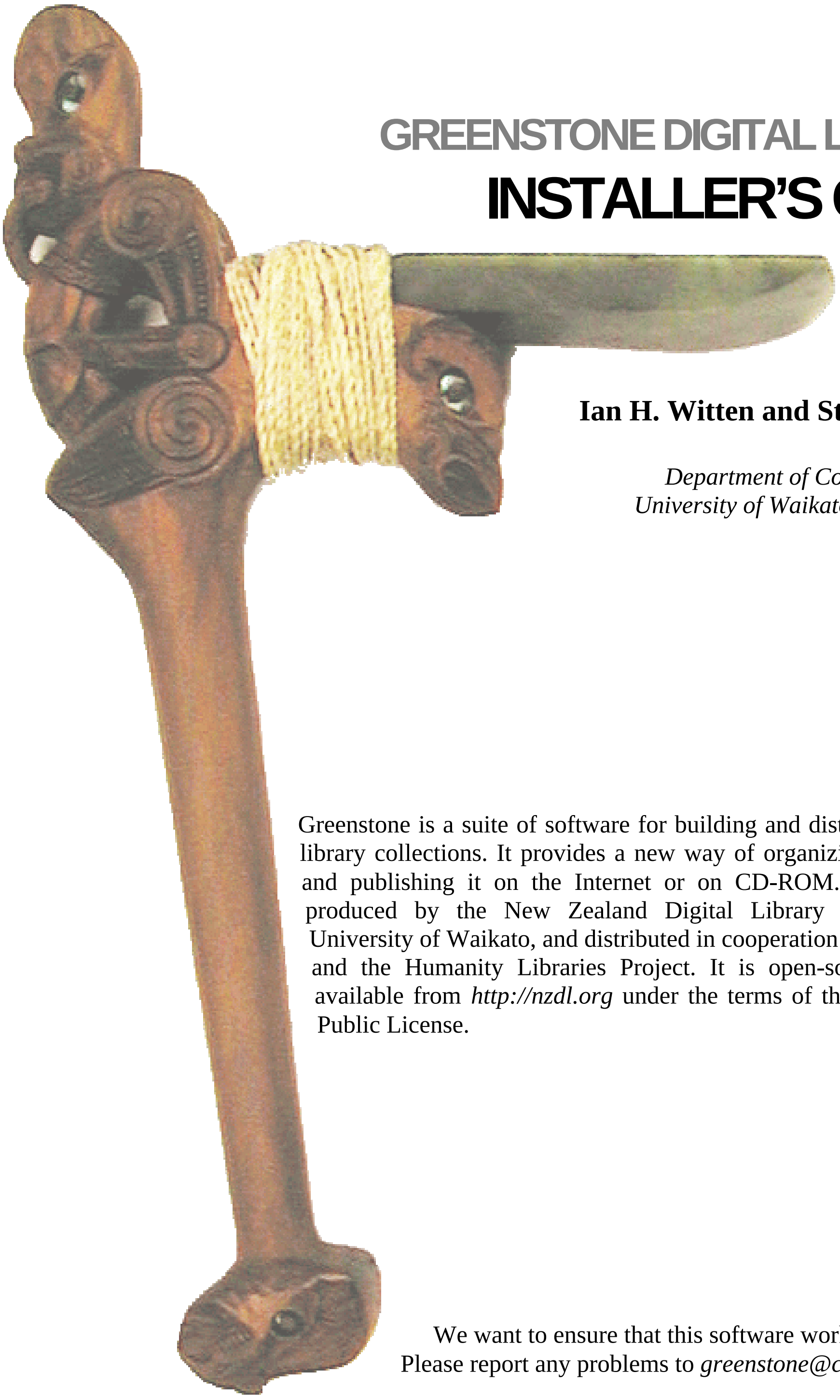


GREENSTONE DIGITAL LIBRARY
INSTALLER'S GUIDE

Ian H. Witten and Stefan Boddie

Department of Computer Science
University of Waikato, New Zealand

Greenstone is a suite of software for building and distributing digital library collections. It provides a new way of organizing information and publishing it on the Internet or on CD-ROM. Greenstone is produced by the New Zealand Digital Library Project at the University of Waikato, and distributed in cooperation with UNESCO and the Humanity Libraries Project. It is open-source software, available from *http://nzdl.org* under the terms of the GNU General Public License.

We want to ensure that this software works well for you.
Please report any problems to *greenstone@cs.waikato.ac.nz*

Greenstone gsdl-2.31 **February 2001**

About this manual

This document explains how to install Greenstone so that you can run it on your own computer. It also describes how to obtain associated software that is freely available—the Apache Webserver and Perl. We have striven to make the installation procedure as simple as it possibly can be.

