To run the Web Library, you also need

Webserver software. One possibility is Apache

Windows webserver configuration (Web Library version only)

An advantage of the Local Library version of Greenstone is that it runs An advantage of the Local Library version of Greenstone is that it runs. If you already have a webserver, some small changes have to be made to its configuration to make your Greenstone installation operate. You may need help from a system administrator to reconfigure an existing webserver—they should be able to understand the instructions printed by the install script.

If you do not already have a webserver, you will have to install one. Then you will have to configure it appropriately. Section 4 gives a detailed account of the parts of a webserver installation that affect Greenstone, and how they need to be altered. It comes down to including half a dozen or so lines in a
configuration file.

Setting up the Webserver

In this section we describe how to set up your webserver to work with Greenstone. Note that all this is unnecessary when using the Windows Local Library, because this software works "out of the box" and does not require a webserver.

We discuss both the Apache webserver, which is freely available for both Windows and Unix (see the Appendix for details) and Microsoft's Personal Web Server (PWS) and Internet Information Services (IIS)
webserver. PWS is the standard Microsoft server for Windows 95/98; IIS is the standard webserver for Windows 2000 and the forthcoming Windows XP; Windows NT can use either. The Apache description
applies equally to the Windows Web Library and Unix versions (though we use Windows-style terminology and pathnames); the PWS/IIS section applies only to the Windows Web Library.

Once you have installed your webserver, the next step is to install Greenstone. We will assume that during the install procedure you have taken the default action for each stage by clicking on the *Next* button. The result is that the directory *C:\Program Files\greenstone* is created and the Web Library binary is stored there, along with some supporting files.

All webservers use the special URL “localhost” to denote the computer that the webserver is running on. Thus when you install a webserver, you can get at your HTML documents by typing the URL *http://localhost* into a browser. If your computer has a domain name set up, this is used instead of localhost to identify your computer from remote sites. Thus on the New Zealand Digital Library’s computer, *http://nzdl.org* and *http://localhost* are equivalent. If you type *http://nzdl.org* on your computer you will get the New Zealand Digital Library webserver, whereas if you type *http://localhost* you will get your own computer’s webserver.

The Apache web server

The Apache webserver is usually installed in *C:\Program Files\Apache Group\Apache* and is configured so that the cgi-bin directory is in the subdirectory *\cgi-bin* and the document root is the subdirectory *\htdocs*. It is reconfigured by editing the configuration file *C:\Program Files\Apache Group\Apache\conf\httpd.conf*. This is a text file: it’s quite easy to read it to see how things are set up.

Depending on how your computer’s networking software is set up, you may have to add this line to Apache’s *httpd.conf* configuration file:

```
ServerName localhost
```

If this line is not included, the system attempts to find your server’s name. However, there are bugs in some versions of Windows that cause this to fail. In this case, Apache will exit immediately when you start it up. It does display an error message, but it is immediately erased and you probably can’t read it.

Setting up the Greenstone cgi-bin directory

Cgi-bin is a directory from which the webserver treats documents as executable programs. Apache’s *ScriptAlias* directive is used to create a cgi-bin directory. Note that this directive can make any directory a cgi executable directory—it doesn’t have to be called “cgi-bin”! Conversely, a directory called “cgi-bin” isn’t special unless *ScriptAlias* has been applied to it.

When installed, Apache has a cgi-bin directory of *C:\Program Files\Apache Group\Apache\cgi-bin*. This means that if presented with the URL *http://localhost/cgi-bin/hello*, the webserver will attempt to execute a file called *hello* from within the above directory.

There is one Greenstone program, which is called “library.exe”, that needs to be executed by the webserver; it in turn reads a file called the Greenstone site configuration file, or “gsdlsite.cfg”, which needs to be located in the same directory.

The best way of arranging this is to use Apache's *ScriptAlias* directive to create a new cgi-bin directory. Here's the excerpt from Apache's *httpd.conf* configuration file that adds *C:\Program Files\Greenstone\cgi-bin* as an additional cgi-bin directory:

```
ScriptAlias  /gsdl/cgi-bin/  "C:/Program  Files/Greenstone/cgi-
bin/"
<Directory C:/Program Files/Greenstone/cgi-bin>
Options None
AllowOverride None
</Directory>
```

(It's a curious fact that Apache configuration files use forward slashes in place of standard Windows backslashes.) This means that any URLs of the form *http://localhost/gsdl/cgi-bin ...* will be sought in the directory *C:\Program Files\gsdl\cgi-bin*, and executed by the web server. For example, if presented with the URL

http://localhost/gsdl/cgi-bin/hello, the web server will attempt to retrieve the file *C:\Program Files\Greenstone\cgi-bin\hello* and execute it. However, the URL *http://localhost/cgi-bin/hello* looks in Apache's regular *cgi-bin* directory for the file *C:\Program Files\Apache Group\Apache\cgibin\*
hello and executes it, just as it did before.

