

## Vim Keys

For a complete list of supported EX command, see [Vim EX commands](#).

### Differences Between SlickEdit Vim and gvim

SlickEdit Vim emulation does not support all Vim EX commands. Some are probably not worth adding. Others haven't been done yet.

### Note

Please post changes/enhancements you would like to see for SlickEdit's Vim emulation on the forum.

Some more significant differences are listed below:

<code>%V, %#, %'m, %l, %c, %V, %[]</code>	Not yet supported in Vim regex syntax.
<code>ze, lz1..lz9, lz*(l)</code>	
<code> </code>	Not supported in EX command line. Vim uses this to add additional commands.

### Vim Cursor Movement

Left arrow, Ctrl+J	Cursor left
Right arrow, Ctrl+L	Cursor right
Up arrow, Ctrl+I	Cursor up
Down arrow, Ctrl+K	Cursor down
Ctrl+Home, Ctrl+X Ctrl+U	Top of buffer
Ctrl+End, Ctrl+X Ctrl+J	Bottom of buffer
Home	Begin line
End, Ctrl+O	End line
PgUp, Ctrl+B	Page up
PgDn, Ctrl+F	Page down
Ctrl+Left	Previous word
Ctrl+Right	Next word
Ctrl+PgUp	Top of window
Ctrl+PgDn	Bottom of window
Tab	Indent to next tab stop
Shift+Tab	Indent to previous tab stop
Ctrl+	Add multiple cursors
Ctrl+Shift+Alt+Down	Add new cursor below current cursor
Ctrl+Shift+Alt+Up	Add new cursor above current cursor

### Vim Cursor Movement - Normal Mode Only

gj	Down screen line (different when lines wrap)
gk	Up screen line (different when lines wrap)
-	Begin previous line
^	Begin text
0	Begin line
\$	End line
G, gg	Go to line
	Go to column
w	Next word
W	Next non-white space word
b	Previous word
B	Previous non-white space word
ge	Backward to end of word
gE	Backward to end of non-white space word
e	End of word
E	End of non-white space word
(	Previous sentence
)	Next sentence
{	Previous paragraph
}	Next paragraph
[[	Previous section

]]	Next section
%	Find matching paren
N %	Move N % down a buffer
H	Move to upper-left corner of window
M	Move to middle of window
L	Move to lower-left corner of window
'	Jump to bookmarked line
~	Jump to bookmarked column
gm	Move to middle of window on the current line

### Vim Cursor Movement - Visual Mode Only

aw	Select a word
iw	Select inner word
aW	Select  WORD
iW	Select inner  WORD
as	Select a sentence
is	Select inner sentence
ap	Select a paragraph
ip	Select inner paragraph
ab	Select a block
ib	Select inner block
aB	Select a Block
iB	Select inner Block

### Vim Inserting Text

Ins	Insert/overwrite toggle
Shift+Enter	Insert a line (no syntax expansion)
Shift+Space	Insert a space (no syntax expansion)

### Vim Inserting Text - Normal Mode Only

i	Insert text
I	Insert text at beginning of line
a	Append text
A	Append text after end of line
o	Insert text below current line
O	Insert text above current line

### Vim Deleting Text

Del	Delete character under cursor
Ctrl+E	Cut to end line
Ctrl+Backspace	Cut line

### Vim Deleting Text - Normal Mode Only

x	Delete character under cursor
(visual) x	Delete selection
X	Delete character before cursor
(visual) d	Delete selection
d	Delete text
D	Delete to end of line

### Vim Searching

Command+F	Find dialog
Ctrl+R (insert mode only)	Replace dialog
Command+G	Find next occurrence
Ctrl+Shift+G, Command+Shift+G	Find previous occurrence
Ctrl+I	Incremental search
Ctrl+Shift+I	Reverse incremental search
Ctrl+Alt+Shift+F2	Stop search or search & replace

