GNU Interactive Tools

A Set of Interactive Programs
Edition 2.9.4, for GNUIT version 4.9.5
January 2008

by Tudor Hulubei, Andrei Pitis and Ian Beckwith

GNUIT: A set of interactive tools, by Tudor Hulubei and Andrei Pitis.

This file documents the GNU Interactive Tools package.

1 Introduction

GNUIT is a set of interactive tools. It contains an extensible file system browser, an ascii/hex file viewer, a process viewer/killer and some other related utilities and shell scripts. It can be used to increase the speed and efficiency of most of the daily tasks such as copying and moving files and directories, invoking editors, compressing and uncompressing files, creating and expanding archives, compiling programs, sending mail, etc. It looks nice, has colors (if the standard ANSI color sequences are supported) and is user-friendly.

GNUIT runs on a wide variety of UNIX systems because it uses the GNU Autoconf package to get system specific information. Please refer to the PLATFORMS file included in the standard distribution for a detailed list of systems on which GNUIT has been tested.

One of the main advantages of GNUIT is its flexibility. It is not limited to a given set of commands. The configuration file can be easily enhanced, allowing the user to add new commands or file operations, depending on its needs or preferences.

GNUIT also provides a shell like command prompt, just to make sure that the entire power of the UNIX shell commands is still there.

2 Distributing GNU Interactive Tools

GNUIT is "free software"; this means that everyone is free to use it and free to redistribute it on certain conditions. GNUIT is not in the public domain; it is copyrighted and there are restrictions on its distribution, but these restrictions are designed to permit everything that a good cooperating citizen would want to do. What is not allowed is to try to prevent others from further sharing any version of GNUIT that they might get from you. The precise conditions are found in the GNU General Public License that comes with GNUIT and also appears following this section.

The easiest way to get a copy of GNUIT is from someone else who has it. You need not ask for our permission to do so, or tell any one else; just copy it. If you have access to the Internet, you can get the latest distribution version of GNUIT from host 'ftp.gnu.org' using anonymous login. See the file '/pub/gnu/GETTING.GNU.SOFTWARE' on that host to find out about your options for copying and which files to use.

You may also receive GNU Interactive Tools when you buy a computer. Computer manufacturers are free to distribute copies on the same terms that apply to everyone else. These terms require them to give you the full sources, including whatever changes they may have made, and to permit you to redistribute the GNU Interactive Tools received from them under the usual terms of the General Public License. In other words, the program must be free for you when you get it, not just free for the manufacturer.

3 Using GNU Interactive Tools

The `GNUIT` package contains three interactive programs and a few additional utilities. Here there is a description of each of them.

3.1 The GIT file system browser

`gitfm` is a file system browser with some shell like features designed to make your work much easier and much efficient. It displays one or two panels, each one containing a file system directory. You can browse the directory tree with the usual cursor keys, pressing *ENTER* when you want to enter or leave a directory and *TAB* when you want to change the panels.

Under the two panels there is a shell like input line which you can use to type normal shell commands. The input line can handle an unlimited number of characters and keeps a history of typed commands (using the GNU history library).

Under the input line there is a status bar. You can see there the status of the currently executed command, the warnings and errors and you will be prompted if a decision has to be taken.

3.1.1 Key binding conventions

`gitfm` now follows a new, easy to remember, scheme to bind commands on keys. This is only a convention, if you define new key bindings you may, or may not follow it.

All the file commands start with *^C*. This prefix can be followed by some modifiers, in order to affect the default behavior of the given command. These modifiers are *b* and *r*.

b - this modifier specifies that the command will run in background:

```
^CM = CHMOD; chmod %s{New mode of %i: ,%m} %i;;;;y
```

defines a command that changes the current selected files mode in foreground, while

```
^CbM = B-CHMOD; chmod %s{New mode of %i: ,%m} %i&;;;;y
```

defines a background command that does the same thing.

r - this modifier specifies that the command will be run recursively:

```
^CrM = R-CHMOD; chmod -R %s{New mode of %i: ,} %i;;;;y
```

defines a command that recursively changes the mode of the selected entries.

The *b* and *r* modifiers can be combined, the resulting command running recursively and in background:

```
^CbrM = B-R-CHMOD; chmod -R %s{New mode of %i: ,} %i&;;;;y
```

You should also note that for some commands (like `gzip`) there is no need for a non-recursive version. Running `gzip` recursively on files is harmless. If there is a directory between these files, `gzip` will recursively compress that directory, so you can use the same key binding for recursively and non-recursively compressing. In fact, it is a matter of selecting files or directories.

Unfortunately, we can't run `chmod` recursively trying to change the mode of all the files in a directory to 0644 because that directory might contain subdirectories and removing the execution permission from them is a bad idea. So, in this case, we need separate commands.

3.1.2 Command line

This is a brief description of the command line arguments.

`-h` print this help message

`-v` print the version number

`-c` use ANSI colors

`-b` don't use ANSI colors

`-l` don't use the last screen character

`-p` output final path at exit

The `-p` option can be used to make gitfm force bash (assuming that you're using it as your shell) chdir to the last directory gitfm was in before quitting. In order to do this, you need to invoke gitfm using this function (put it into your .profile):

```
function g
{
    gitfm -p $ 3> /tmp/gitfm.p.$$

    if test -s /tmp/gitfm.p.$$; then
if test -d "`cat /tmp/gitfm.p.$$`"; then
    cd "`cat /tmp/gitfm.p.$$`"
else
    cd
fi
    fi

    rm -f /tmp/gitfm.p.$$
}
```

This will not work if you suspend gitfm. Nothing bad will happen, just the chdir will not be performed.

3.1.3 Panel modes

`gitfm` has three major modes of displaying the panels. In the first (default) mode, two panels are displayed, each one using half of the screen. In the second mode, only one panel uses the entire screen. In the third mode, only the status bar and the input line are displayed, both panels being hidden.

Briefly, a panel can use the entire screen or just half of it. Even when a panel is hidden, it still exists.

Users can switch between these three major modes as needed:

^X 0

Enlarges the other panel to use the entire screen. It also changes the minor mode to '`Enable all`'. The current panel will become invisible ('`enlarge-other-panel`').

^X 1

Enlarges the current panel to use the entire screen. It also changes the minor mode to '`Enable all`'. The other panel will become invisible ('enlarge-panel').

^X 2

Switches back to the two panel mode ('two-panel-mode').

^O, ESC o

Switches to the tty mode (no panels on the screen) ('tty-mode').

A panel displays the files and subdirectories in a directory. You can optionally specify some additional information about each entry (file, directory, . . .) to be displayed (a minor mode).

When using the full screen mode, all the minor modes here can be used. In half screen mode, the 'panel-enable-all' mode is not available.

These are the panel minor modes:

ESC e o

Display the entry owner and group ('panel-enable-owner-group').

ESC e d

Display the entry date and time ('panel-enable-date-time').

ESC e s

Display the entry size ('panel-enable-size').

ESC e S

Display the entry size, scaled (e.g. '123M') ('panel-enable-abbrevsize').

ESC e m

Display the entry mode ('panel-enable-mode').

ESC e f

Display the entry full name ('panel-enable-full-name').

ESC e a

Display the entire information about file ('panel-enable-all').
This mode is only available if the panel has been enlarged to use the entire screen with 'enlarge-panel' or 'enlarge-other-panel' ('panel-enable-all').

There is another way of changing the panel minor modes:

^], ^[]

Switches to the next panel minor mode ('panel-enable-next-mode').

3.1.4 Sorting methods

Entries in a panel can be sorted in different ways. These are the available options:

ESC s n

Display the panel entries sorted by their names ('panel-sort-by-name').

ESC s e

Display the panel entries sorted by their extensions ('panel-sort-by-extension').

ESC s s

Display the panel entries sorted by their sizes
('panel-sort-by-size').

ESC s d

Display the panel entries sorted by their 'last modified' stamps
('panel-sort-by-date').

ESC s m

Display the panel entries sorted by their modes
('panel-sort-by-mode').

ESC s o i

Display the panel entries sorted by their owner ids
('panel-sort-by-owner-id').

ESC s g i

Display the panel entries sorted by their group ids
('panel-sort-by-group-id').

ESC s o n

Display the panel entries sorted by their owner names
('panel-sort-by-owner-name').

ESC s g n

Display the panel entries sorted by their group names
('panel-sort-by-group-name').

There is also another way to change the sort method:

ESC s u

Switch to the next panel sort method ('panel-sort-next-method').

3.1.5 Moving the cursor in the panel

Moving the cursor in the panel is very easy. If your keyboard has arrows, use them. If the arrow keys don't work (it might be due to a badly configured *TERM* environment variable), you can use the Emacs commands bindings as well.