The software runs on different platforms, and in different configurations. Consequently there are many issues that affect (or might affect) the installation procedure. Section 1 mentions some questions that you will need to consider before installing Greenstone. Section 2 details the installation procedure for all the different versions; you need only read the parts that relate to your operating system. Section 3 describes the demonstration digital library collections that are included in the distribution. Section 4 lists the various pieces of associated software and how to obtain them. Section 5 describes a typical Greenstone installation that uses the Apache webserver, and covers several features that need to be set up correctly for Greenstone to work properly. Finally, Section 6 shows how to make a personalized home page for your Greenstone digital library installation.

Companion documents

The complete set of Greenstone documents include three volumes:

- Greenstone Digital Library Installer's Guide *(this document)*
- Greenstone Digital Library User's Guide
- Greenstone Digital Library Developer's Guide

Acknowledgements

The Greenstone software is a collaborative effort between many people. Rodger McNab and Stefan Boddie are the principal architects and implementors. Contributions have been made by David Bainbridge, George Buchanan, Hong Chen, Elke Duncker, Carl Gutwin, Geoff Holmes, John McPherson, Craig Nevill-Manning, Gordon Paynter, Bernhard Pfahringer, Todd Reed, Bill Rogers, and Stuart Yeates. Other members of the New Zealand Digital Library project provided advice and inspiration in the design of the system: Mark Apperley, Sally Jo Cunningham, Steve Jones, Te Taka Keegan, Michel Loots, Malika Mahoui and Lloyd Smith. We would also like to thank all those who have contributed to the GNU-licensed packages included in this distribution: MG, GDBM, WGET, WV, PDF2HTML.

Contents

1
Versions of Greenstone

The Greenstone software runs on different platforms, and in different configurations, as summarized in Figure 1.

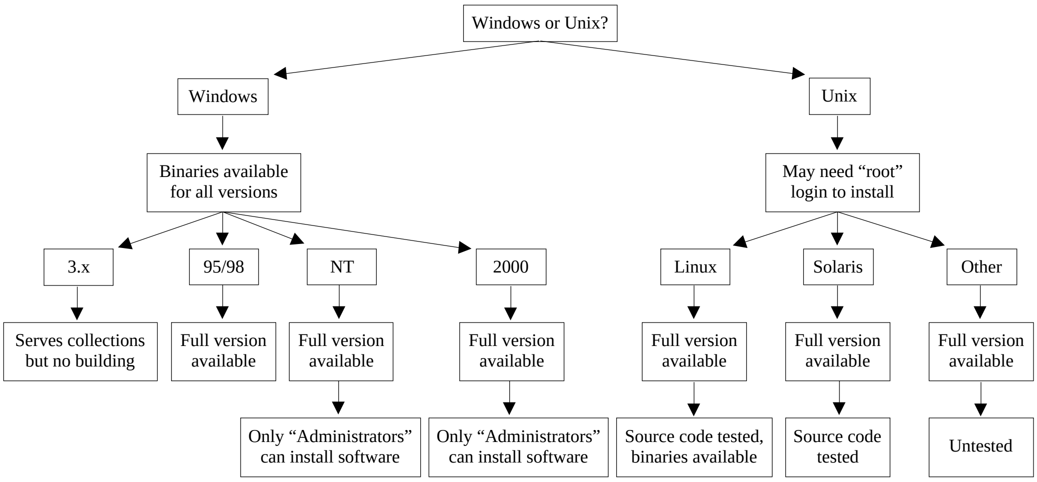

Windows or Unix?
Windows
Unix
Binaries available for all versions
May need "root" login to install
3.x
95/98
NT
2000
Linux
Solaris
Other
Serves collections but no building
Full version available
Full version available
Full version available
Full version available
Full version available
Full version available
Only "Administrators" can install software
Only "Administrators" can install software
Source code tested, binaries available
Source code tested
Untested

Figure 1 The different options for Windows and Unix versions of Greenstone

There are many issues that affect (or might affect) the installation procedure. Before reading on, you should consider these questions:

- Are you using Windows or Unix?
- If Windows, are you using Windows 3.1/3.11 or a more recent version? Although you can view collections on 3.1/3.11 machines, and serve other computers on the same network, you cannot build new collections. The full Greenstone software runs on NT, 95/98, and 2000.
- If Unix, are you using Linux or another version of Unix? For Linux, a binary version of the complete system is provided which is easy to

install. For other types of Unix you will have to install the source code and compile it. This may require you to install some additional software on your machine.

- If Windows NT/2000 or Unix, can you log in as the system “administrator” or “root”? This may be required to configure a webserver appropriately for Greenstone.
- Do you want the source code? If you are using Windows or Linux, you can just install binaries. But you may want the source code as well—it’s in the Greenstone distribution.
- Do you want to build new digital library collections? If so, you need to have Perl, which is freely available for both Windows and Unix.
- Is your computer running a webserver? The Greenstone software comes with a Windows webserver. However, if you are already running a Web server, you may want to stay with it. For Unix, you need to run a webserver.
- Do you know how to reconfigure your webserver? If you don’t use the Greenstone webserver, you will have to reconfigure your existing one slightly to recognize the Greenstone software.

