NAME

ex – text editor

SYNOPSIS

/usr/bin/ex [– | –**s**] [–**l**] [–**L**] [–**R**] [–**r** [*file*]] [–**t** *tag*] [–**v**] [–**V**] [–**x**] [–**w***n*] [–**C**] [+*command* | –**c** *command*] *file*. . .

/usr/xpg4/bin/ex [– | –**s**] [–**l**] [–**L**] [–**R**] [–**r** [*file*]] [–**t** *tag*] [–**v**] [–**V**] [–**x**] [–**w***n*] [–**C**] [+*command* | –**c** *command*] *file*. . .

AVAILABILITY

/usr/bin/ex

SUNWcsu

/usr/xpg4/bin/ex

SUNWxcu4

DESCRIPTION

ex is the root of a family of editors: **ex** and **vi**. **ex** is a superset of **ed**(1), with the most notable extension being a display editing facility. Display based editing is the focus of **vi**.

If you have a CRT terminal, you may wish to use a display based editor; in this case see **vi**(1), which is a command which focuses on the display-editing portion of **ex**.

For ed Users

If you have used **ed** you will find that, in addition to having all of the **ed** commands available, **ex** has a number of additional features useful on CRT terminals. Intelligent terminals and high speed terminals are very pleasant to use with **vi**. Generally, the **ex** editor uses far more of the capabilities of terminals than **ed** does, and uses the terminal capability data base (see **terminfo**(4)) and the type of the terminal you are using from the environment variable **TERM** to determine how to drive your terminal efficiently. The editor makes use of features such as insert and delete character and line in its **visual** command (which can be abbreviated **vi**) and which is the central mode of editing when using the **vi** command.

ex contains a number of features for easily viewing the text of the file. The **z** command gives easy access to windows of text. Typing **^D** (CTRL-D) causes the editor to scroll a half-window of text and is more useful for quickly stepping through a file than just typing return. Of course, the screen-oriented **visual** mode gives constant access to editing context.

ex gives you help when you make mistakes. The **undo** (**u**) command allows you to reverse any single change which goes astray. **ex** gives you a lot of feedback, normally printing changed lines, and indicates when more than a few lines are affected by a command so that it is easy to detect when a command has affected more lines than it should have.

The editor also normally prevents overwriting existing files, unless you edited them, so that you do not accidentally overwrite a file other than the one you are editing. If the system (or editor) crashes, or you accidentally hang up the telephone, you can use the editor **recover** command (or –**r** *file* option) to retrieve your work. This will get you back to within a few lines of where you left off.

ex has several features for dealing with more than one file at a time. You can give it a list of files on the command line and use the **next** (**n**) command to deal with each in turn. The **next** command can also be given a list of file names, or a pattern as used by the shell to specify a new set of files to be dealt with. In general, file names in the editor may be formed with full shell metasyntax. The metacharacter '%' is also available in forming file names and is replaced by the name of the current file.

The editor has a group of buffers whose names are the ASCII lower-case letters (**a**-**z**). You can place text in these named buffers where it is available to be inserted elsewhere in the file. The contents of these buffers remain available when you begin editing a new file using the **edit** (**e**) command.

There is a command **&** in **ex** which repeats the last **substitute** command. In addition, there is a confirmed substitute command. You give a range of substitutions to be done and the editor interactively asks whether each substitution is desired.

It is possible to ignore the case of letters in searches and substitutions. **ex** also allows regular expressions which match words to be constructed. This is convenient, for example, in searching for the word "edit" if your document also contains the word "editor."

ex has a set of options which you can set to tailor it to your liking. One option which is very useful is the **autoindent** option that allows the editor to supply leading white space to align text automatically. You can then use **^D** as a backtab and space or tab to move forward to align new code easily.

Miscellaneous useful features include an intelligent **join** (**j**) command that supplies white space between joined lines automatically, commands **<** and **>** which shift groups of lines, and the ability to filter portions of the buffer through commands such as **sort**.

OPTIONS

The following options are supported:

– | **–s** Suppress all interactive user feedback. This is useful when processing editor scripts.

–l Set up for editing LISP programs.

–L List the name of all files saved as the result of an editor or system crash.

–R **Readonly** mode; the **readonly** flag is set, preventing accidental overwriting of the file.

–r *file* Edit *file* after an editor or system crash. (Recovers the version of *file* that was in the buffer when the crash occurred.)

–t *tag* Edit the file containing the *tag* and position the editor at its definition.

–v Start up in display editing state using **vi**. You can achieve the same effect by simply typing the **vi** command itself.

–V Verbose. Any non-tty input will be echoed on standard error. This may be useful when processing editor commands within shell scripts.

–x Encryption option. Simulates the **X** command and prompts the user for a key. This key is used to encrypt and decrypt text using the algorithm of the **crypt** command. The **X** command makes an educated guess to determine whether text read in is encrypted or not. The temporary buffer file is encrypted also, using a transformed version of the key typed in for the −**x** option.

