GXedit User Manual

Revision 1.22

© 1998-99 Patrick Lambert <drow@darkelf.net>
http://devplanet.fastethernet.net/gxedit.html

General help

Syntax:
gxedit [text files] [-v] [-f] [-n] [-i] [-d] [-r] [-p project] [-s script]

The -**v** option is used to open the file in view only mode.

The -**f** option disables the menus and toolbars

The -**n** option disables the networking part. Useful if you do not have access to a network.

The **-i** option waits for input. Useful with tail -f

The **-d** option gives debug information on startup

The **-r** option resets the config file

The **-p** option loads a project file on startup

The **-s** option loads a script on startup

The compilation of GXedit is very simple. You need to run *./setup* and then a graphical setup box will appear. You can also customize compilation-time variables. To run Gxedit, simply type the syntax above.

GXedit is a graphical multi function text editor. It allows you to edit text in a X Window program.

You can call all the available GXedit functions from the menus. Also, you can use the toolbars for functions like Open and Exit.

The main toolbar's icons, in order, are:

New, Open, Save, Copy, Paste, Settings, Help, Exit.

You'll notice that from the File menu you can import various formats. These commands use simple converters to convert a foreign format to plain text. It might not be completely accurate, but it makes it easier to read if you have no program that natively reads that file format. The converting functions should get better in next versions. Also other import programs can be used as import filters can be added from the settings panel.

From the Edit menu, you can Copy and Paste text, Find strings, and you can include man tags in your text.

The Edit menu also has an Undo command, that will restore the text as it was just before you used a scripting command, a macro or checked spelling.

An interesting thing you will find is the mirror features. With them, you can set your GXedit as a server where other people using the program on other systems will be able to connect to yours, and see everything you type as you type it. It allows up to 99 connections. This can be very useful for a meeting across the Internet or on a LAN.

Note that the data is not encrypted and anyone can connect if they know the server/port. Use only on a local LAN if sensitive data will be sent.

From the Links menu, you can open your text in various other programs. You can also run Gxedit scripts from there.

There is also the Options menu for various tools like spell checking. It's also there that will appear the GXedit scripts you can add in the settings panel.

The lines count, word count and bytes count are approximate.

Finally, the Help menu provides this help screen.

GXedit also has an autosave feature that can save your text in *~/.gxedit.autosave* while you're writing, until you exit GXedit, in case something crashes. You can turn this on or off in the settings panel. You can reload the autosave file from within Edit/Undo.

It also has a debug window. Major notices will appear in a message box, but other ones will appear in the debug window.

Note that while this is a text editor, if you try to load images, an image viewer will be called to load them. You can set your image viewer in the settings panel.

You should not confuse the Add man tags option and the Import man page option. One is to make a man page and will insert 'tags' in your text, while the other will load a man page and convert its tags to plain text.

GXedit now provides syntax highlighting. For this you need to install the GtkEditor patch to GTK (http://www.daimi.aau.dk/~mailund/gtk.html) and enable it in the Settings panel and at compile time. Note that GtkEditor is in early alpha and syntax highlighting is not very stable.

Network options

GXedit is fully network aware. You can load an e-mail from your e-mails folder, send your current text as e-mail, load a text from the web, and more.

All these functions are available from the menus.

Note that you'll want to configure your settings correctly before using any network function.

Also, the News fetching may be slow when Gxedit needs to fetch the complete newsgroups list.

Settings panel

The settings panel can be accessed from the Options menu or from the toolbar.

The settings are kept in a files called *.gxedit** in your home directory, and the main file is created the first time you start GXedit.

There are several panels. The display settings are easy to set up. The advanced panel can be left unchanged, but you have to configure the networking part.

The advanced panel is for fonts and colors. Interesting fonts include '*misc fixed*', '*adobe courier*', '*adobe helvetica*' and '*b&h lucida*'. Check with *xfontsel* to get your fonts list.

If you don't know what your mail server or e-mail address is, you should ask your system administrator.

Note that if the Reply option is on, any mail or news article will be opened with a reply char '>' and *~/.signature* will be appended at the end of the text, to allow easy replying.

The auto indent option means that the indent level will go up when you add a { and go down when you type }.

The importing filters are simple filters used to add choices to the Import menu. If you have binaries on your system that convert a file format to plain text, like *dvi2tty* does, then you can use a filter to add an import option to GXedit. In the filter line, $i means the input file and $o the output one. GXedit will replace them as needed. The default choices are dvi to text, RPM file info and more. This obviously means that you must have *dvi2tty* and RPM installed on your system.

The last panels are Editing and Menus. From them, you can toggle on or off the option to have bolded menu and toolbar text, the Audio menu, and other misc things.