2
The Installation Procedure

Versions of Greenstone are available for both Windows and Unix, as binaries and in source code form. The Greenstone user interface uses a Web browser: Netscape Navigator or Internet Explorer (version 4.0 or greater in both cases) are both suitable. Netscape is provided on the CD-ROM.

2.1 Windows

If you want just a simple, straightforward installation, go through the following “Windows installation” procedure.

If you choose anything other than the default Greenstone setup, you will have to decide whether you want to install the binary code, the source code, or both. The installation procedure is the same. The following sections tell you more about the options you will be presented with.

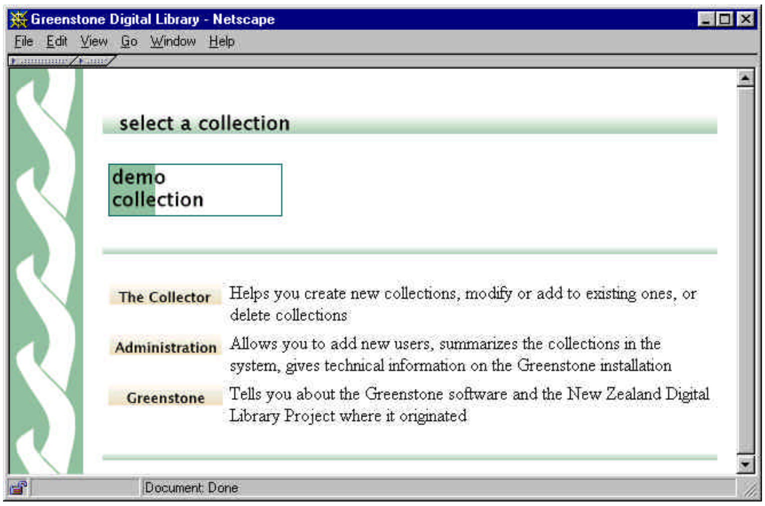
Simple installation

To install the Windows version from the CD-ROM, insert the disk into the drive (e.g. into *D:*). If the installation procedure does not start automatically after about 20 seconds, click on the *Start* menu, select *Run* and type *D:\Windows\Setup.exe*, where “*D*” is the letter that identifies your CD-ROM drive. For Windows 3.1, select *Run* from the “File manager” and type *D:\Windows\Setup.exe*.

For the simplest installation, just accept the default at each point by clicking the *Next* button. That’s all you need to do!

Once installation is complete, to start your Greenstone system click on the *Start* button, open the *Program* menu, and select *Greenstone DL Software*. This automatically starts your Internet browser and loads the Greenstone Digital Library home page, which should look something like the example in Figure 2. You enter the Greenstone Demo collection by

Figure 2
Your Greenstone home page


Greenstone Digital Library - Netscape
File Edit View Go Window Help

select a collection

demo collection

The Collector	Helps you create new collections, modify or add to existing ones, or delete collections
Administration	Allows you to add new users, summarizes the collections in the system, gives technical information on the Greenstone installation
Greenstone	Tells you about the Greenstone software and the New Zealand Digital Library Project where it originated

Document: Done

clicking on its icon.

Windows binaries

There are two separate Windows binary programs on the CD-ROM: the *Local Library* and the *Web Library*. The default installation described above selects the Local Library version. We strongly recommend that you use this version. The Web Library, which is much harder to set up, is only necessary if you already run a web server and experience port assignment conflicts when using the Local Library. Despite its modest name, it offers a complete, self-continued, web-serving capability.

Local Library. This enables any Windows computer to serve pre-built Greenstone collections. The Greenstone Demo collection will automatically be installed; you can also install the other collections on the CD-ROM (Section 3). The Local Library software is the same as that used on CD-ROMs produced by the Greenstone system.

The Local Library is intended for use on standalone computers or computers that do not already have webserver software. It contains a small built-in webserver so that other computers on the same network can also access the library. (However, the webserver has limited configurability.)

The Local Library software automatically determines whether your computer has network software installed or is connected to a network. It operates correctly under any combinations of these conditions. However, there are two possible problems that may be encountered. Greenstone may

- cause an unwanted telephone dialup operation;
- fail to run because network software is installed, but installed incorrectly.

A restricted version of the Local Library is supplied which is intended for use in these situations. The restricted version only works with Netscape (not Internet Explorer). When you invoke the Local Library version of Greenstone, the dialogue box contains a button that allows you to use the restricted version instead. Unless the above problems arise, you should always use the standard version.

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 (see Section 3).

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. We recommend Apache (see Section 4).

The Collector. This component, which is included in both the Local Library and the Web Library, allows you to build collections containing material of your choice. To run it, you also need

- Perl. We recommend ActivePerl (see Section 4).

(Because of this requirement, it is unlikely that you will be able to use the Collector on a Windows 3.1/3.11 machine.) On some versions of Windows, problems occur when using the Collector in conjunction with the Web Library—which is one reason we recommend the use of the Local Library.