## Vim Searching - Normal Mode Only

/	Search forward (accommodates multipliers)
?	Search backward (accommodates multipliers)
n	Forward repeat last search
N	Backward repeat last search
f	Forward character search
F	Backward character search
t	Move cursor up to character
T	Move cursor backward after character
Semicolon (;)	Repeat character search
Comma (,)	Reverse repeat character search
m	Set bookmark

## Vim Selection

Ctrl+L	Select line
Shift+F7	Shift block selection left
Shift+F8	Shift block selection right
Shift+<Cursor keys>	Start or extend char/stream selection
Click & Drag	Start char/stream selection
Right-Click & Drag	Start block/column selection
Shift+Click	Extend selection
Ctrl+Right-Click	Move selection to cursor
Ctrl+Shift+Right-Click	Copy selection to cursor
Double-Click	Select word
Triple-Click	Select line
Ctrl+Click	Add cursor or selection
Shift+Right-Click	Add multiple partial line characters selections based on a block/column selection.
Ctrl+Double-Click	Add selection of words

## Vim Selection - Normal Mode Only

v	Character visual mode
V	Line visual mode
Ctrl+V	Block visual mode
o	Move cursor to beginning (or end) of selection

## Vim Clipboard

Ctrl+C, Command+C	Copy selection to clipboard
Ctrl+K	Copy word to clipboard
Ctrl+Shift+V	List clipboards and optionally paste one
Ctrl+V, Command+V, Shift+Ins	Paste
Ctrl+Backspace	Cut line
Ctrl+E	Cut to end of line
Ctrl+X (Insert mode only), Command+X, Shift+Del	Cut selection

## Vim Clipboard - Normal Mode Only

p	Paste text after cursor
]p	Paste text after cursor (adjust indent)
gp	Paste text after cursor (leave cursor after new text)
(visual) p	Paste clipboard contents over selection
P	Paste text before cursor
]P	Paste text before cursor (adjust indent)
gP	Paste text before cursor (leave cursor after new text)
y	Copy text to clipboard
(visual) y	Copy selection to clipboard
Y	Copy line to clipboard

## Vim Command Line and Text Box Editing

The following keys are different in all Text Boxes except the command line if the CUA Text Box check box is enabled (**Tools > Options > Redefine Common Keys**):

Ctrl+X	Cut selection
Ctrl+C	Copy selection to clipboard
Ctrl+V	Paste
Alt+A..Alt+Z	Taken over by dialog manager for selecting controls

## Vim Command Line and Text Box Editing - Normal Mode Only

Ctrl+Q	Cursor to command line toggle
Space	Complete argument
?	List arguments
Ctrl+Left	Previous word
Ctrl+Right	Next word
Ctrl+Shift+U	Uppcase selection
Ctrl+Shift+L	Lowcase selection
Ctrl+E	Cut to end of line
Ctrl+Backspace	Cut line
Ins	Insert/overwrite toggle
Ctrl+V	Paste
Ctrl+K	Copy word to clipboard
Ctrl+Shift+V	List clipboards and optionally paste one
Ctrl+Space	Expand alias at cursor. Use the <b>alias</b> command to define aliases or the Ex command <b>:abbr</b> .
Up arrow, Ctrl+I	Retrieve previous command
Down arrow, Ctrl+K	Retrieve next command
Shift+<Cursor keys>	Start or extend char/stream selection
Click & Drag	Start char/stream selection
Shift+Click	Extend selection
Double-Click	Select word
Triple-Click	Select line

## Vim Files and Buffers

F2, Ctrl+S, Command+S	Save current buffer
F3	Quit current buffer
F8	Next buffer
Ctrl+F8	Previous buffer
F4	Save and quit current buffer
F7, Ctrl+O, Command+O	Open a file or find buffer
Ctrl+Shift+B	List buffers
F6	File compare