UP, *^P*

Move the cursor vertically up one entry ('previous-line').

DOWN, *^N*

Move the cursor vertically down one entry ('next-line').

HOME, *ESC <*

Move the cursor on the first entry in the panel
('beginning-of-panel').

END, *ESC >*

Move the cursor on the last entry in the panel ('end-of-panel').

PGUP, *ESC v*

Move the cursor vertically down one page ('scroll-down').

PGDOWN, *^V*

Move the cursor vertically down one page (‘scroll-up’).

ESC g
: Scroll the panel entries to the left (‘horizontal-scroll-left’).

ESC j
: Scroll the panel entries to the right (‘horizontal-scroll-right’).

^X P
: In order to optimize the screen output, you can modify the scroll step (‘set-scroll-step’). This is the number of lines to try scrolling a panel when the cursor moves out. The ‘StartupScrollStep’ specifies the initial scroll step, but using ‘set-scroll-step’ you can dynamically change it.

TAB, *^I*, *^X o*
: Move the cursor in the other panel (‘other-panel’).

^X P
: Switch the two panels. This command works even when gitfm is not in the ‘two panels’ mode (‘switch-panels’).

3.1.6 Selecting files

INS, *^T*, *^X *, *^*
: Toggle the ‘selected’ flag of the current entry (‘select-entry’).

^C s
: Select (marks) all the files matching at least one pattern from a space separated list of shell patterns. Spaces and \s are allowed in the patterns but they have to be escaped with a \. The user will be prompted for a pattern to match against (‘select-files-matching-pattern’).

^C u
: Unselect (unmarks) all the files matching at least one pattern from a space separated list of shell patterns. Spaces and \s are allowed in the patterns but they have to be escaped with a \. The user will be prompted for a pattern to match against (‘unselect-files-matching-pattern’).

ESC +
: Select (marks) all the files having the same extension as the current file. If the current file name doesn’t have an extension or starts with a dot, no files are selected (‘select-extension’).

ESC -
: Unselect (unmarks) all the files having the same extension as the current file. If the current file name doesn’t have an extension or starts with a dot, no files are unselected (‘unselect-extension’).

See Section 3.1.8.8 [Selecting Files], page 10, for additional ways of selecting and unselecting files.

3.1.7 Incremental searching files in a panel

Users sometime need to search a file in a panel, especially when the panel contains a big number of entries. For that reason `gitfm` provides an incremental search feature. Using forward and backward incremental search, files can be very easy located. Wrapped incremental search is also provided.

`^S`, `^Xs`

Incremental search forward a file name in the current panel (‘`isearch-forward`’). Pressing `^S` or `^Xs` again will force `gitfm` to go to the next entry that matches the current isearched string. When the end of the panel is reached, the isearch is restarted from its beginning.

`^R`, `^Xr`

Incremental search backward a file name in the current panel (‘`isearch-backward`’). Pressing `^R` or `^Xr` again will force `gitfm` to go to the next entry that matches the current isearched string. When the beginning of the panel is reached, the isearch is restarted from its end.

3.1.8 Using the input line

The input line is one of the main methods used by `gitfm` to interact with the user. All the answers the user should give in order to perform some operation and all the shell like commands are built using it. So here is a description of all the basic editing operations that the ‘`input line`’ provides. They are very much inspired from `Emacs`, so `Emacs` users should have no problem using them.

3.1.8.1 Inserting Text

Typing characters is the most usual way of inserting text into the input line. Key sequences starting with printable ascii characters are not allowed in `gitfm` so typing `a` for example results in inserting `a` at the current point position. Of course, there are some other ways of inserting text into the command line and here there is a description of most of them.

ESC RET

Copy the current entry name into the input line at the current point position (‘`entry-to-input-line`’).

ESC ESC RET

Copy the other panel path into the input line at the current point position (‘`other-path-to-input-line`’).

`^X ^I`

Copy the names of all the selected entries into the input line at the current point position (‘`selected-entries-to-input-line`’).

3.1.8.2 Moving Point

`^B`, *LEFT*

Move the point backward one character (‘`backward-char`’).

`^F`, *RIGHT*

Move the point forward one character (‘`forward-char`’).

ESC b

Move the point one word backward (‘`backward-word`’).

ESC f

Move the point one word forward (‘`forward-word`’).

^A

Move the cursor at the beginning of the input line (‘`beginning-of-line`’).

^E

Move the cursor at the end of the input line (‘`end-of-line`’).

3.1.8.3 Deleting and killing text

DEL, *^D*

Delete the character under the cursor (‘`delete-char`’).

^H, *BKSPC*

Delete the character before the cursor (‘`backward-delete-char`’).

ESC BKSPC

Delete backward one word (‘`backward-kill-word`’).

ESC d

Delete forward one word (‘`kill-word`’).

ESC k

Delete the entire line (‘`kill-line`’).

^U

Delete all the characters between the beginning of the input line and the point (‘`kill-to-beginning-of-line`’).

^K

Delete all the characters between the point and the end of the input line (‘`kill-to-end-of-line`’).

ESC SPC

Delete all the spaces around the point, leaving only one space (‘`just-one-space`’).

*ESC *

Delete all the spaces around the point (‘`delete-horizontal-space`’).

^W

Save the region between the point and the mark into the kill "ring" and then kills it (‘`kill-region`’). Note that there is no real kill-ring here. The so-called kill-ring has only one entry.

ESC w

Save the region between the point and the mark without killing it (‘`kill-ring-save`’).

3.1.8.4 Case conversion of words.

ESC l

Convert the following word to lower case, moving over. (‘downcase-word’).

ESC u

Convert the following word to upper case, moving over. (‘upcase-word’).

ESC c

Capitalize the following word, moving over. (‘capitalize-word’).

3.1.8.5 Reusing recent input line arguments

A separate history is kept for both built-in and user-defined commands. If you call a command that you have used before, you can re-edit a previously entered string in order to minimize the amount of characters needed to be typed for the new one. There is no limit on the number of strings that can be kept in the history.

ESC p

Walk backward through the history of previously entered strings (‘previous-history-element’).

ESC n

Walk forward through the history of previously entered strings (‘next-history-element’).

3.1.8.6 Commands to set the mark

^SPC

Set the mark at the current point position (‘set-mark’).

^X ^X

Exchange the current point position with the mark one (‘exchange-point-and-mark’).

3.1.8.7 Reinserting recently killed text

^Y

Reinsert a previously killed text at the current point position (‘yank’).

3.1.8.8 Selecting files matching patterns

If the very first character in the input line is a ‘+’, what comes after it is considered a (space separated) list of shell patterns, and all the files that match at least one pattern from that list will be marked as selected. An empty list of shell patterns (i.e. the ‘+’ by itself) will cause all the files to be selected.

If the very first character in the input line is a ‘-’, the space separated list of shell patterns that follows is used to unselect files. An empty list of shell patterns (i.e. the ‘-’ by itself) will cause all the selected files to be unselected.

Finally, if the first and only character in the input line is a '*', then all the selected files will become unselected, and all the unselected files will become selected.

3.1.9 File operations

3.1.9.1 Copying Files

F5, *ESC 5*, *^C C*

Copy the currently selected entries to the user supplied path ('`copy`').

^C b C

Copy the currently selected entries to the user supplied path. The operation is performed in background ('`B-COPY`').

3.1.9.2 Moving Files

F6, *ESC 6*, *^C T*

Move the currently selected entries to the user supplied path ('`move`').

^C b T

Move the currently selected entries to the user supplied path. The operation is performed in background ('`B-MOVE`').

3.1.9.3 Creating Files

The easiest way to create a new file is to start an editor passing the file name as an argument. Most editors will try to create the file if the file doesn't exist. See Section 3.1.9.10 [Editing Files], page 14, for more information.

3.1.9.4 Deleting Files

F8, *ESC 8*, *^C D*

Delete the currently selected entries ('`delete`').

^C b D

Delete the currently selected entries. The operation is performed in background ('`B-DELETE`').

3.1.9.5 Linking Files

^C H

Create a hard link from the current files to a user supplied file name ('`LINK`').

^C b H

Create a hard link from the current files to a user supplied file name ('`B-LINK`'). The action is performed in background.

^C S

Create a symbolic link from the current files to a user supplied file name ('`SYMLINK`').

^C b S

Create a symbolic link from the current files to a user supplied file name (`B-SYMLINK`). The action is performed in background.

3.1.9.6 Renaming Files

^C R

Rename the current file or directory with the user supplied name (`RENAME`).

^C b R

Rename the current file or directory with the user supplied name. The operation is performed in background (`B-RENAME`).