Windows webserver configuration (Web Library version only)

An advantage of the Local Library version of Greenstone is that it runs “out of the box” and does not require any special configuration. For the Web Library version, however, you will have to make some adjustments to your webserver setup.

If you already have a webserver, some small changes have to be made to its configuration to make your Greenstone installation operate. The install script explains what these are. 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. (See Section 4 for information on the Apache webserver.) Then you will have to configure it appropriately. Section 5 gives a detailed account of the parts of an Apache 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.

On a Windows NT machine, you do not need to be the system "administrator" to go through the installation procedure above. When it comes to configuring an existing Apache server, however, you may need administrator privileges—it all depends on how Apache is set up. If you install Apache yourself, you can do it as a user without administrator privileges. If necessary you can always install a second Apache webserver on your computer—even if one exists already. Alternatively, just use the Local Library version of Greenstone rather than the Web Library.

Windows source

To compile the Greenstone source on Windows, you need

- The Microsoft Visual C++ compiler, Version 6.0 or greater. (We are currently sorting out some minor problems in compiling Greenstone with various Windows ports of GNU GCC.)

(You do not need GDBM, the Gnu database manager, because it is included in the Greenstone source distribution.)

It is unlikely that you will be able to compile Greenstone on a Windows 3.1/3.11 machine.

In the event that you recompile Greenstone and wish to use the recompiled version to create CD-ROMs, you should note that code produced by Version 6.0 of the Visual C++ compiler does not run under Windows 3.1/3.11, although there is no problem with later Windows systems (95, 98, NT, 2000). If you want your CD-ROMs to operate on early Windows machines, you will need a different version of the compiler. Moreover, Greenstone uses STL, the C++ standard template library, and although these compilers sometimes come with STL, the provided version does not always work properly. Hence to recompile Greenstone in such a way that it produces CD-ROMs that work on early versions of Windows, you need

- The Microsoft Visual C++ compiler, Version 4.0 or 4.2.
- An external version of STL, the C++ standard template library (www.sgi.com/Technology/STL).

2.2 How to find Greenstone

If you are using the Local Library, simply run the *Greenstone* program from the *Start* menu. This automatically starts your Internet browser and loads the Greenstone Digital Library home page. The Greenstone Demo collection should be accessible from this page. Using the URLs below you can access the Collector and Greenstone Administrative pages.

URLs

If you are using the Web Library, once you have installed the software and configured the webserver, use these URLs to enter your Greenstone system:

- Digital Library
  http://localhost/gsdl/cgi-bin/library
- Collector
  http://localhost/gsdl/cgi-bin/library?a=collector&p=intro
- Administration Page
  http://localhost/gsdl/cgi-bin/library?a=status&p=frameset

Links to the Collector and the Administration Page are provided on the digital library page.

Administrative user

Your “administrator” user is called *admin*, with a password that you specified as the last step of the installation process. That user is authorized to add new users, and to build collections.

Testing the Collector

The Collector allows you to build collections containing material of your choice. To test this, use the Collector to create a new collection. When you get to the *Source data* page, specify a directory that contains several Web pages. If you like, you can read files off the CD-ROM by specifying

```
file://D:/collect/dlpeople
```

on Windows, or

```
file:///cdrom/collect/dlpeople
```

on Unix. These contain the home pages of New Zealand Digital Library project members, for you to use as a test. Check when you get to the *Build collection* page that the building operation terminates correctly, and then look on your Greenstone home page for the new collection. More details appear in *Greenstone Digital Library User's Guide*, Section 3.

Troubleshooting

Problem	**Try this**
When I invoke Greenstone my computer asks me to dail up my Internet Service Provider	In certain unusual situations the Windows network software insists on doing this. To solve the problem, chose the "Restricted version" when you run Greenstone. This version only works with Netscape
When I try to install Greenstone on Windows, the procedure suddenly terminates without warning	Check that you have sufficient privileges to install software. On Windows 2000, for example, you have to be an "Administrator"
When I start Apache up, it quits immediately	Add a *ServerName localhost* directive to the Apache configuration file (see Section 5.1)
When I point my browser at the digital library, it can't find that page	Check your Internet Proxy settings and turn proxies off (use *Edit preferences* on Netscape or *Internet options* on Explorer)
When I point my browser at the digital library, it *still* can't find that page	Try using 127.0.0.1 in place of *localhost*. This reserved IP number is defined to be a "loopback" to your local computer
When I point my browser at the digital library, it displays a binary file	Check the *ScriptAlias* directive in the configuration file, making sure it comes before the *Alias* directive (see Sections 5.2 and 5.3)
When I point my browser at the digital library, it insists on dialing up my Internet Service Provider first	Try canceling the dialup dialog box and pressing your browser's *Reload* button. If the problem persists, use the Local Library version of Greenstone instead of the Web Library version
I'm using the Local Library and it still insists on dialing up my Internet Service Provider	Use the Restricted Version of the Local Library, which you can select when you invoke the *Greenstone* program