The Audio menu choices require you to type '*make say*' and '*make say-install*' in the GXedit source tree. The say utility is based on *rsynth*. See the *README* in the access dir for more.

You can compile GXedit to access a SOCKS proxy server (with the proxy option on) if you define USE_SOCKS and link with the SOCKS static lib.

Keybindings

The key bindings are now based on the F keys. For example, F1 n starts a new instance of GXedit. See the Quick Reference Card for a complete list (*quickref.ps*)

The F keys are defined in *config.h*, and right now relies on GDK to provide the right codes on the keyboard. If they don't work, try to modify them in the config file.

Encryption

GXedit can use the *des(1)* or *idea(1)* binary to encrypt and decrypt text files. The DES binary is part of the libdes library and can be found at ftp://ftp.psy.uq.oz.au/pub/Crypto/DES/ while IDEA is available from ftp://idea.sec.dsi.unimi.it/pub/crypt/code/

The encryption is based on key using. You encrypt a file with a key, and decrypt it with the same key. The encrypted text is saved under *filename.enc*

Note that this encryption could possibly be broken by powerful computers, even if IDEA is more secure then what DES can offer. Also, since GXedit uses temporary files, the text is not secure on the computer where GXedit is used, only wherever you send the *.enc* file. Also make sure the key doesn't appears in logs like *.bash_history*

The encryption command can be useful for data storage, or for sending a text over an insecure network.

Also note that this is not compatible with SunOS's default *des(1)* binary. See the *des(1)* and *idea(1)* man pages for more information.

GXedit scripts

GXedit scripts must not be confused with AWK and SED support in

GXedit.

The scripts are used to add functions in the menu Links. With them, you can call other programs, shell commands, and bring that text back into GXedit.

An example script is in *userinfo.scr*, and you can add scripts to GXedit in the settings panel, Misc tab.

The script can contain any of those 4 commands:

command:	which is used to execute a shell command
command-bg:	which also calls a command but in the background
open-file:	this command opens a file
open-file-new:	open a new GXedit window to open the text file
chdir:	Change the current directory

To include a script in GXedit, from the settings panel, you include this syntax:
script: <number> <script filename> <description>

The following are variables you can use on the command and command-bg lines:

!tmpname	Temp name
!login	Login name
!images	Images viewer
!program	GXedit program name
!mail-server	Mail server name
!news-server	News server name
!file-manager	File manager
!documents-dir	Your documents dir
!gzip	GZ compress program
!compile	Your set compile line
!xterm	The Xterm program

This version of GXedit comes with *gtk-shell*, a useful utility to provide visual queries and messages to the user, to be used in the scripts (see *userinfo.scr*)

AWK and SED scripting

The scripting function in the Options menu can be very useful for complex text manipulation.

The scripting languages chosen in GXedit are Awk and Sed, because those languages go well together to manipulate text.

Let's see some examples of what you could do with Awk. Open the

README file and use the scripting command:
/is/
This will remove every line that does not have 'is' in it.

On the contrary, if you want all the lines that do not contain 'is', you would use:
!/is/

If you wanted the pattern to match both 'is' and 'Is', you could have used:
/[iI]s/

Let's take another example:
{ count=count+1 } END { print count }

Always remember that anything before BEGIN and after END is executed once, while what's between the two is executed for every line of text. So this command means add 1 to the variable count for every line of text, and when all lines have been counted, print the variable. The result is Awk will print the number of lines.

Now let's say you want to change all your text to upper-case chars. Awk has a function called toupper() to do that. So the proper command line would be:
{ print toupper($0) }

Where $0 means the complete line, and $1 would mean the first word of that line, $2 the second word, etc...

Let's say that you need a quick way to get all the real names of all users on the system. Open /etc/passwd then use this script:
BEGIN { FS=":" } { print $5 }
This tells awk to use ':' between the words, and to print word 5 of every line.

Another way to do it is:
{ print $5 }
with:
-F:
In the 'Optional flags' entry box.

Another interesting command is length. This command line will print every line with at least 10 chars in it:
length > 10

This command will print the total amount of chars (letters or spaces) in your text:
{ count=count+length($0) } END { print count }

Also, this command will count the number of times the word 'is' can be found and print the result:
{ for (i=1; i<=NF; i++) if ($i=="is") count=count+1 } END { print count }

Finally, you might need to add a flag to the Optional flag entry box. If you want to see all the hidden chars like EOL, then enter the letter l in the command line, and -n in flags.

Copyright

Available at http://devplanet.fastethernet.net
or http://mirrors.darkelf.net
or http://sunsite.unc.edu/pub/Linux

I'd like to thank everyone who gave me ideas and provided comments about GXedit, especially Glen Parker.

See the source code for other possible credits.