## Vim Files and Buffers - Normal Mode Only

<b>:w</b>	Save current buffer
<b>:q</b>	When in "One file per window mode" close window and buffer. When in "Multiple files share window" mode, close tile without closing buffer or close buffer if only one tile left.
<b>:clo[se], :bdelete</b>	When in "One file per window" mode close window and buffer. When in "Multiple files share window" mode, close buffer.
<b>:bn[ext]</b>	Next buffer. When in "One file per window" mode, switches to next window. When in "Multiple files share window" mode, switches to next buffer within the current window.
<b>:bp[revious]</b>	Previous buffer. When in "One file per window" mode, switches to previous window. When in "Multiple files share window" mode, switches to previous buffer within the current window.
<b>:buffers</b>	List buffers.
<b>:e[dit] filename</b>	Open file specified

<b>:wq</b>	Save and quit current buffer
<b>:q!</b>	Quit current buffer without saving
<b>:wa[!]</b>	Write all buffers
<b>:qa[!]</b>	Quit all buffers
<b>:wqa[!]</b>	Write and quit all buffers
<b>:qa[!]!</b>	Quit all buffers without saving
<b>:xa[!]</b>	Write all changed buffers and exit

### Vim Windowing

Ctrl+Tab	Next window
Ctrl+Shift+Tab	Previous window
Ctrl+Shift+Z	Zoom window toggle

### Vim Windowing - Normal Mode Only

<b>:b[uffer]</b> [buffer-id]	Edits the file corresponding to the buffer id specified.
<b>:sb[uffer]</b> [buffer-id]	Split window horizontally. If [buffer-id] is specified, the file corresponding to [buffer-id] is opened.
<b>:sb[uffer]</b> [file]	Split window horizontally. If [file] is specified, file is opened.
<b>:sp[lit]</b> [file]	Split window horizontally. If [file] is specified, [file] is opened.
<b>:vs[plit]</b> [file]	Split window vertically. If [file] is specified, [file] is opened.
Ctrl+w ]	Split window and jump to symbol under cursor
Ctrl+w f	Split window and edit file name under the cursor
Ctrl+w n	Split window with empty new window
Ctrl+w o	Make current window the only visible window
Ctrl+w j	Move cursor to window below
Ctrl+w k	Move cursor to window above
Ctrl+w Ctrl+w	Move cursor to window below (wrap)
Ctrl+w W	Move cursor to window above (wrap)
Ctrl+w t	Move cursor to top window
Ctrl+w b	Move cursor to bottom window

### Vim Compiling and Programming Support

Alt+Dot	(Pro only) List symbols
Ctrl+PgUp/Ctrl+PgDn	When listing symbols. Next/previous definition
Shift+PgUp/Shift+PgDn	When listing symbols. Page up/down argument list section.
Alt+Comma	(Pro only) Parameter Info
Ctrl+Space	(Pro only) Complete symbol
Ctrl+Dot	(Pro only) Push a bookmark and go to the definition of the symbol at cursor
Ctrl+Alt+Dot	(Pro only) Push a bookmark and go to the declaration of the symbol at cursor
Ctrl+/	(Pro only) Push a bookmark and go to the first reference to the symbol at cursor
Ctrl+Comma	Pop a pushed bookmark
Ctrl+M	(Pro only) Build project
Ctrl+Shift+Down	Next error
Ctrl+Shift+Up	Previous error
Ctrl+Shift+S	Set next error
Ctrl+F5	(Pro only) Execute project
Shift+10	(Pro only) Compile current buffer
Alt+1	Cursor to error
F12	Make load current macro buffer
Ctrl+Shift+M	(Pro only) Start concurrent process
Ctrl+Shift+P	Expand extension specific alias
Ctrl+Shift+O	Expand global alias

### Vim Debugging (Pro only)

F5	Start/continue debugging
Shift+F5	Stop debugging
Ctrl+Shift+F5	Restart debugging
F9	Toggle breakpoint
Ctrl+F9	Toggle breakpoint enable
Ctrl+Shift+F9	Clear all breakpoints
F10	Step over
F11	Step into
Ctrl+F10	Run to cursor
Alt+PadStar	Show next statement
Alt+F9	Activate breakpoints window
Ctrl+Alt+W	Activate watch window
Ctrl+Alt+V	Activate variables window
Ctrl+Alt+C	Activate call stack
Ctrl+Alt+H	Activate threads window