^C n d

Change the name of all the selected entries to lowercase. (`name-downcase`).

^C n u

Change the name of all the selected entries to uppercase. (`name-upcase`).

3.1.9.7 Splitting files into smaller parts

^C /

Split the current file into several smaller files of a given size and named based on a given prefix (`SPLIT`).

^C b /

Split the current file into several smaller files of a given size and named based on a given prefix. The operation is performed in background (`B-SPLIT`).

3.1.9.8 Packing files into the minimum number of bins

^C ~

Pack the files into the smallest number of bins. This is an approximation - the problem is NP-complete and no known algorithm can guarantee a solution better than `(11/9) * OPTIMAL + 4`.

To make things even worse, for large files, there is no portable way to predict how many blocks the file system implementation will require for indirect blocks, directories, etc. So keep in mind that this is only an approximation.

Bin packing can be useful when you want to put a bunch of files on floppies or zip disks and you want to optimize things a little bit (`bin-packing`).

`gitfm` assumes that you want to pack all the files in the current directory - if there is any selected file in that directory it will be unselected first. Then `gitfm` will ask for a bin size, and select the files that should go in the first bin. You are supposed to place those files in the first bin (e.g. a tar archive), remove them from the current directory, then run 'bin-packing' again, to obtain the list of the files that should go into the second bin, etc.

3.1.9.9 Changing a file's mode, owner and group

^C M

Change the mode of the currently selected entries ('CHMOD').

^C b M

Change the mode of the currently selected entries. The operation is performed in background ('B-CHMOD').

^C r M

Recursively change the modes of the selected entries if one of them is a directory ('R-CHMOD').

^C b r M

Recursively change the modes of the selected entries if one of them is a directory. The operation is performed in background ('B-R-CHMOD').

^C O

Change the owner of the currently selected entries ('CHOWN').

^C b O

Change the owner of the currently selected entries. The operation is performed in background ('B-CHOWN').

^C r O

Recursively change the owners of the selected entries if one of them is a directory ('R-CHOWN').

^C b r O

Recursively change the owners of the selected entries if one of them is a directory. The operation is performed in background ('B-R-CHOWN').

^C G

Change the group of the currently selected entries ('CHGRP').

^C b G

Change the group of the currently selected entries. The operation is performed in background ('B-CHGRP').

^C r G

Recursively change the groups of the selected entries if one of them is a directory ('R-CHGRP').

^C b r G

Recursively change the groups of the selected entries if one of them is a directory. The operation is performed in background ('B-R-CHGRP').

3.1.9.10 Editing Files

F4, *ESC 4*

Call the default editor with the current file name as an argument ('EDIT').

^X e

Call the default editor with the selected entry names as arguments ('MULTIPLE-EDIT').

^X ^F

Create a new file by calling the default editor with the user supplied file name as an argument ('FILE-CREATE').

^X 4 a

Call the default editor in order to edit the 'ChangeLog' file ('CHANGE-LOG').

The default editor can be specified using the *EDITOR* or *GNUIT_EDITOR* environment variables. See Section 4.1 [Environment Variables], page 33, for more information.

$GNUIT_EDITOR used to be called *$GIT_EDITOR*. The old name is still accepted for backwards-compatibility.

3.1.9.11 Viewing Files

F3, *ESC 3*

Call the default viewer (`gitview`) with the current file name as argument ('VIEW').

^X v

Call the default pager (`more`) with the currently selected entry names as arguments ('MULTIPLE-VIEW').

3.1.9.12 Compressing Files

^C z

Compress the currently selected entries with `gzip` ('COMPRESS').

^C b z

Compress the currently selected entries with `gzip`. The operation is performed in background ('B-COMPRESS').

^C Z

Uncompress the currently selected entries with `gunzip` ('UNCOMPRESS').

^C b Z

Uncompress the currently selected entries with `gunzip`. The operation is performed in background ('B-UNCOMPRESS').

^C f Z

Uncompress the currently selected entries with `gunzip` (‘`F-UNCOMPRESS`’). Force uncompression of links.

^C b f Z

Uncompress the currently selected entries with `gunzip`. Force uncompression of links. The operation is performed in background (‘`BF-UNCOMPRESS`’).

^C .

Compress the currently selected entries with `bzip2` (‘`BZIP2-COMPRESS`’).

^C b .

Compress the currently selected entries with `bzip2`. The operation is performed in background (‘`B-BZIP2-COMPRESS`’).

^C o

Uncompress the currently selected entries with `bunzip2` (‘`BZIP2-UNCOMPRESS`’).

^C b o

Uncompress the currently selected entries with `bunzip2`. The operation is performed in background (‘`B-BZIP2-UNCOMPRESS`’).

3.1.9.13 Encoding Files

^C e

Encode the currently selected file (‘`UUENCODE`’).

^C b e

Encode the currently selected file. The operation is performed in background (‘`B-UUENCODE`’).

^C E

Decode the currently selected file (‘`UUDECODE`’).

^C b E

Decode the currently selected file. The operation is performed in background (‘`B-UUDECODE`’).

^C k

Encode the currently selected file using mpack (‘`MIME-PACK`’).

^C b k

Encode the currently selected file using mpack. The operation is performed in background (‘`B-MIME-PACK`’).

^C K

Decode the currently selected file using munpack (‘`MIME-UNPACK`’).

^C b K

Decode the currently selected file using munpack. The operation is performed in background (‘`B-MIME-UNPACK`’).

3.1.9.14 Encrypting Files

^C p

Encrypt (using `pgp`) the current file (‘`ENCRYPT`’).

^C P

Decrypt (using `pgp`) the current file (‘`DECRYPT`’).

3.1.9.15 Comparing Files

^C =

Compare (using `diff`) the current ASCII file with the other panel’s current file (‘`DIFF`’). If both entries are directories, a recursive diff is performed.

^C ESC =

Compare (using `diff`) the current ASCII file with its latest backup. The latest backup is the file having the same name and a ’~’ at the end (‘`LAST-BACKUP-DIFF`’).

^C B

Compare the current file with the other panel current file. A binary comparison is performed (‘`compare`’).

3.1.9.16 Spell Checking Files

^X I

Run the `ispell` command with the current file name as an argument.

3.1.9.17 Printing Files

^C j

Print the selected files via `lpr` to the user specified printer or to the default one if no `lpr` options are given (‘`PRINT-JOB`’).

^C J

Print the list of active printing jobs via `lpq`. The default printer is queried if no `lpq` options are given (‘`PRINT-JOB-LIST`’).

3.1.9.18 Wiping Files

^C W

Call `gitwipe` to wipe the selected files. Asks for confirmation before actually wiping them in order to avoid errors (‘`WIPE`’).

See Section 3.5 [gitwipe], page 31, for more information.

3.1.9.19 Searching Files

ESC %

Search files on the file system, starting from the current directory ('FIND').

ESC &

Use 'locate' to search files on the file system, starting from the current directory ('LOCATE').

^X w

Locate the binary, source, and manual page files for a command ('WHEREIS').

^X W

Locate a command; display its pathname or alias ('WHICH').

3.1.9.20 Managing tar based archive files

^C a

Create a `tar` archive containing all the currently selected entries ('TAR').

^C b a

Create a `tar` archive containing all the currently selected entries. The operation is performed in background ('B-TAR').

^C x

Create a compressed `tar` archive containing the current entry, provided it is a directory ('TAR-COMPRESS').

^C b x

Create a compressed `tar` archive containing the current entry, provided it is directory. The operation is performed in background ('B-TAR-COMPRESS').

^C -

Create a bzip2 compressed `tar` archive containing the current entry, provided it is a directory ('TAR-BZIP2').

^C b -

Create a bzip2 compressed `tar` archive containing the current entry, provided it is a directory. The operation is performed in background ('B-TAR-BZIP2').

^C X

Expand the selected archives into the current directory. The utility used is selected based on the extension of the file ('GENERIC-UNPACK').

^C b X

Expand the selected archives into the current directory. The utility used is based on the extension of the file. The operation is performed in background ('B-GENERIC-UNPACK').

^C V

Expand the selected archives into an user supplied current directory. The utility used is selected based on the extension of the file (‘GENERIC-UNPACK-INTO’).

^C b V

Expand the selected archives into an user supplied directory. The utility used is based on the extension of the file. The operation is performed in background (‘B-GENERIC-UNPACK-INTO’).

3.1.9.21 Working with DPKG

`dpkg` provides commands for manipulating and querying the dpkg database, `gitfm`’s DPKG commands allow quick access to most of those which operate on binary packages, files, or require package names as arguments. All commands expecting package names can also be given binary packages, the package names will be automatically extracted. Most commands use their single character `dpkg` option letter as the key command.