−**w***n* Set the default window size to *n*. This is useful when using the editor over a slow speed line.

−**C** Encryption option. Same as the −**x** option, except simulates the **C** command. The **C** command is like the **X** command, except that all text read in is assumed to have been encrypted.

+*command* | −**c** *command*
Begin editing by executing the specified editor *command* (usually a search or positioning command).

/usr/xpg4/bin/ex

If both the −**t** *tag* and the −**c** *command* options are given, the −**t** *tag* will be processed first. That is, the file containing the tag is selected by −**t** and then the command is executed.

OPERANDS

The following operand is supported:

file A path name of a file to be edited.

USAGE

ex States

Command Normal and initial state. Input prompted for by ‘‘**:**’’. Your line kill character cancels a partial command.

Insert Entered by **a**, **i**, or **c**. Arbitrary text may be entered. Insert state normally is terminated by a line having only "**.**" on it, or, abnormally, with an interrupt.

Visual Entered by typing **vi**; terminated by typing **Q** or ^\ (CTRL-\).

ex Command Names and Abbreviations

abbrev	**ab**	map		set	**se**
append	**a**	mark	**ma**	shell	**sh**
args	**ar**	move	**m**	source	**so**
change	**c**	next	**n**	substitute	**s**
copy	**co**	number	**nu**	unabbrev	**unab**
delete	**d**	preserve	**pre**	undo	**u**
edit	**e**	print	**p**	unmap	**unm**
file	**f**	put	**pu**	version	**ve**
global	**g**	quit	**q**	visual	**vi**
insert	**i**	read	**r**	write	**w**
join	**j**	recover	**rec**	xit	**x**
list	**l**	rewind	**rew**	yank	**ya**

/usr/xpg4/bin/ex

ex Command Arguments

For all of **ex** commands listed below, If both a count and a range are specified for a command that uses them, the number of lines affected will be taken from the count value rather than the range. The starting line for the command is taken to be the first line addressed by the range.

Abbreviate	ab[brev] word rhs
Append	[line] a[ppend][!]
Arguments	ar[gs]
Change	[range] c[hange][!] [count]
Change Directory	chd[ir][!] [directory]; cd[!] [directory]
Copy	[range] co[py] line [flags]; [range] t line [flags]
Delete	[range] d[elete] [buffer] [count] [flags]
Edit	e[dit][!] [+line][file]; ex[!] [+line] [file]
File	f[ile] [file]
Global	[range] g[lobal] /pattern/ [commands]; [range] v /pattern/ [commands]
Insert	[line] i[nsert][!]
Join	[range] j[oin][!] [count] [flags]
List	[range] l[ist] [count] [flags]
Map	map[!] [x rhs]
Mark	[line] ma[rk] x; [line] k x
Move	[range] m[ove] line
Next	n[ext][!] [file ...]
Number	[range] nu[mber] [count] [flags]; [range] # [count] [flags]
Open	[line] o[pen] /pattern/ [flags]
Preserve	pre[serve]
Print	[range] p[rint] [count] [flags]
Put	[line] pu[t] [buffer]
Quit	q[uit][!]
Read	[line] r[ead][!] [file]
Recover	rec[over] file
Rewind	rew[ind][!] Set se[t] [option[=[value]]...] [nooption...] [option?...] [all]
Shell	sh[ell]
Source	so[urce] file
Substitute	[range] s[ubstitute] [/pattern/repl/[options] [count] [flags]]
Suspend	su[spend][!]; st[op][!]
Tag	ta[g][!] tagstring
Unabbreviate	una[bbrev] word
Undo	u[ndo]
Unmap	unm[ap][!] x
Visual	[line] vi[sual] [type] [count] [flags]
Write	[range} w[rite][!] [>>] [file]; [range} w[rite] [!] [file]; [range} wq[!] [>>] [file]
Write and Exit	[range] x[it][!] [file]
Yank	[range] ya[nk] [buffer] [count]
Adjust Window	[line] z [type] [count] [flags]
Escape	! command [range]! command
Shift Left	[range] < [count] [flags]
Shift Right	[range] > [count] [flags]

Resubstitute	[range] & [options] [count] [flags]; [range] s[ubstitute] [options] [count] [flags]; [range] ˜ [options] [count] [flags]
Scroll	EOF
Write Line Number	[line] = [flags]
Execute	@ buffer; ∗ buffer

ex Commands

forced encryption	**C**	heuristic encryption	**X**
resubst	**&**	print next	CR
rshift	>	lshift	<
scroll	**ˆD**	window	**z**
shell escape	**!**		

ex Command Addresses

n	line *n*	/*pat*	next with *pat*
.	current	**?***pat*	previous with *pat*
$	last	*x*-*n*	*n* before *x*
+	next	*x*,*y*	*x* through *y*
–	previous	´*x*	marked with *x*
+*n*	*n* forward	´´	previous context
%	1,$		