Problem	**Try this**
I get the Greenstone home page (Figure 2), but the Demo collection icon does not appear	Run the program *library* (in the cgi-bin directory) from the DOS (or shell) prompt to generate debugging information that will help you locate the problem
My browser complains that it can't find *main.cfg*	Check that the Greenstone files exist and are world-readable. If the *library* program runs OK from the command line, the problem is most likely file permissions (see Section 5.5). If it does not, it is likely that the *gsdlhome* variable is not set correctly in the *gsdlsite.cfg* configuration file (see Section 5.4)
I'm having trouble using the Collector	Read the *Greenstone Digital Library User's Guide,* Section 3

2.3 Unix

You need to choose whether you want to install the binary code or the source code.

Unix binaries

The binary code requires an Intel x86-based Linux distribution which includes ELF binary support. Distributions that meet these requirements include:

- RedHat 5.1
- SuSE Linux 6.1
- Debian 2.1
- Slackware 4.0

More recent versions of these distributions should also work.

You will need a webserver: we recommend Apache. If you want to build new digital library collections, you will also need Perl if this is not already on your system. To check, type *perl –v* and see if a message appears specifying, amongst other things, the version number. For most versions of Linux, Perl is installed by default. Section 4 gives information on how to obtain Apache and Perl.

Unix source

The source code is the same for Windows as for Unix. It has been compiled and tested on Linux and Solaris; it should be a fairly routine matter to port it to other flavors of Unix.

To compile the Greenstone source code on Unix, you need

- GCC, the Gnu C++ compiler.
- GDBM, the Gnu database manager.

To run the Greenstone software, you also need a Web server and Perl, as described above under *Unix binaries*.

Unix installation

To install the Unix version from the CD-ROM, insert the disk into the drive, and type

`mount /cdrom`	mount the CD-ROM device (this command may differ from one system to another)
`cd /cdrom`	change directory to the CD-ROM's top level
`cd Unix`	change directory to where the Unix install script resides
`sh Install.sh`	begin the installation process (an explicit *sh* is used because many installations forbid you to execute programs directly from CD-ROM)

The final command begins an interactive dialogue which requests the information that is needed to install Greenstone on your system, and gives detailed feedback on what is happening.

The installation procedure begins by asking you which directory to install Greenstone into. The first file placed there is the "uninstall" program that cleans up any partial installation, should you encounter problems or terminate the installation prematurely. Next you choose whether you want to install binaries or source code. You are then asked some questions about your webserver setup. You need to have a valid cgi executable directory (normally called "cgi-bin" on Unix systems); you can either create a new one or use your existing one (these options are explained in the dialogue). If you create a new one, you will need to enter this information in your webserver's configuration file. In either case you need to enter the web address of the cgi directory; again, this is explained in the dialogue. It is important to set the file permissions correctly on certain directories, and you are prompted for the necessary information.

Finally, you are prompted for a password for the “administrator” user *admin*.

Installing the binaries takes just a few minutes, enough time for you to answer the appropriate questions. If you install the source code, the installation script will compile it, which can take some time—from ten minutes to an hour or so, depending on the speed of your processor.

Once the software has been installed, you will be asked whether you want to install any Greenstone collections. The Greenstone Demo collection is installed automatically; other collections on the CD-ROM are described in Section 3.

Unix webserver configuration

If you already have a webserver, some small changes will have to be made to its configuration to make your Greenstone installation operate. The install script explains what these are. You will probably need help from your system administrator to reconfigure the webserver—he or she should be able to understand the instructions output by the install script.

If you do not already have a webserver, you will have to install one. Section 4 gives information on Apache. Then you will have to configure it appropriately. Section 5 gives a detailed account of the parts of an Apache 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.

You do not need to be the Unix “root” user to go through the installation procedure above. When it comes to configuring an existing Apache server, however, you may need “root” privileges—it all depends on how Apache is set up. If you install Apache yourself, you can do it as a user without “root” privileges. If you need to work your way around an uncooperative system administrator, you can always install a second Apache webserver on your computer—even if one exists already.

How to find Greenstone

Go to Section 2.2 above.

12 THE INSTALLATION PROCEDURE

3
Greenstone Collections

Several demonstration Greenstone collections are included on the CD-ROM. If you have Web access, many others can be downloaded, in either pre-built or unbuilt form, from the New Zealand Digital Library Project website (*nzdl.org*).