^C ^D i

Install the `deb` file(s) selected or pointed by the cursor (‘DPKG-INSTALL’).

^C ^D R i

Recursively install the `deb` file(s) in the directories selected or pointed by the cursor (‘R-DPKG-INSTALL’).

^C ^D u

Unpack the `deb` file(s) selected or pointed by the cursor, but don’t configure it (‘DPKG-UNPACK’).

^C ^D R u

Recursively unpack the `deb` file(s) in the directories selected or pointed by the cursor, but don’t configure them (‘R-DPKG-UNPACK’).

^C ^D C

Configure the unpacked package(s) selected or pointed by the cursor (‘DPKG-CONFIGURE’).

^C ^D r

Remove the package(s) selected or pointed by the cursor (‘DPKG-REMOVE’).

^C ^D P

Purge the package(s) selected or pointed by the cursor (‘DPKG-PURGE’).

^C ^D A

Update dpkg and dselect’s idea of which packages are available with information from the `deb` file(s) selected or pointed by the cursor (‘DPKG-RECORDAVAIL’).

^C ^D R A

Recursively update dpkg and dselect’s idea of which packages are available with information from the `deb` file(s) in the directories selected or pointed by the cursor (‘R-DPKG-RECORDAVAIL’).

^C ^D h

Display quick help file for GITFM’s DPKG commands. (‘DPKG-HELP’).

^C ^D c

Lists the contents of the filesystem tree archive portion of the `deb` file pointed by the cursor (‘DPKG-CONTENTS’).

^C ^D f

Extracts control file information from a `deb` file pointed by the cursor (‘DPKG-FIELD’). You are presented with the file name and can either hit ENTER to see all fields, or add control file field names (space separated) to see only those fields.

^C ^D I

Provides information about a `deb` file pointed by the cursor (‘DPKG-INFO’).

^C ^D l

List the package(s) selected or pointed by the cursor (‘DPKG-LISTPKGS’).

^C ^D s

Display status details for the package(s) selected or pointed by the cursor (‘DPKG-STATUS’).

^C ^D L

List files owned by the package(s) selected or pointed by the cursor (‘DPKG-LISTFILES’).

^C ^D S

Find package owning file pointed to by the cursor (‘DPKG-SEARCH’).

^C ^D p

Display available version details of package(s) selected or pointed by the cursor (‘DPKG-PRINTAVAIL’).

3.1.9.22 Installing and Uninstalling RPM packages

Red Hat Linux Distributions use a very powerful package manager called `rpm`. It is used whenever you need to install/remove/upgrade/etc a software package. The packages used by `rpm` use the extension ‘`.rpm`’. `gitfm` provides default key bindings for some of the basic operations `rpm` can perform on packages: install, uninstall, upgrade and query.

^C I

Install the `rpm` package pointed by the cursor (‘RPM-INSTALL’).

^C b I

Install the `rpm` package pointed by the cursor (‘`B-RPM-INSTALL`’). The operation is performed in background.

^C i

Uninstall the `rpm` package pointed by the cursor (‘`RPM-UNINSTALL`’).

^C b i

Uninstall the `rpm` package pointed by the cursor (‘`B-RPM-UNINSTALL`’). The operation is performed in background.

^C N

Upgrade the `rpm` package pointed by the cursor (‘`RPM-UPGRADE`’).

^C b N

Upgrade the `rpm` package pointed by the cursor (‘`B-RPM-UPGRADE`’). The operation is performed in background.

^C q

Query the `rpm` package manager (‘`RPM-QUERY`’). The default options used are ‘`-qil`’. See the `rpm` manual page for more details on using `rpm`.

3.1.9.23 File Types

^C t

Print the type of a file using the `file` utility (‘`FILE-TYPE`’).

3.1.9.24 Accessing MSDOS Files

^C m d, *^C m ^D*

Change the current `MSDOS` directory to a given directory (‘`MTOOLS-CHDIR`’).

^C m P

Put the currently selected files on a `MSDOS` floppy (‘`MTOOLS-PUT`’).

^C b m P

Put the currently selected files on a `MSDOS` floppy (‘`B-MTOOLS-PUT`’). The operation is performed in background.

^C m G

Get some files from a `MSDOS` floppy and put them in a given directory (‘`MTOOLS-GET`’).

^C b m G

Get some files from a `MSDOS` floppy and put them in a given directory (‘`B-MTOOLS-GET`’). The operation is performed in background.

^C m D

Delete files from a MSDOS floppy ('MTOOLS-DELETE').

^C b m D

Delete files from a MSDOS floppy ('B-MTOOLS-DELETE'). The operation is performed in background.

^C m L

List files and directories on a MSDOS floppy ('MTOOLS-DIR').

^C m F

Format a MSDOS floppy ('MTOOLS-FORMAT').

^C b m F

Format a MSDOS floppy ('B-MTOOLS-FORMAT'). The operation is performed in background.

^C m M

Create a directory on a MSDOS floppy ('MTOOLS-MKDIR').

^C b m M

Create a directory on a MSDOS floppy ('B-MTOOLS-MKDIR'). The operation is performed in background.

^C m K

Remove a directory from a MSDOS floppy ('MTOOLS-RMDIR'). All the files and subdirectories in that directory are removed as well.

^C b m K

Remove a directory from a MSDOS floppy ('B-MTOOLS-RMDIR'). All the files and subdirectories in that directory are removed as well. The operation is performed in background.

^C m R

Rename a file or directory on a MSDOS floppy ('MTOOLS-RENAME').

^C b m R

Rename a file or directory on a MSDOS floppy ('B-MTOOLS-RENAME'). The operation is performed in background.

^C m T

Display the contents of a file located on a MSDOS floppy ('MTOOLS-TYPE').

3.1.9.25 A different action for each file type

Many files on UNIX systems have one or more extensions specifying their types. For example, a file that ends in '.c' is a file containing a C program, while a file ending in '.tar.gz' is a tar archive compressed with the gzip utility. Having a default action for each file type, binded on the same key, seems to be a good idea because you can use that key to obtain type specific information about a file or to process it in some type specific way much easier. The GNUIT package contains a script called gitaction that is used to detect the current file type and perform a type specific action. See Section 3.7 [gitaction], page 31, for more information.

F2, *ESC 2*, *^X a*

Perform an action on the current file, depending on its type (‘FILE-ACTION’).

3.1.10 Directory operations

3.1.10.1 Creating directories

F7, *ESC 7*, *^X M*

Create a new subdirectory in the current directory with the user supplied name (‘make-directory’).

3.1.10.2 Copying directories

F5 (for directories), *ESC 5 (for directories)*, *^C C (for directories)*

Copy the currently selected entries to the user supplied path (‘copy’).

^C b C (for directories)

Copy the currently selected entries to the user supplied path. The operation is performed in background (‘B-COPY’).

3.1.10.3 Deleting directories

F8 (for subdirectories), *ESC 8 (for directories)*, *^C D (for subdirectories)*

Delete the currently selected entries (‘delete’).

^C b D (for directories)

Delete the currently selected entries. The operation is performed in background (‘B-DELETE’).

3.1.10.4 Moving directories

F6 (for directories), *ESC 6 (for directories)*, *^C T (for directories)*

Move the currently selected entries to the user supplied path (‘move’).

^C b T (for directories)

Move the currently selected entries to the user supplied path. The operation is performed in background (‘B-MOVE’).

3.1.10.5 Renaming directories

^C R (for directories)

Rename the current file or directory with the user supplied name (‘RENAME’).

^C b R (for directories)

Rename the current file or directory with the user supplied name. The operation is performed in background (‘B-RENAME’).

3.1.10.6 Comparing Directories

^C c q

Quickly compare the files in the left panel with the files in the right one. Only the file names, sizes and time stamps are considered in the comparison.

^C c t

Compare the files in the left panel with the files in the right one. The contents of each file in the current panel will be compared against the contents of its counterpart (if any) from the other panel.

^C f d

Recursively compare (using `diff -r -q`) the current directory with the other panel's current directory ('`FAST-DIFF`'). Print on standard output the names of the files that differ.

^C d

Compare (using `diff`) the current directory with the other panel current directory. For successful operation, both panels should contain the same directory ('`DIR-DIFF`').

^C b d

Compare (using `diff`) the current directory with the other panel current directory. For successful operation, both panels should contain the same directory. The operation is performed in background ('`DIR-DIFF`').

3.1.10.7 Summarize directory usage

^C U

Display the output of the `du -s` command on the status line ('`DIRECTORY-USAGE`').

3.1.10.8 Changing directories

^X d, *^X ^D*

Change the current working directory. The user is asked for a new directory name and the new directory is added to the directory history ('`change-directory`').