The document root directory

The document root directory is the root of your webserver's directory structure. When installed, Apache has a document root *of C:\Program Files\Apache Group\Apache\htdocs*. This means that if presented with the URL *http://localhost/hello.html*, the webserver will attempt to retrieve a file called *hello.html* from within the above directory.

Several files within Greenstone need to be read by the webserver. The simplest way to arrange this is to use the *Alias* directive, which is just like *ScriptAlias* except that it applies to ordinary web pages, not cgi scripts.

Insert these lines into your Apache configuration file, after the *ScriptAlias* directive, to add *C:\Program Files\Greenstone* as an additional place to look for documents.

```
Alias /gsdl/ "C:/Program Files/Greenstone/"
<Directory C:/Program Files/Greenstone>
Options Indexes MultiViews FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

This means that any URLs that match the first argument of Alias (gsdl) are sought as files in the place corresponding to the second argument. In other words, URLs of the form *http://localhost/gsdl/ ...* will be sought as files in the directory *C:\Program Files\Greenstone*. For example, if presented with the URL *http://localhost/gsdl/hello.html*, the webserver will attempt to retrieve the file *C:\Program Files\Greenstone\hello.html*. However, the URL *http://localhost/hello.html* looks in the regular *htdocs* directory for the file *C:\Program Files\Apache Group\Apache\htdocs\hello.html*, just as it did before.

Be sure to add the *Alias* directive after the *ScriptAlias* directive. Instructing Apache to alias */gsdl* before */gsdl/cgi-bin* would match the URL */gsdl/cgi-bin/library* against the Alias directive rather than the
ScriptAlias, and it would be interpreted as a request for a document rather than the result of executing a program. The outcome would be to "display" the binary program file as a page in the Web browser, instead of executing it.

Security

You should be aware that if the web library version of Greenstone is set up as instructed above, anyone will be allowed to download any file in the *gsdl* directory structure. This includes the index files and source
documents of any collections you make, the user database, usage logs, etc.

If you are concerned about this, you can easily tighten up your webserver configuration to improve security. For the Apache webserver, put these lines into the configuration file instead of those given in the previous subsection:

```
Alias /gsdl/ "C:/Program Files/Greenstone/"
<Directory "C:/Program Files/greenstone">
Order allow,deny
Deny from all
<FilesMatch
"\.(gif|jpe?g|png|css|mov|mpeg|ps|pdf|doc|rtf|jar|class)$">
Order allow,deny
Allow from all
</FilesMatch>
</Directory>
```

This means that only files whose extensions match the regular expression in the *FilesMatch* line may be downloaded.

Building Greenstone collections remotely

Until now, using the GLI has required either a complete Greenstone installation, or the GLI applet to be installed on a server machine. In both cases, collaboration on collections is not possible.

This new functionality keeps the collections on a remote Greenstone server, thus allowing users to collaborate on collections (at different times), and without a local Greenstone installation

Installation

This section describes how to install the remote building functionality on server and client.

Server

The server can be a Linux, Mac OS X or Windows machine. It must have the Java run-time installed (version 1.4 or newer).

To install the server-side functionality:

1. Download Greenstone v2.72 and install it. Choose the "Web Library" installation option.

2. Configure your webserver for use with Greenstone:

   Apache v2.0 (Download)
   After installation, configure Apache by editing the conf\httpd.conf file as described in the Greenstone Installer's Guide or in the library.txt file installed in the Greenstone directory. Start Apache.
   Apache v1.3
   Apache v1.3 is not suitable for this application; use Apache v2.0 instead.

   1. Edit the first line of the Greenstone "cgi-bin/gliserver.pl" file and specify the full path of the perl binary. On Windows this will be (if installed in the default location):

      ```
          #!C:\Program Files\Greenstone\bin\windows\perl\bin\perl -w
      ```

      On Unix it is likely to be:

      ```
          #!/usr/bin/perl -w
      ```

   2. Visit

      ```
      http://<your-machine-name>:<port>/gsdl/cgi-bin/gliserver.pl?cmd=check-installation
      ```

      in a web browser. You should get a message saying "Java found" and "Installation OK!". If you get a message saying "Java failed", check that the Java run-time is installed and on the webserver's path. If you get a "500 Internal Server Error", check the error log of your webserver for the cause.

   3. (*Linux/Mac OS X*) Make the Greenstone "collect" directory writeable by the webserver user.

   4. Add some user accounts by visiting the Greenstone home page and clicking the "Administration Page" button, then "add a new user". See the Authentication section below for more information.

Client

The clients can be Linux, Mac OS X or Windows machines. To use the stand-alone GLI client:

1. Download gli-client-2.72.zip (5MB) and unzip it.

2. Run "client-gli.bat" (Windows) or "client-gli.sh" (Linux/Mac OS X). The first time you run the GLI client on a machine it will ask for the Greenstone library and gliserver URLs. These will be
   `http://<your-machine-name>:<port>/gsdl/cgi-bin/library`
   (library.exe if the server is Windows) and
   `http://<your-machine-name>:<port>/gsdl/cgi-bin/gliserver.pl`
   respectively.

Notes

General

IMPORTANT: Switch off Firewall (control Panel – Network and Internet Connections – Windows Firewall) to access the Greenstone Weblibrary interface through client machines.

There can be a lot of data transferred between the client and the server. This can make using the client impractical if you don't have a high speed connection between it and the server.

Authentication

The existing Greenstone user account system is used for authentication. User information is stored in the etc/users.db file, and the Administration pages (linked from your Greenstone library homepage) are used for adding, editing and removing users.

Groups are used to control the actions that users are allowed to perform on collections. **The group settings have changed for Greenstone v2.71, and you will need to edit your existing users if you are upgrading.** The possible group settings are:

- **all-collections-editor**: Users in this group can create new collections and edit all collections. (Equivalent to the "remote-superuser" group of Greenstone v2.70w and earlier).
- **personal-collections-editor**: Users in this group can create and edit "personal" collections. Personal collections have the user's username at the start of the internal collection name, and are created when the "this is a personal collection" option is ticked in the GLI "New Collection" dialog.
- **<collection-name>-collection-editor**: Users is this group can create and edit the "<collection-name>" collection. (Equivalent to the "<collection-name>-maintainer" group of Greenstone v2.70w and earlier).

For example, a user who needs to create and edit their own collections, and collaborate with others on a shared "papers" collection, should be in the "personal-collections-editor", and "papers-collection-editor" groups.

Collection locking

Each collection may only be open by one person at a time, to prevent synchronization problems. When a request is sent to the server to perform an action on a collection, the server will check for a gli.lck file in the collection directory. This file contains the username of the person who has the collection locked. When the collection is closed, this lock file is deleted.

If the collection is locked by someone other than the person making the request, the action fails. This is reported to the user on the client side, and this user is given the option of "stealing" the lock. Generally this is not recommended, since work may be lost if multiple users are editing a collection at one time. Stealing the lock should only be done in the case where the GLI has exited abnormally and the lock file was not deleted, and only after consulting with the user who has the collection locked.

E-mail notifications

The server can be configured to e-mail the system administrator whenever a collection finishes building. To enable this, edit the Greenstone "cgi-bin/gliserver.pl" file and set "$mail_enabled" to "1", and "$mail_to_address", "$mail_from_address", and "$mail_smtp_server" appropriately.

Missing functionality

There are a few items of functionality that are available in the standalone GLI but not in the client/applet version. These are:

- The Download pane
- The File → Write CD/DVD Image... menu item
- The File → Export... menu item
- The Rename option when right-clicking on a file or folder in the collection tree *(will be available in Greenstone 2.73)*
- The Replace option when right-clicking on a file or folder in the collection tree

This functionality may be added in the future.

Troubleshooting

If you are experiencing problems or error messages when using the client/server version of the GLI, please follow these steps:

1. Make sure you are using the latest version of Greenstone and have downloaded any patches on this page.
2. Record any popup GLI error messages, and the last action you performed.
3. Check for Java exceptions. If you're using the client version of the GLI, these will appear in the black GLI window (Windows) or in the terminal where you ran the GLI (Unix). If you're using the applet version of the GLI, these will appear in the Java Console (available from one of your browser menus -- for Firefox you may have to download this extension).
4. Check for errors at the bottom of the log files of your webserver. If you're using Apache (recommended), look in the "error_log" file in the Apache "logs" directory.
5. If you are having problems with the applet version, please check if you have the same problems with the client version.

If you think you have found a bug, or still can't get this functionality working, send a message to the Greenstone Users mailing list. Please include the following information:

- The operating system of the server machine

- The version of Greenstone installed on the server machine
- The version of Java installed on the server machine
- The operating system of the client machine
- Whether you are using the client or applet version of the GLI
- The actions you performed leading up to where the error or problem occurs
- The complete text of any popup GLI error messages, exceptions or errors in the webserver log file

Miscellaneous problems

- "Premature EOF" errors when building collections are caused by the webserver timing out when no output is generated by the build scripts for some time. The solution is to increase the webserver's timeout setting. For Apache this means increasing the "Timeout" value in the conf/httpd.conf file (don't forget to restart Apache).

Future Work

- When trying to load the dictionary (in Dictionary.java: ResourceBundle.getBundle(...)), the applet looks in the wrong place initially, causing errors in the Apache error_log. Finding some way of telling Java to look in the JAR file immediately would be nice.
- Pressing the "Cancel Build" button during the importing or building process doesn't have an immediate effect. The GLI code needs to be changed so GShell (and at a further level, RemoteGreenstoneServer) is a listener on the Cancel button. This will mean the cancel event can be processed much quicker.
- When the GLI is quit with jobs still on the remote Greenstone server queue, it will wait until these are finished before exiting. A dialog telling the user what is happening would be nice. This should probably have a "force quit" button, even if this is not recommended.
- Loading the options for DBPlug.pm causes an exception when using a Windows server. If you need to use DBPlug then you must install the DBI and DBD modules that it requires.

****