



Full credit is given to the above companies including the Operating System (OS) that this PDF file was generated!

Rocky Enterprise Linux 9.2 Manual Pages on command 'xedit.1'

\$ man xedit.1

XEDIT(1) General Commands Manual XEDIT(1)

NAME

xedit - simple text editor for X

SYNTAX

xedit [-toolkitoption ...] [filename ...]

DESCRIPTION

Xedit provides a window consisting of the following four areas:

Commands Section A set of commands that allow you to exit xedit, save the file, or load a new file into the edit window.

Message Window Displays xedit messages. In addition, this window can be also used as a scratch pad.

Filename Display Displays the name of the file currently being edited, and whether this file is Read-Write or Read Only.

Edit Window Displays the text of the file that you are editing or creating.

OPTIONS

Xedit accepts all of the standard X Toolkit command line options (see X(7)). The order of the command line options is not important.

filename

Specifies the file(s) that are to be loaded during start-up. This is the file which will be edited. If a file is not specified, xedit lets you load files or create new files after it has started up.

EDITING

The Athena Text widget is used for the three sections of this application that allow text

input. The characters typed will go to the Text widget that has the input focus, or the Text widget that the pointer cursor is currently over.

The following keystroke combinations are defined:

Ctrl-a	Beginning Of Line
Meta-b	Backward Word
Ctrl-b	Backward Character
Meta-f	Forward Word
Ctrl-d	Delete Next Character
Meta-i	Insert File
Ctrl-e	End Of Line
Meta-k	Kill To End Of Paragraph
Ctrl-f	Forward Character
Meta-q	Form Paragraph
Ctrl-g	Keyboard Reset
Meta-v	Previous Page
Ctrl-h	Delete Previous Character
Meta-y	Insert Current Selection
Ctrl-j	Newline And Indent
Meta-z	Scroll One Line Down
Ctrl-k	Kill To End Of Line
Meta-d	Delete Next Word
Ctrl-l	Redraw Display
Meta-D	Kill Word
Ctrl-m	Newline
Meta-h	Delete Previous Word
Ctrl-n	Next Line
Meta-H	Backward Kill Word
Ctrl-o	Newline And Backup
Meta-<	Beginning Of File
Ctrl-p	Previous Line
Meta->	End Of File
Ctrl-r	Search/Replace Backward
Meta-]	Forward Paragraph

Ctrl-s Search/Replace Forward
 Meta-[Backward Paragraph
 Ctrl-t Transpose Characters
 Ctrl-u [number] Multiply by 4 or number
 Meta-Delete Delete Previous Word
 Ctrl-v Next Page
 Meta-Shift Delete Kill Previous Word
 Ctrl-w Kill Selection
 Meta-Backspace Delete Previous Word
 Ctrl-y Unkill
 Meta-Shift Backspace Kill Previous Word
 Ctrl-z Scroll One Line Up
 Meta-z Scroll One Line Down
 Ctrl-_ Undo
 Escape Line Edit Mode

In addition, the pointer may be used to cut and paste text:

Button 1 Down Start Selection
 Button 1 Motion Adjust Selection
 Button 1 Up End Selection (cut)
 Button 2 Down Insert Current Selection (paste)
 Button 3 Down Extend Current Selection
 Button 3 Motion Adjust Selection
 Button 3 Up End Selection (cut)

LINE EDIT MODE

Line edit mode enables several shortcut commands for searching and replacing text in a xedit buffer. Line edit mode commands have the format:

[line-number[,line-number]]command[parameters]

Line number may be specified as:

. The current text line.

\$ The last line of the file.

number The literal line number.

- or ^ The previous line. Equivalent to -1.

-number or ^number

The current line minus number.

+ The next line. Equivalent to +1.

+number The current line plus number.

, or % From the first to the last line. Equivalent to 1,\$.

; From the current to the last line. Equivalent to .,\$.

Command may be specified as:

s Substitute text in the specified lines.

/re/ Search forward for the regular expression pattern re.

?re? Search backward for the regular expression pattern re.

Parameters may be specified as:

/re/ Works as a parameter to i or as a command.

/re/text/

Search forward for re and substitute by text.

Options may follow or be parameters, known values are:

i Case insensitive search.

g Global match when replacing text. Unless specified, only the nth, that defaults to 1, match will be replaced.

c Confirm before replacing text.

number Replace only the occurrence referenced by number.

Commands accept some variations, examples:

/pattern/i

i/pattern/

i/pattern

Search forward for pattern.

,sc/pattern/text

,sc/pattern/text/

,s/pattern/text/c

Search the entire buffer and ask confirmation to replace pattern with text.

,s/pattern/text/number

Replace the match number in the text line. If not specified, defaults to the first occurrence.

When searching for text, type <Return> to go to the next match. When interactively replacing text, type y or Y to accept the change, and n or N to ignore it and go to the next

match.

COMMANDS

Quit Quits the current editing session. If any changes have not been saved, `xedit` displays a warning message, allowing the user to save them.

Save If file backups are enabled (see **RESOURCES**, below) `xedit` stores a copy of the original, unedited file in `<prefix>file<suffix>`, then overwrites the file with the contents of the edit window. The filename is retrieved from the Text widget directly to the right of the Load button.

Load Loads the file named in the text widget immediately to the right of the button and displays it in the Edit window.

RESOURCES

For `xedit` the available resources are:

`tagsName` (Class `TagsName`)

Specifies the name of the tags file to search when loading a new file. Default value is `tags`.

`loadTags` (Class `LoadTags`)

Boolean value to enable or disabling searching for tags files. Default is `True`.