See Section 3.1.10.9 [Dirs History], page 24, for more information.

ESC a *c*

Change the current directory of the current panel to the directory of the other panel ('`adapt-current-directory`').

ESC a *o*

Change the current directory of the other panel to the directory of the current panel ('`adapt-other-directory`').

3.1.10.9 Directory History

Users usually work on a limited set of subdirectories. Providing a fast method of switching between a number of intensively used directories is a good idea and `gitfm` has a set of builtin commands for doing it.

Usually new directories are added to the directory history when the '`change-directory`' built-in command is used. `gitfm` also adds the current directory to the history list when started, when the directory history is reset and when a command having a non empty '`new-dir`' field successfully completes its execution. See Section 4.2.3.22 [new-dir], page 38, for more information.

^X ^N

Go to the next directory in the history ('`next-directory`').

^X ^P

Go to the previous directory in the history ('`previous-directory`').

^X ^R

Reset the entire directory history. As explained above, the current directory becomes the only directory in the history ('`reset-directory-history`').

3.1.10.10 Hot Keys

`gitfm` provides default key bindings for switching to a number of important directories as "/", "..", "$HOME", etc.

ESC /

Go to the '`/`' directory ('`ROOT-DIR`').

ESC .

Go to the '`..`' directory ('`up-one-dir`').

ESC h

Go to the '`~`' ($HOME) directory ('`HOME-DIR`').

ESC i

Go to the '`/usr/include`' directory ('`INCLUDE-DIR`').

ESC ESC 1

Go to the '`/mnt/fd0`' directory ('`FIRST-FLOPPY-DIR`').

ESC ESC 2

Go to the '`/mnt/fd1`' directory ('`SECOND-FLOPPY-DIR`').

3.1.11 Compiling programs

F9, *ESC 9*, *^X m*

Run the `make` command in the current directory. Use -k as the default option ('`MAKE`').

^X b m

Run the `make` command in background in the current directory ('`B-MAKE`').

See Section 3.7 [gitaction], page 31, for more information.

3.1.12 Sending/receiving ascii/binary mail

^C 2 a

Send the current current ascii file by mail to an user supplied email address ('ASCII-MAIL').

^C b 2 a

The same as 'ASCII-MAIL', the only difference being that the command runs in background ('B-ASCII-MAIL').

^C 2 b

Send the current current binary file by mail to an user supplied list of email addresses. The file is uuencoded first ('BINARY-MAIL').

^C b 2 b

The same as 'BINARY-MAIL', the only difference being that the command runs in background ('B-BINARY-MAIL').

^C 2 m

Send the current current binary file by mail to an user supplied list of email addresses. The file is encoded with mpack first ('MIME-MAIL').

^C b 2 m

The same as 'MIME-MAIL', the only difference being that the command runs in background ('B-MIME-MAIL').

ESC x r m

Run the `emacs -f rmail` command. This will start the Emacs's 'rmail' function so that you can read your mail ('READ-MAIL').

3.1.13 Starting a sub-shell

^X z

Call a sub-shell as specified by the *$GNUIT_SHELL* environment variable ('SUB-SHELL').

$GNUIT_SHELL used to be called *$GIT_SHELL*. The old name is still accepted for backwards-compatibility.

See Section 4.1 [Environment Variables], page 33, for more information.

3.1.14 Using grep and recursive grep

^X g

Search using `grep` all the selected files for a given pattern ('GREP').

^X g

Search recursively using `gitrgrep` all the user specified files and directories for a given pattern ('RECURSIVE-GREP').

See Section 3.9 [gitrgrep], page 32, for more information.

3.1.15 Locking your console

Having a lock feature might be a good idea and, since not all the `UNIX` systems provide one, `gitfm` tries to get around the problem . . .

^X p

Prompt the user for a password and locks the console until the same password is reinserted (‘`lock`’).

3.1.16 Refreshing the screen contents

Sometimes your screen needs to be refreshed. Just think about what happens when somebody wants to talk with you and the talk daemon writes something like this

Message from Talk_Daemon@galei.cs.vu.nl at 12:15 ...
talk: connection requested by andrei@galei.cs.vu.nl.
talk: respond with: talk andrei@galei.cs.vu.nl

on your screen. And sometimes you might also want to re-read the current directories. `gitfm` provides a built-in command for refreshing the screen contents.

^L

Re-read the directories contents and refresh the screen (‘`refresh`’).

3.1.17 Resetting your terminal

^X ^L

Call `reset` in order to reset the terminal to its default settings (‘`TTY-RESET`’).

3.1.18 Mounting/unmounting file systems

People dealing with lots of files usually need to save/restore/copy files from/to other file systems. In order to be more efficient, `gitfm` provides a set of key bindings for mounting and unmounting file systems. See Section 3.6 [gitmount], page 31, for more information.

The default key bindings set has been designed to work under `Linux`, but it can be easily changed for other `UNIX` systems with different device names. Reading the configuration file ‘`gnuitrc.common`’ should be enough. See Section 3.1.10.10 [Hot Keys], page 24, for more information.

As a convention, the ‘`/mnt`’ directory is used to store an empty subdirectory for each mountable file system. Each file system is actually mounted in its counterpart ‘`/mnt`’ subdirectory. Try to follow this convention since the `gitmount` script is heavily based on it. See Chapter 4 [Customization], page 33, for more information.

ESC m `a`

Call `mount`(1) in order to mount the first floppy (‘`/dev/fd0`’) in the ‘`/mnt/fd0`’ directory (‘`MOUNT-A`’).

ESC m b

Call `mount`(1) in order to mount the second floppy (‘`/dev/fd1`’) in the ‘`/mnt/fd1`’ directory (‘`MOUNT-B`’).

ESC m c

Call `mount`(1) in order to mount the cdrom
('`/dev/cdrom`') in the '`/mnt/cdrom`' directory ('`MOUNT-CDROM`').

ESC m f

Call `mount`(1) in order to mount the first floppy
('`/dev/fd0`') in the '`/mnt/floppy`' directory ('`MOUNT-FLOPPY`').

ESC m z

Call `mount`(1) in order to mount the zip drive
('`/dev/zip`') in the '`/mnt/zip`' directory ('`MOUNT-ZIP`').

ESC m j

Call `mount`(1) in order to mount the jaz drive
('`/dev/jaz`') in the '`/mnt/jaz`' directory ('`MOUNT-JAZ`').

ESC m t

Call `mount`(1) in order to mount the file systems corresponding to the selected subdirectories. For example, if you are in the '`/mnt`' directory and the '`cdrom`' and '`zip`' subdirectories are selected, the cdrom and the zip disk will be mounted ('`MOUNT-THESE`').

ESC r a

Call `umount`(1) in order to remove (unmount) the first floppy
('`/dev/fd0`') ('`UMOUNT-A`').

ESC r b

Call `umount`(1) in order to remove (unmount) the second floppy
('`/dev/fd1`') ('`UMOUNT-B`').

ESC r c

Call `umount`(1) in order to remove (unmount) the cdrom
('`/dev/cdrom`') ('`UMOUNT-CDROM`').

findex UMOUNT-CDROM

ESC r f

Call `umount`(1) in order to remove (unmount) the first floppy
('`/dev/fd0`') ('`UMOUNT-FLOPPY`').

ESC r z

Call `umount`(1) in order to remove (unmount) the zip drive
('`/dev/zip`') ('`UMOUNT-ZIP`').

ESC r j

Call `umount`(1) in order to remove (unmount) the jaz drive
('`/dev/jaz`') ('`UMOUNT-JAZ`').

ESC r t

Call `umount`(1) in order to remove (unmount) the file systems mounted into the selected subdirectories. For example, if the current directory is '`/mnt`' and the '`cdrom`' and '`zip`' subdirectories are selected, the cdrom and the zip disk will be unmounted ('`UMOUNT-THESE`').

3.1.19 Getting some useful system information

^X T

Call `date`(1) in order to display the current time/date ('`DATE`').

ESC S f

Call `finger`(1) in order to display information about local and remote users ('`FINGER`').

ESC S m

Call `mount`(1) in order to display a list of the currently mounted file systems ('`MOUNTED-FILE-SYSTEMS`').

ESC S q

Call `quota`(1) in order to display a user file system disk quota and quota ('`QUOTA`').

ESC S s

Call `df`(1) in order to get the status of the currently mounted file systems ('`DISK-FREE-SPACE`').

ESC S u

Call `users`(1) in order to get the name of the currently logged in users ('`USERS`').