The Greenstone Demo collection is a small subset of a polished collection, and illustrates that relatively rich browsing capabilities can be provided (so long as suitable metadata is available). The other collections demonstrate various capabilities of Greenstone; you can add these to your own collection by using the Collector to clone the appropriate demonstration collection.

demo	Greenstone Demo	A small subset of the Humanity Development Library. If you clone this collection, the full facilities will only appear if your new files provide appropriate metadata information.
chinese	Chinese demo	A small collection of classic Chinese literature. You will need to change your Preferences (Presentation preferences) to Chinese.
gsarch	Greenstone archives	An archive of the Greenstone mailing list, which shows how the software can be used to search and browse E-mail archives. (This collection includes some MIME-encoded E-mails, which the E-mail plugin doesn’t yet handle properly—but they are still legible.)
folktales	Language extraction demo	Based on a collection of Japanese folktales that have been translated into a variety of languages, this demonstrates Greenstone’s automatic language identification ability.

fnl	Food and Nutrition Library	This is a larger collection in the same style as the Greenstone Demo collection, and like it, should only be cloned if you know about the metadata information that needs to be provided to make it work.
wordpdf	MSWord and PDF demo	Contains a small collection of papers, written by various members of the NZDL project, in both Microsoft Word and PDF format. The original source documents are provided for viewing. The HTML versions are used for full-text indexing, but they are poorly formatted because of deficiencies in the programs that convert documents to HTML.

The Greenstone Demo collection is installed automatically by the installation process. The other collections are included in pre-built form, and the install dialogue will ask you whether you want to install them.

The Chinese collection requires appropriate browser language support. It works with recent versions of Internet Explorer (Version 5 or greater), provided you download the Chinese character set (if you have not done this, Explorer will prompt you). It can be viewed on Unix with Netscape (4.5 or greater). If you want to use Netscape under Windows, you will have to download a Chinese language module like NJStar Communicator, and you may have to make Greenstone use a different coding (on the Preferences menu, select *GBK* rather than *UTF-8*, the default). To input Chinese you can use NJStar Communicator on Windows, or CXterm on Unix (copy/paste the Chinese characters to the search box—they are not displayed properly, but the search works).

4
Associated Software

Various pieces of software have been mentioned above: here is information on what versions are needed and how to obtain them.

4.1 Apache Webserver

To run any version of Greenstone apart from the Windows Local Library version, you need an external webserver. Many installations, particularly larger ones, will already have a webserver. If not, the Apache Webserver from *www.apache.org* is free, and easy to install.

4.2 Perl

To build collections with Greenstone you need the Perl language. Most Unix systems already have Perl installed. If not, source code and binaries for a wide range of Unix platforms are freely available at *www.perl.com*. For Windows, we recommend ActivePerl from *www.activestate.com*.

You need Perl Version 5 or higher.

4.3 GCC

The Unix version of Greenstone compiles under the Gnu C++ compiler, GCC. Greenstone makes extensive use of the C++ standard template library (we’ve found it to be broken on some older versions of GCC; please tell us if you have STL problems).

4.4 GDBM

All versions of Greenstone use the Gnu Database Manager, GDBM. It is supplied with all Windows versions of Greenstone and installed automatically during the installation procedure. Linux systems already have GDBM, so we do not provide it for Linux. Most other Unix systems have it, but if necessary you can download it from *www.gnu.org*.

5 A Typical Greenstone Installation

In this section we describe a typical Greenstone installation that uses the Apache webserver. The description applies equally to Windows and Unix: we use Windows-style pathnames. However, it doesn't apply to the Windows Local Library, because this software requires no webserver and works "out of the box".

The first step is to install Greenstone. We will assume that during the install procedure the default action for each stage has been taken by clicking on the *Next* button—except that the Web Library was chosen instead of the Local Library, and the directory *C:\gsdl* is specified instead of the default, which is a subdirectory of *C:\Program Files*. The result is that this directory is created and the Web Library binary, and some supporting files, are stored there.

5.1 Setting up 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 in *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.

The special URL "localhost" is used by Apache to denote the computer that the webserver is running on. Thus when you install Apache, 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.

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.

5.2 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:\gsdl\cgi-bin* as an additional cgi-bin directory:

```
ScriptAlias /gsdl/cgi-bin/ "C:/gsdl/cgi-bin/"
<Directory C:/gsdl/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:\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:\gsdl\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\cgi-bin\hello* and executes it, just as it did before.

5.3 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:\gsdl* as an additional place to look for documents.

```
Alias /gsdl/ "C:/Program Files/gsdl/"
<Directory C:/Program Files/gsdl>
  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:\gsdl*. For example, if presented with the URL *http://localhost/gsdl/hello.html*, the webserver will attempt to retrieve the file *C:\gsdl\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.

5.4 The gsdlsite.cfg configuration file

The install procedure creates a generic Greenstone site configuration file based on your installation choices. For a default installation the created file is *C:\gsdl\cgi-bin\gsdlsite.cfg* and its content is:

```
# Site configuration file for Greenstone.
# Lines begining with
# are comments.
# This file should be placed in the same directory as your library
# executable file. it should be edited to suit your site.

# points to the GSDLHOME directory
gsdlhome  C:/gsdl

# this is the http address of GSDLHOME
# if your webservers DocumentRoot is set to $GSDLHOME
# then httpprefix can be commented out
httpprefix /gsdl

# this is the http address of the directory which
# contains the images for the interface.
httpimg   /gsdl/images

# should contain the http address of this cgi script. This
# is not needed if the http server sets the environment variable
# SCRIPT_NAME
#gwcgi    /cgi-bin/library

# maxrequests is the most requests a fastcgi process
# will serve before it exits. This can be set to a
# low figure (like 1) while debugging and then set
# to a high figure (like 10000) when everything is
# working well.
#maxrequests 10000
```

You can customise your installation by editing this file, although you probably will not need to do so.