`enableBackups` (Class `EnableBackups`)

Specifies that, when edits made to an existing file are saved, `xedit` is to copy the original version of that file to `<prefix>file<suffix>` before it saves the changes. The default value for this resource is `on`, stating that backups should be created.

`backupNamePrefix` (Class `BackupNamePrefix`)

Specifies a string that is to be prepended to the backup filename. The default is that no string shall be prepended.

`backupNameSuffix` (Class `BackupNameSuffix`)

Specifies a string that is to be appended to the backup filename. The default is to use `~` as the suffix.

`positionFormat` (Class `Format`)

Specifies a format string used to display the cursor position. This string uses `printf(3)` like notation, where `%l` prints the line number, `%c` prints the column number, `%p` prints the insert position offset, and `%s` prints the current file size. It is also allowed to specify field sizes, with the notation `%-?[0-9]+`. The default

fault format string is ``L%l'', which shows the character ``L'' followed by the line number.

hints (Class Hints)

Specifies a list of strings, separated by new lines, that will be displayed in the bc_label window.

hintsInterval (Class Interval)

Specifies the interval in seconds, which the hint string in the bc_label window will be changed.

changedBitmap (Class Bitmap)

Specifies the name of the Bitmap that will be displayed in the fileMenu, when the file being edited is changed.

autoReplace (Class Replace)

This resource is useful to automatically correct common misspelling errors, but can also be used to create simple macros. The format is {non-blanks}{blanks}{{string}}. Fields are separated by newlines. Example of use:

```
nto      not\n\
```

```
/macro some long string with \\n newlines \\n
```

Will automatically replace the word nto by not, and /macro by some long string with

newlines when you type that words.

ispell.dictionaries (Class ispell.Dictionary)

Specifies a list of dictionary names, separated by spaces, available to the ispell program. The default value is "american americamed+ english".

ispell.dictionary (Class ispell.Dictionary)

Specifies the default dictionary to use.

ispell*<DICTIONARY>.wordChars (Class ispell*Chars)

Specifies a set of characters that can be part of a legal word. The <DICTIONARY> field is one of the dictionaries specified in the dictionaries resource.

ispell.ispellCommand (Class ispell.CommandLine)

The path to the ispell program, and possibly, additional arguments. You don't need to specify the ``-w'' option, neither the ``-a'' option. Refer to the ispell(1) manpage for more information on ispell options.

ispell.formatting (Class ispell.TextFormat)

Specifies which text formatting to use while spell checking the file. The available formats are text and html.

`ispell*text.skipLines` (Class `ispell*text.Skip`)

Lines starting with one of the characters in this string will not be spell checked. This resource is only used in text mode.

`ispell.terseMode` (Class `ispell.Terse`)

When enabled, runs ispell in terse mode, not asking user interaction for words generated through compound formation (when using the `ispell -C` option), or words generated through affix removal. The default value is `False`.

`ispell.lookCommand` (Class `ispell.CommandLine`)

The path to the program to search for alternate words, and possibly, additional arguments. The default program used is `/usr/bin/egrep`.

`ispell.wordsFile` (Class `ispell.Words`)

The path to the file[s] to search for alternate words. The default file is `/usr/share/dict/words`.

`ispell.guessLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when ispell returns a guess list of one or more words. The default value is `Guess`.

`ispell.missLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when ispell returns a list of one or more words to match a misspelled one. The default value is `Miss`.

`ispell.rootLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when the word is not in the dictionary, but it can be formed through a root one. The default value is `Root: , and is followed by a space and the root word`.

`ispell.noneLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when there is no near misses. The default value is `None`.

`ispell.compoundLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when the word being checked is formed by concatenation of two words. The default value is `Compound`.

`ispell.okLabel` (Class `ispell.Status`)

String displayed in the ispell status bar when the checked word is in the dictionary.

nary. This string is only displayed when using the check button in the xedit ispell interface. The default value is Ok.

ispell.eofLabel (Class ispell.Status)

The string displayed in the ispell status bar when the end of the file is reached.

The default value is End Of File.

ispell.repeatLabel (Class ispell.Status)

The string displayed in the ispell status bar when two identical words are found together in the file. The default value is Repeat.

ispell.lookLabel (Class ispell.Status)

The string displayed in the ispell status bar after displaying the results of the Look command. If no results are found, the value of the ispell.noneLabel resource is shown.

ispell.workingLabel (Class ispell.Status)

The string displayed in the ispell status bar while xedit is communicating with ispell. The default value is

WIDGETS

In order to specify resources, it is useful to know the hierarchy of the widgets which compose xedit. In the notation below, indentation indicates hierarchical structure. The widget class name is given first, followed by the widget instance name.

Xedit xedit

 Paned paned

 Paned buttons

 Command quit

 Command save

 Command load

 Text filename

 Label bc_label

 Text messageWindow

 Label labelWindow

 Text editWindow

ENVIRONMENT

DISPLAY to get the default host and display number.

XENVIRONMENT to get the name of a resource file that overrides the global resources

stored in the RESOURCE_MANAGER property.

FILES

/etc/X11/app-defaults/Xedit

specifies required resources

SEE ALSO

X(7), xrdp(1), Athena Widget Set

RESTRICTIONS

Xedit is not a replacement to Emacs.

COPYRIGHT

Copyright 1988, Digital Equipment Corporation.

Copyright 1989, X Consortium

Copyright 1998, The XFree86 Project

See X(7) for a full statement of rights and permissions.

AUTHORS

Chris D. Peterson, MIT X Consortium

Paulo Cesar Pereira de Andrade, The XFree86 Project

X Version 11

xedit 1.2.2

XEDIT(1)