ESC S v

Call `$GNUIT_VMSTAT`(1) in order to get the current virtual memory status. This is very system dependent, `Linux` uses `free`, other systems use `vmstat`, so the *$GNUIT_VMSTAT* variable is used to deal with this ('`VIRTUAL-MEMORY-STATUS`').

$GNUIT_VMSTAT used to be called *$GIT_VMSTAT*. The old name is still accepted for backwards-compatibility.

See Section 4.1 [Environment Variables], page 33, for more information.

ESC S w

Call `who`(1) in order to find out who is on the system ('`WHO`').

3.1.20 How to look at the environment variables

^X E

Call `env`(1) in order to display the current environment ('`ENV`').

^X H

Call `xhost`(1) in order to add/remove hosts names to the list allowed to make connection to the X server ('`XHOST`').

3.1.21 Viewing/killing processes

There are at least two kinds of `ps`(1) utilities. One that accepts (more or less) combinations of the 'a', 'u', and 'x' flags and another that accepts combinations of 'e', 'f' and 'l' flags. Since is quite difficult to test which one works fine on a given `UNIX` system, `gitfm` provides key bindings for both of them. Anyway, if your `ps`(1) fails to accept the predefined combinations, please take a look in its manual and then modify the '`.gnuitrc.TERM`' file as needed.

Since the number of possible combinations of flags in the `ps` command line is quite big and *very* system dependent, there is no real reason to display them all here. We are only interested in giving you a starting point in your search through the '`.gnuitrc.TERM`' file.

Note also that you can display a list of processes using `ps`(1) or browse through a list of them (killing as needed) using `gitps`. As a convention, we have used the same key sequence for a given set of `ps`(1) flags for both `ps`(1) and `gitps`, the only difference being that `ps`(1) keys end in an uppercase letter. See Section 3.2 [gitps], page 30, for more information.

Under Linux it is possible to see a tree of processes using `pstree`(1).

Here there are the default key bindings for the 'e', 'f' and 'l' `ps`(1) flags combinations:

ESC P b, *ESC P c*, *ESC P e*

Call `gitps` or `ps`(1) in order to browse through or display a list of currently running processes ('`GITPS`', '`PS`').

... and the default key bindings for the 'a', 'u' and 'x' `ps`(1) flags combinations:

ESC P a, *ESC P l*, *ESC P u* *ESC P x*, *ESC P y*

Call `gitps` or `ps`(1) in order to browse through or display a list of currently running processes ('`GITPS`', '`PS`').

ESC P T

Call `pstree`(1) in order to display the tree of currently running processes ('`PSTREE`').

^X k

Call `kill`(1) in order to kill a user specified process with a given signal ('`KILL`').

3.1.22 Synchronizing the file systems

^X S

Call `sync`(1) in order to synchronize all the file systems ('`SYNC`').

3.1.23 Reading the documentation

^X q

Read a manual page. The user is prompted for its name ('`MAN`').

F1, *ESC 1*, *^X i*

Read an info documentation. The user is prompted for the documentation name ('`INFO`').

^X h

Read the html documentation using the viewer specified in *GNUIT_BROWSER*, or with lynx if *GNUIT_BROWSER* is not set ('HTML').

$GNUIT_BROWSER used to be called *$GIT_BROWSER*. The old name is still accepted for backwards-compatibility.

^X ^A

Show the key-bindings for commands matching a specified string ('apropos').

3.1.24 Exiting GNU Interactive Tools

F10, *ESC 0*, *^X ^C*, *^X c*

Exit GNU Interactive Tools ('exit').

3.2 The GNUIT process viewer/killer

gitps is an interactive process viewer/killer. It calls internally the ps(1) utility. This is a brief description of the command line arguments.

-h print this help message

-v print the version number

-i print the installation directory

-c use ANSI colors

-b don't use ANSI colors

-l don't use the last screen character

-p pass the remaining arguments to ps(1)

Running gitps is self explanatory. Use the *arrows*, *PageUp*, *PageDown*, *Home*, *End*, *^N*, *^P*, *^V*, *ESC v*, *Space* and *Backspace* to move in the list, *^L* to refresh it, *Enter* to change the default signal and *F10*, *q* or *^X ^C* to leave.

You can change these keys, just read the GITPS-Setup, GITPS-Color, GITPS-Monochrome and GITPS-Keys sections in the configuration files 'gnuitrc.TERM'.

The selected signal can also be changed by pressing its first letter (in uppercase), e.g. for 'SIGTERM' press *T*, or by pressing the key corresponding to the signal number. For signals > 10, prefix with *^X*, and for signals greater than 20, prefix with *^C*. For instance, to select signal 15 ('SIGTERM'), press *^X5*.

Keys for signal numbers are hard-coded to the versions in i386 Linux, however most correspond to POSIX. Compare signal(7) or signal(5) with http://linux.die.net/man/7/signal.

3.3 The GNUIT ASCII/HEX file viewer

gitview is an ASCII/HEX file viewer. Use the *arrows*, *PageUp*, *PageDown*, *Home*, *End*, *^N*, *^P*, *^V*, *ESC v*, *Space* and *Backspace* to move in the file, *^L* to refresh the screen and *F10*, *q* or *^X ^C* to leave.

You can change these keys, just read the GITVIEW-Setup, GITVIEW-Color, GITVIEW-Monochrome and GITVIEW-Keys sections in the configuration files '`gnuitrc.TERM`'.

Here is a brief description of the command line arguments:

`-h` print this help message

`-v` print the version number

`-i` print the installation directory

`-c` use ANSI colors

`-b` don't use ANSI colors

`-l` don't use the last screen character

3.4 The GNUIT key sequences display utility

`gitkeys` is a program that displays the key sequence sent by the pressed key. This is the key sequence received by `GIT` tools, so this program is useful when setting up the '`.gnuitrc.TERM`' configuration files.

3.5 The GNUIT wipe file utility

`gitwipe` is an utility for wiping files. It overwrites the file contents with a random sequence of numbers and then calls '`sync`'().

Note that `gitwipe` does *not* remove the wiped file since (under `Linux` at least) the '`sync`'() system call might return before actually writing the new file contents to disk. Removing the file might be dangerous because some file systems can detect that the blocks in the removed wiped file are no longer used and never write them back to disk in order to improve performance. It is up to you to remove the file(s) at a later moment.

3.6 The GNUIT mount utility

`gitmount` is a script that allows you to mount a list of block devices (specified in the command line), without specifying the file system type. With a command like 'gitmount fd0 cdrom' the first floppy will be mounted in '`/mnt/fd0`' and the cdrom will be mounted in '`/mnt/cdrom`'. Make sure your '`/etc/fstab`' settings are correct.

You don't need to know the file system type anymore. If you want to use `gitmount` with the block device '`/dev/xxx`' then the directory '`/mnt/xxx`' is created if it doesn't exist. `gitmount` will attempt to create the necessary directories, but root permissions might be required.

3.7 The GNUIT per file type action script

`gitaction` is a script that executes a different action for each file type specified. It is called by the `gitfm` program when pressing *F2*, *ESC 2* or *^Xa*.

The first parameter is the current directory name and the second one is the file name to be matched against the default patterns. The matching is done using the shell 'case' statement.

If you press *F2*, *ESC 2* or *^Xa* on a '`*.html`' file, `gitfm` will invoke a browser to view it, if you press *F2*, *ESC 2* or *^Xa* on a '`*.tar.gz`' file, `gitfm` will list the tar archive contents, if you press the same keys on a '*.gz' file, gitfm will display its uncompressed contents on the screen, etc ...

If you press *F2*, *ESC 2* or *^Xa* on a '*.gif' file or '*.jpg' file and you have the xzgv utility installed, you will be able to see it. If you want to change the gif/jpeg viewer, all you need to do is to change its name in the gitaction script.

There are many more file types and viewers that gitaction knows about. In addition, if you are running under GNOME, gnome-open is used, and on MacOS, open(1) is used. see(1) and metamail(1) are also used as fallbacks. If all else fails, the file is displayed using $GNUIT_PAGER.

If you want to find out what the default action for each file type is (or if you want to modify it), just read/modify the gitaction script.

Also, you can add a .gitaction shell script in your home directory and/or in any other directory. Before trying to match a file name, gitaction will attempt to execute ./.gitaction. If that one fails to match the file name against its patterns, it backs up to $HOME/.gitaction. When this one fails too the patterns in gitaction are tried. For an example of how to write .gitaction scripts take a look at the .gitaction shell script provided as part of the distribution and installed in the '$(prefix)/bin' directory. Debian users will find it in '/usr/share/doc/gnuit/examples/'.

3.8 Unified archive unpacking

gitunpack is a shell script that accepts a directory and a set of archives as its command line parameters, and then attempts to unpack those archives in the given directory, selecting the utility used to unpack the archives based on the archive extensions.