The *gsdlhome* line simply points to the *C:\gsdl* directory.

httpprefix is the web address of the directory that Greenstone is installed in. We explained earlier how to create an alias so that URLs of the form *http://localhost/gsdl/* ... are sought in the *C:\gsdl* directory. Putting a line *httpprefix /gsdl* into the *gsdlsite* configuration file establishes the same convention for the Greenstone software.

httpimg is the web address of the *C:\gsdl\images* directory, which contains all the gif images used in the interface. In any standard Greenstone installation this will always be *httpprefix/images*, and the line in the file above is left untouched.

gwcgi is the web address of the library cgi program. This is not required by most webservers (including Apache), and should remain commented out. Don’t uncomment it unless you’re sure you need to, because that may introduce problems.

maxrequests is only used by versions of Greenstone that are compiled with the “fast-cgi” option on. The standard binary distribution does not include this option because not all webservers are configured to support it. Fastcgi speeds up cgi executions by keeping the main executable in memory between invocations of the software, rather than loading it in from disk each time a web page is requested from the Greenstone software. The trade-off is the amount of memory used, which can grow the longer the program remains in memory. Once *maxrequests* pages have been generated, the cgi program quits, thereby freeing any accumulated memory. To respond to the next request for a Web page, the cgi program is read in from disk again, and a new cycle of page requests is begun. Most installations use the standard cgi protocol, which means that *maxrequests* can be safely ignored.

5.5 File permissions

This section is irrelevant for Windows 95/98, because these systems don’t identify the owners of files.

On Windows NT, 2000 and Unix systems, cgi scripts don’t run as normal users, because users can’t be identified over the Web. Instead, they might run as the user who started up the webserver program (common on Windows systems), or as a special user (commonly called *nobody* on Unix systems). Because of this, all files and directories within *C:\gsdl* need to be globally readable (or at least readable by the cgi-script user, perhaps “*nobody*”). To test whether file permissions are set up correctly, run the program *library.exe* from the command line. If the files are in the right places but the permissions are set incorrectly, it will run from the command line—that is, when *you* execute it—but not from a browser—that is, when the “*nobody*” user executes it. Another test is to log in as another user to see if the file permissions are specific to your original user account.

To work through a Web browser, all the Greenstone directories must be globally readable. Also, the *C:\gsdl\etc* directory and all its contents must be globally *writable*. This is the directory into which the library program write the usage log, error and initialization logs, and various user databases. If you’re reluctant to make this directory globally writable, you can set permissions so that just the files *errout.txt*, *initout.txt*, *key.db*, *users.db*, *history.db* and *usage.txt* are writable by the cgi user.

If file permissions are not set up correctly for *C:\gsdl\etc*, you may find that user authentication and search history do not work, and that no usage log (*usage.txt*) is generated.

6 Personalizing your installation

Probably the first thing you will want to do once your Greenstone installation is up and running is personalize the home page. The file that generates the Greenstone home page is called *home.dm*, and is located in the *macros* subdirectory of the directory into which you installed Greenstone. (The default for Windows systems is *C:\Program Files\gsdl.*) This is a plain text file that you will have to edit to create a new home page. Instead of editing it, we recommend creating a new file, say *yourhome.dm*. This will be like *home.dm* but will define “package home”—which is the bit that does the actual work—in a different way.

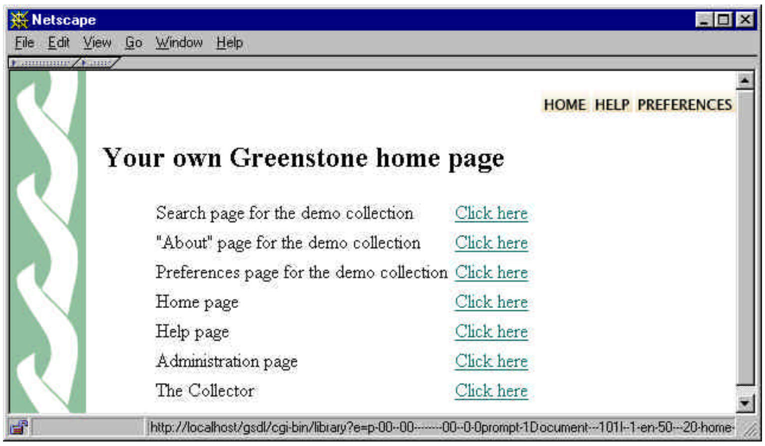
When you make a different home page, there must be some way of linking in to the digital library pages so that you can search and browse the collections on your system. The solution that Greenstone adopts is to use “macros”. That’s why the home-page file is called “.dm” and not “.html”—it’s a “macro” file rather than a regular HTML file. But don’t quail: the macro file basically contains just HTML, sprinkled with a few mystical incantantations which are explained below. The macro language is a powerful facility, and only a small part of it is described below—see the *Greenstone Digital Library Developer’s Guide* for more information.