Initializing options

EXINIT	place **set**'s here in environment variable
$HOME/.exrc	editor initialization file
./.exrc	editor initialization file
set *x*	enable option *x*
set no*x*	disable option *x*
set *x***=val**	give value *val* to option *x*
set	show changed options
set all	show all options
set *x***?**	show value of option *x*

Most useful options and their abbreviations

autoindent	**ai**	supply indent
autowrite	**aw**	write before changing files
directory		pathname of directory for temporary work files
exrc	**ex**	allow **vi**/**ex** to read the **.exrc** in the current directory. This option is set in the **EXINIT** shell variable or in the **.exrc** file in the **$HOME** directory.
ignorecase	**ic**	ignore case of letters in scanning
list		print **ˆI** for tab, $ at end
magic		treat **. [** ∗ special in patterns
modelines		first five lines and last five lines executed as **vi**/**ex** commands if they are of the form **ex:***command***:** or **vi:***command***:**
number	**nu**	number lines

paragraphs	**para**	macro names that start paragraphs
redraw		simulate smart terminal
report		informs you if the number of lines modified by the last command is greater than the value of the **report** variable
scroll		command mode lines
sections	**sect**	macro names that start sections
shiftwidth	**sw**	for **< >**, and input **^D**
showmatch	**sm**	to **)** and **}** as typed
showmode	**smd**	show insert mode in **vi**
slowopen	**slow**	stop updates during insert
term		specifies to **vi** the type of terminal being used (the default is the value of the environment variable **TERM**)
window		visual mode lines
wrapmargin	**wm**	automatic line splitting
wrapscan	**ws**	search around end (or beginning) of buffer

Scanning pattern formation

^	beginning of line
$	end of line
.	any character
\<	beginning of word
\>	end of word
[str]	any character in *str*
[^str]	any character not in *str*
[*x–y***]**	any character between *x* and *y*
*	any number of preceding characters

ENVIRONMENT

See **environ**(5) for descriptions of the following environment variables that affect the execution of **ex**: **HOME**, **PATH**, **SHELL**, **TERM**, **LC_COLLATE**, **LC_CTYPE**, **LC_MESSAGES**, and **NLSPATH**.

COLUMNS Override the system-selected horizontal screen size.

EXINIT Determine a list of **ex** commands that are executed on editor start-up, before reading the first file. The list can contain multiple commands by separating them using a vertical-line (**|**) character.

LINES Override the system-selected vertical screen size, used as the number of lines in a screenful and the vertical screen size in visual mode.

EXIT STATUS

The following exit values are returned:

0 Successful completion.

>0 An error occurred.

FILES

/var/tmp/Ex*nnnnn*	editor temporary
/var/tmp/Rx*nnnnn*	named buffer temporary
/usr/lib/expreserve	preserve command

/usr/lib/exrecover	recover command
/usr/lib/exstrings	error messages
/usr/share/lib/terminfo/∗	describes capabilities of terminals
/var/preserve/login	preservation directory (where **login** is the user's login)
$HOME/.exrc	editor startup file
./.exrc	editor startup file

SEE ALSO

ed(1), **edit**(1), **grep**(1), **sed**(1), **sort**(1), **vi**(1), **curses**(3X), **term**(4), **terminfo**(4), **environ**(5)

Solaris Advanced User's Guide

AUTHOR

vi and **ex** are based on software developed by The University of California, Berkeley California, Computer Science Division, Department of Electrical Engineering and Computer Science.

NOTES

Several options, although they continue to be supported, have been replaced in the documentation by options that follow the Command Syntax Standard (see **intro**(1)). The – option has been replaced by –**s**, a –**r** option that is not followed with an option-argument has been replaced by –**L**, and **+***command* has been replaced by –**c** *command*.

The message **file too large to recover with** –**r option**, which is seen when a file is loaded, indicates that the file can be edited and saved successfully, but if the editing session is lost, recovery of the file with the –**r** option will not be possible.

The **z** command prints the number of logical rather than physical lines. More than a screen full of output may result if long lines are present.

File input/output errors do not print a name if the command line –**s** option is used.

The editing environment defaults to certain configuration options. When an editing session is initiated, **ex** attempts to read the **EXINIT** environment variable. If it exists, the editor uses the values defined in **EXINIT**, otherwise the values set in **$HOME/.exrc** are used. If **$HOME/.exrc** does not exist, the default values are used.

To use a copy of **.exrc** located in the current directory other than **$HOME**, set the *exrc* option in **EXINIT** or **$HOME/.exrc**. Options set in **EXINIT** can be turned off in a local **.exrc** only if *exrc* is set in **EXINIT** or **$HOME/.exrc**.

There is no easy way to do a single scan ignoring case.

The editor does not warn if text is placed in named buffers and not used before exiting the editor.

Null characters are discarded in input files and cannot appear in resultant files.

The standard Solaris version of **ex** will be replaced by the POSIX.2 conformant version in the future. Scripts which use the **ex** family of addressing and features should use the **/usr/xpg4/bin** version of these utilities.