3.9 The GIT recursive grep script

gitrgrep is a very small script that calls grep recursively. It accepts grep like options / parameters, the only difference being that file specifications should be quoted:

```
gitrgrep main '*.c'

or

gitrgrep errno '*.c *.h'
```

gitregrep and gitrfgrep are recursive versions of the egrep and fgrep programs.

4 Customizing GNU Interactive Tools

4.1 Environment Variables

The configuration files use shell environment variables to call the shell, editor, mail reader, html viewer, compress and virtual memory status utility. That means that if you set *GNUIT_SHELL*, *GNUIT_EDITOR*, *GNUIT_RMAIL*, *GNUIT_BROWSER*, or *GNUIT_VMSTAT* to some value, that value will be used instead of the default one. The defaults are:

GNUIT_SHELL='/bin/sh'
GNUIT_EDITOR='vi'
GNUIT_RMAIL='emacs -f rmail'
GNUIT_PAGER='more
GNUIT_VMSTAT='free'
GNUIT_BROWSER='lynx'

if the `configure` script is passed '`--enable-debian`', some of the defaults are changed as follows:

GNUIT_EDITOR='sensible-editor'
GNUIT_PAGER='sensible-pager'
GNUIT_BROWSER='sensible-browser'

If *SHELL* is defined, *GNUIT_SHELL* will be set to that value. If *PAGER* is defined, *GNUIT_PAGER* will be set to that value. If *EDITOR* is defined, *GNUIT_EDITOR* will be set to that value. If you want to change the default settings, put something like this into your '`.profile`':

export GNUIT_SHELL='/usr/local/bin/bash'
export GNUIT_EDITOR='emacs'
export GNUIT_RMAIL='elm'
export GNUIT_PAGER='less'
export GNUIT_VMSTAT='vmstat'
export GNUIT_BROWSER='netscape'

These variables used to be prefixed with *GIT_* (e.g. *GIT_PAGER*). The old names are still accepted for backwards compatibility.

4.2 Configuration Files

There is one configuration file per terminal type in `GNUIT`. The configuration file(s) reside in the user's home directory or (the default versions) in the directory '`/usr/share/gnuit/`'). Files in the user's home directory start with a leading "."

Their generic name is '`gnuitrc.TERM`'. `GNUIT` allows each terminal type to have its own configuration file (*TERM* is the value of the *TERM* environment variable (e.g '`vt102`'); for the `Linux` console the configuration file is '`gnuitrc.console`').

Since most of the key bindings are common to all the terminal types, a configuration file called '`gnuitrc.common`' is parsed before parsing the normal '`.gnuitrc.TERM`' configuration file, the later one defining only those keys that are terminal specific. However, if a key binding is redefined in the '`gnuitrc.TERM`' file, that binding will be used.

If the GNUIT package have been compiled without passing the '--enable-terminfo' option to the configure script and your system has a huge 'termcap' database ('/etc/termcap'), you can copy the termcap definition(s) of your terminal(s) in a file called, lets say '.termcap' and put it in your home directory. After that, set your *TERMCAP* environment variable to point to it. You should add something like this to your '.profile':

TERMCAP='/home/mike/.termcap'

The interactive programs in the GNUIT package can run without such a file, but on systems with huge 'termcap' databases, copying the definitions of the most used terminals in a local '.termcap' file will lead to a faster start.

The '.gnuitrc.TERM' is first searched for in the home directory then, if not found, in the directory '/usr/share/gnuit/'). (without the leading "."). The configuration file is structured in sections, each section containing variables in the following format:

'variable-name' = 'first-field';'second-field'; ...

After the 'variable-name' at least one space or tab is required. All characters after a '#' are ignored and if you comment a section name, the whole section is ignored.

Section names are enclosed in rectangular brackets ('[' and ']'). Note that this manual don't include them while refering to section names.

The GNUIT package contains three major programs: gitfm, gitps and gitview. Each one has its own sections in the configuration files. There is also a global setup section called 'Setup' that is used by all these programs.

Note that now git has been renamed to gitfm, the corresponding sections have been renamed to GITFM, however for backwards compatibility, the old name (GIT) is still checked if the GITFM section is not found.

4.2.1 Writing key sequences

GNUIT contains three interactive programs. Their names are: gitfm (this is the file system browser), gitps (this is the process viewer/killer and gitview (this is the ASCII/HEX file viewer). Each one of these programs has its own set of key bindings.

The convention used in describing key bindings are very simple. Here there are some examples that will help you to understand them. The corresponding Emacs conventions will help you even more.

^A means keeping the Ctrl key down and pressing the a key (*C*-a).

The *ESC* character is represented as *^[* so that you can use the meta character (*M-*) where available (or the *ESC* key):

*^[*a corresponds to *M*-a (pressing the *ESC* key and then a).

The ^ character is represented as ^^.

The *backspace* character is represented as ^_.

The *Ctrl-SPACE* character (*C-SPC*) is represented as *^$*.

The space (*SPC*) character is represented as ^@.

Note that the key bindings notation described here is only used in the configuration files. For the sake of readability this manual uses *ESC* for the *ESC* key, *SPC* for the *SPACE* key and *RET* for the *RETURN* (*ENTER*) key.

4.2.2 The global setup section

In this section the variables have only one field.

'AnsiColors'

This variable should be set to 'ON' if the terminal supports standard 'ANSI' color sequences. Otherwise it should be 'OFF'. If 'AnsiColors' is 'ON', 'GITxxx-Color' sections will be used in the configuration files 'gnuitrc.TERM'. Otherwise, GNUIT interactive programs will use the 'GITxxx-Monochrome' sections.

'UseLastScreenChar'

This variable is used for terminals that can't write on the last character of the screen without scrolling the entire screen. If your terminal has no problem writing there (Linux console, vt100, vt102, xterm, ...) set it to 'ON'. Otherwise (hpterm), it should be 'OFF'.

'StartupScrollStep'

This variable specifies the scroll step initial value for both panels.

4.2.3 gitfm Sections

4.2.3.1 gitfm Setup

In this section the variables have only one field.

'StartupFileDisplayMode'

This variable specifies the file specific information displayed at startup. It can be any of 'OwnerGroup', 'DateTime', 'Size', 'AbbrevSize', 'Mode' or 'FullName'. Its value initially affects both panels but it can be changed separately afterward.

'StartupFileSortMethod'

This variable specifies the startup sort method. It can be any of 'Name', 'Extension', 'Size', 'Date', 'Mode', 'OwnerId', 'GroupId', 'OwnerName' or 'GroupName'. Its value initially affects both panels but it can be changed separately afterward.

'MaxUnscaledDigits'

Maximum number of digits a number may be before being scaled (e.g. to '123M'). If you want number to always be scaled, either set this to '0', or set 'StartupFileDisplayMode' to 'AbbrevSize'. Note that scaling may happen anyway if the number is too large for the display field.

'GroupDigits'

If this variable is 'ON', digits of file sizes will be grouped according to your locale, (e.g. '123,456,789').

'ConfirmOnExit'

If this variable is 'ON', the user is prompted for confirmation at exit.

'HistoryFile'

This variable specifies the history file name. The default value is '~/.githistory'.

'InfoDisplay'

If this variable is 'OFF', auxiliary file informations are not displayed. This can be useful if you are using a very slow terminal.

'LeadingDotMatch'

If this variable is '`OFF`' when matching files for select-files-matching-pattern / unselect-files-matching-pattern then the leading '.' in the file name is matched only explicitly.

'`TypeSensitivity`'

If this variable is '`OFF`', colors are not used when displaying files. Normally, the information in the '`GITFM-FTI`' section is used to display files with different colors, depending on their types. Note that '`TypeSensitivity`' is automatically set to '`OFF`' when '`AnsiColors`' is '`OFF`'. See Section 4.2.4 [GITFM-FTI], page 39, for mor information.

'`NormalModeHelp`' '`CommandLineModeHelp`'

These variables describe the status bar contents for each `gitfm` mode when no errors occurred. `gitfm` can display on the status bar a help string and/or some system information (system type, hostname, machine type and the current date) using escape characters:

```
\s  ->  the system type
\h  ->  the host name
\m  ->  the machine type
\d  ->  the current date
```

See Section 3.1.3 [Modes], page 4, for more information.

4.2.3.2 Using gitfm on color displays

In this sections the variables have only one field.

These section allows you to customize the colors of `gitfm`. Reading the '`gnuitrc.TERM`' configuration file is self explanatory.

4.2.3.3 Using gitfm on monochrome displays

In this sections the variables have only one field.

These section allows you to customize the appearance of `gitfm` on monochrome displays. Reading the '`gnuitrc.TERM`' configuration file is self explanatory.