6.1 Example

Figure 3 shows an example of a new digital library home page. Each of the “Click here” links takes you to the appropriate Greenstone facility. This page was generated by the file called *yourhome.dm* shown in Figure 4.

You can use Figure 4 as a template for creating your own specialized Greenstone home page. Basically, it defines a macro called *content*. Inside the curly braces is ordinary HTML. You could insert additional text,

Figure 3
Your own Greenstone home page



Your own Greenstone home page

Search page for the demo collection	Click here
"About" page for the demo collection	Click here
Preferences page for the demo collection	Click here
Home page	Click here
Help page	Click here
Administration page	Click here
The Collector	Click here

http://localhost/gsdl/cgi-bin/library?e=p-00-00-------00-0-0prompt-1Document---101l-1-en-50---20-home-

Figure 4
yourhome.dm used to create Figure 3

```
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>

}

# if you hate the squirly green bar down the left-hand side of the
# page, uncomment these lines:

# _header_ {
#           }
```

along with any HTML formatting commands, to put the content that you want to see on the page. The text is regular HTML; if you want you can include hyperlinks and use all the other facilities that HTML provides.

To make your new home page link in with other digital library pages, you need to use an appropriate magic spell. In this macro language, magic spells are words flanked by underscores. You can see these in Figure 4. For example, *_httppagehome_* takes you to the home page, *_httppagehelp_* to the help page, and so on. In some cases you need to include a collection name. For example, *_httpquery_&c=demo* specifies the search page for the demo collection; for other collections you should replace *demo* by the appropriate collection name.

The definition of the macro called *_content_* is plain HTML. Any standard HTML code may be placed within a macro definition. However, the special characters '{', '}', '\', and '_' must be escaped with a backslash to prevent them from being processed by the macro language interpreter.

Note that the *_content_* macro definition does not contain any HTML header or footer. If you want to change the header or footer of your home page, you should define *_header_* and/or *_footer_* macros, adding them to the *yourhome.dm* file in the form

```
_macroname_ {
  ...
}
```

For example, the squirly green bar down the left-hand side of Greenstone pages is defined in the *_header_* macro, and making this macro null will remove it, as indicated at the end of Figure 4.

6.2 How to make it work

You have to tell Greenstone about the new home page *yourhome.dm*. The system reads in the macro files that are specified in the main configuration file *main.cfg*, so if you create a new one you must include it there. Name clashes are handled sensibly: the most recent definition takes precedence.

Thus to make the Greenstone digital library software use the home page in Figure 3 instead of the default, first put the *yourhome.dm* file in Figure 4 into the *macros* directory. Then edit the *main.cfg* configuration file to replace *home.dm* with *yourhome.dm* in the list of macro files that are loaded at startup.

6.3 Redirecting a URL to Greenstone

You may want to redirect a more convenient URL to your Greenstone cgi program. For example, on our system the URL *http://nzdl.org* (which is shorthand for *http://nzdl.org/index.html)* is redirected to *http://nzdl.org/cgi-bin/library*. The Apache webserver accomplishes this with the *Redirect* directive. Along with other directives, this goes into the *C:\Program Files\Apache Group\Apache\conf\httpd.conf* configuration file. To redirect the URL *http://www.yourserver.com* to *http://www.yourserver.com/cgi-bin/library*, put this line into *httpd.conf*:

```
Redirect /index.html http://www.yourserver.com/cgi-bin/library
```

Then you will reach your digital library system directly from the URL *http://www.yourserver.com*. Instead, if you wanted a URL like *http://www.yourserver.com/greenstone* to be redirected to *http://www.yourserver.com/cgi-bin/library*, include in the *httpd.conf* file

```
Redirect /greenstone http://www.yourserver.com/cgi-bin/library
```

If your computer doesn't have a domain name (like the "www.yourserver.com" above), just replace *www.yourserver.com* by *localhost* in the lines above. So long as the browser is running on the same machine as the webserver—which it surely is if your computer doesn't have a domain name—this has the same effect as the above redirections.

Instead of putting redirect directives into the file *httpd.conf*, you can equally well put them into a file called *.htaccess* within your server's document root directory. In fact, doing so has two advantages. First, changes to *.htaccess* take effect immediately, whereas you have to restart the Apache webserver to see the effect of changes to *httpd.conf*. Second, on Unix systems you usually have to be logged in as the "root" user to edit *httpd.conf*, whereas you don't to edit *.htaccess*.