4.2.3.4 Defining keys

These section describes the actions `gitfm` takes when a specified key is pressed. A variable can have up to 6 fields separated by ';'. Each line in this section looks like:

```
'key-sequence' = 'command-name';'formatted-command';'new-dir';
    'save-screen';'pause';'hide'
```

Note that you can't continue the variable fields description on the next line.

4.2.3.5 The key-sequence field

'`key-sequence`' is the key sequence associated with the given command. You can use any key sequence that doesn't start with an ascii character (0x20 to 0x7e).

Symbolic key names (*F0*, *F1*, *F2*, ... *F10*, *UP*, *DOWN*, *RIGHT*, *LEFT*, *INS*, *DEL*, *HOME*, *END*, *PGUP* and *PGDOWN*) can be used instead of the key sequence. If some keys don't have a '`termcap`'/ '`terminfo`' description (like the *F11*/*F12* keys on the `Linux` console) you can specify the key sequence in the usual way.

4.2.3.6 The command-name field

'command-name' is a command generic name. Even if it is not always used, the 'command-name' must be present (if a command is associated with a 'key-sequence'). If it is not, no action will be taken when pressing 'key-sequence'.

There are two types of commands in gitfm: built-in commands and user defined commands. If the 'command-name' section contains a built-in command specification, the other fields are ignored.

Note that by convention built-in command names contain only lower case letters while user defined command names contain only upper case letters.

4.2.3.7 The formatted-command field

'formatted-command' is a shell command which can contain some scanf(3)-like format specifiers. They are used to get the current entry name, owner, group, mode, etc.

Note that using uppercase 'format specifiers' you will be able to access the other panel path, file and directory names, etc.

These are the available 'format specifiers':

4.2.3.8 The %s format specifier

The format of %s is: %s{question,default_answer}.

When gitfm encounters a %s in the 'formatted-command' it asks the user the question 'question' whose default answer is 'default_answer' and replaces the '%s{ , }' with the user's answer. Both 'question' and 'default_answer' can contain any other 'format specifiers' except %s.

Note that there should be no spaces between %s and '{'.

4.2.3.9 The %f format specifier

gitfm will replace %f with the current directory entry name only if it is a file (not a directory).

4.2.3.10 The %d format specifier

gitfm will replace %d with the current directory entry name only if it is a directory (not a file).

4.2.3.11 The %l format specifier

gitfm will replace %l with the current directory entry name only if it is a symbolic link with no target.

4.2.3.12 The %t format specifier

gitfm will replace %t with the current directory entry name only if it is a named pipe.

4.2.3.13 The %z format specifier

gitfm will replace %z with the current directory entry name only if it is a socket.

4.2.3.14 The %a format specifier

gitfm will always replace %a with the current directory entry name.

4.2.3.15 The %m format specifier

`gitfm` will always replace %m with the current file mode.

4.2.3.16 The %g format specifier

`gitfm` will always replace %g with the current file group.

4.2.3.17 The %o format specifier

`gitfm` will always replace %o with the current file owner.

4.2.3.18 The %p format specifier

`gitfm` will always replace %p with the current panel path.

4.2.3.19 The %b format specifier

`gitfm` will always replace %b with the current panel directory name.

4.2.3.20 The %i format specifier

`gitfm` will always replace %i with all the current panel selected entry names.

4.2.3.21 The %? format specifier

The format of %? is: %?{confirmation}.

`gitfm` uses this format specifier only to ask for confirmation before expanding / executing the current command. The '`confirmation`' string is displayed and, if the user doesn't confirm, the command is aborted. Otherwise, %?{confirmation} expands to a null string and the command is expanded / executed normally.

4.2.3.22 The new-dir field

If the '`formatted-command`' successfully exits (exit code = 0) or it has no body and this field is present then '`new-dir`' will become the current panel directory.

The character '~' used at the beginning of the '`new-dir`' field is replaced by the user's home directory.

4.2.3.23 The save-screen field

This field is a character (usually 'y' or 'n') that tells `gitfm` to save ('y') or not to save ('n') the terminal's screen after executing the '`formatted-command`'. Saving the screen is not necessary while editing or viewing a file because the information left after the editor or the viewer exits is not important. Saving the screen means that that screen will be restored before the execution of the next command. Currently this field is used only if you are working as a super user under `Linux` on a virtual console. Its default value is 'y'.

4.2.3.24 The pause field

Users may wish to read the result of some commands before repainting the panels. If this field is present gitfm will wait for a key to be pressed before restoring the panels. Its default value is 'n'.

4.2.3.25 The hide field

Some commands that don't displaying any useful information if successfully complete their execution: `mount`, `chmod`, `chown`, `chgrp`, `sync` ... and, if an error occurs, a line or two are sent to stderr. If this option is 'y', the stdout and stderr will be redirected to some files ('`git.1.pid`' and '`git.2.pid`', where pid is `gitfm`'s pid) and only if the command's exit code is not 0, the '`git.2.pid`' file will be displayed, line by line, onto the status bar. This way the panels will not be deleted and then repainted and the command appears to be built-in. '`git.1.pid`' and '`git.2.pid`' are created in the temporary directory specified in the `TMPDIR` environment variable (or "/tmp" if `TMPDIR` is not defined). The default value of the `hide` field is 'n'.

4.2.4 Setting up colors for different file types

This sections contains entries of the form:

```
'pattern' = 'foreground'; 'background'; 'brightness'
```

where '`pattern`' is a file name matching pattern, '`foreground`', '`background`' and '`brightness`' are the color specification to be used when a file whose name match the given '`pattern`' is displayed in a panel. Colors can be turned off using the '`TypeSensitivity`' variable in the '`GITFM-Setup`' section.

4.2.5 gitps Sections

4.2.5.1 gitps Setup

In this section the variables have only one field.

'`Help`'

This variable describe `gitps`'s status bar contents.

4.2.5.2 Using gitps on color displays

In this sections the variables have only one field.

These section allows you to customize the colors of `gitps`. Reading the '`gnuitrc.TERM`' configuration file is self explanatory.

4.2.5.3 Using gitps on monochrome displays

In this sections the variables have only one field.

These section allows you to customize the appearance of `gitps` on monochrome displays. Reading the '`gnuitrc.TERM`' configuration file is self explanatory.

4.2.5.4 Defining keys

4.2.6 gitview Sections

4.2.6.1 gitview Setup

In this section the variables have only one field.

'`Help`'

This variable describe `gitps`'s status bar contents.

4.2.6.2 Using gitview on color displays

In this sections the variables have only one field.

These section allows you to customize the colors of `gitview`. Reading the 'gnuitrc.TERM' configuration file is self explanatory.

4.2.6.3 Using gitview on monochrome displays

In this sections the variables have only one field.

These section allows you to customize the appearance of `gitview` on monochrome displays. Reading the 'gnuitrc.TERM' configuration file is self explanatory.

4.2.6.4 Defining keys

5 GNU Interactive Tools limitations

Background commands (& terminated)can be specified in the configuration file but their result (stdout and stderr redirection), will be overwritten by the result of newer commands and, if an error occurs, it will not be seen.

When `gitfm` is compiled for `Linux`, the default built-in color descriptions are for color monitors, so you can't (decently) run `gitfm` on a b/w monitor without the '`gnuitrc.TERM`' file correctly configured. '`gnuitrc.TERM`' should be configured with '`AnsiColors`' = OFF. However, if your system knows about the linux-m terminal type, using that may be a better solution.

Job support is implemented only in `gitfm`.

Due to the fact that the ';' character is used as a field separator in the configuration files, you can't write something like that in the '`gnuitrc.TERM`' files:

^AAA = SHOW-USERS-AND-GROUPS; more /etc/passwd; more /etc/group

because 'more /etc/group' will be considered as a directory to switch to. You must write a small script instead:

#! /bin/sh

more /etc/passwd more /etc/group

Supposing the script name is `show_ug`, the '`gnuitrc.TERM`' line will look like this:

^AAA = SHOW-USERS-AND-GROUPS; show_ug

There is no support for appearance modes on magic-cookie terminals.

6 GNU Interactive Tools bugs

Any questions, comments, or bug reports, should be emailed to '`gnuit-dev@gnu.org`'. Please include the version number.

Appendix A Copying This Manual

under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

   You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

   The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

   In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

   Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

    The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See `http://www.gnu.org/copyleft/`.

    Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

    "Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

    "CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

    "Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

    An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

    The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Key Index

^

D

E

Command Index

E

F

G

H

I

J

K

L

M

N

O

P

Variable Index

Concept Index

A

B

C

D

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y

Short Contents

Table of Contents