### Vim Macros

Ctrl+F11	Start/end macro recording
F12	Make and load current macro buffer
Ctrl+Shift+F12,<key>	Stops macro recording and binds macro to <key> (which can be 0-9, A-Z, or F1-F12).
Ctrl+Break	Halt Slick-C@ macro that is prompting for a key with get_event()
Ctrl+Alt+Shift+F2	Halt Slick-C macro that is executing. Use this to terminate infinite loops.
Ctrl+Shift+Space	If editing dialog box or macro, load and run dialog box/macro.

### Vim Macros - Normal Mode Only

Dot	Repeat last insert or delete
q{a-zA-Z0-9}	Recording keyboard macro to clipboard(register) id that follows. Press 'q' again to end recording.
@{a-zA-Z0-9}	Playback keyboard macro for clipboard(register) id that follows.

### Vim Miscellaneous

Alt+F4, Command+Q, Command+F4	Safe exit
F1	Help for mode or context
Command+,	Configuration menu
Ctrl+Shift+H	Hex display toggle
Ctrl+Shift+Comma	Complete previous word/variable
Ctrl+Shift+Dot	Complete next word/variable
Ctrl+Shift+Space	Complete more
Esc	Cancel
Ctrl+Z (insert mode only), Command+Z, Alt+Backspace	Undo
Shift+F9	Undo with cursor motion grouping
Ctrl+Y (insert mode only), Command+Shift+Z	Redo
Ctrl+A (command or visual mode only)	Increment number
Ctrl+X (command or visual mode only)	Decrement number
Ctrl+Shift+U	Uppcase selection
Ctrl+Shift+L	Lowcase selection
Ctrl+] ]	Match parenthesis
Ctrl+\	Expand or collapse selective display
Ctrl+Shift+O	Expand alias at cursor

Alt+/, Command+/ Alias change directory
--

### Vim Miscellaneous - Normal Mode Only

u	Full undo
(visual) u	Change selected text to lowercase
U	Undo with cursor motion grouping
(visual) U	Change selected text to uppercase
~	Toggle the case of current character
(visual) ~	Toggle the case of the selected text
c	Change text
C	Change to end of line
(visual) c, C	Change text in selection
r	Overstrike character(s)
(visual) r	Replace text in selection
R	Overstrike text
s	Substitute character(s)
S	Substitute line
<	Shift text left
>	Shift text right
J	Join line (insert spaces)
(visual) J	Join the selected lines (insert spaces)
(visual) gJ	Join the selected lines (no spaces)
:	EX command mode
Q	EX editor mode
&	Repeat last SUBSTITUTE
Esc, Ctrl+[	Normal mode

### Vim EX command line

Many Vim EX commands take the following form:

`:[range]ex-cmd`

*range* can either be a single range specifier (see table below) or a two comma (or semicolon) delimited range specifies (as in 1,\$).

### Vim EX range specifiers

{ <i>number</i> }	Absolute line number.
.	Current line.
\$	Last line of file.
%	Same as 1,\$ which specifies the entire file.
't	Line number corresponding to book mark specified.
'<	Line number corresponding to first line of selection.
'>	Line number corresponding to last line of selection.
/ <i>pattern</i> /	Next line which matches <i>pattern</i> . Pattern defaults to Vim regular expression syntax (see ???).
? <i>pattern</i> ?	Previous line which matches <i>pattern</i> . Pattern defaults to Vim regular expression syntax (see ???).

### Vim EX commands

!: <i>external-program</i>	Run <i>external-program</i> specified.
:[ <i>range</i> ]! <i>external-program</i>	Filter line(s) specified with <i>external-program</i> specified.
:[ <i>range</i> ]<[ <i>NumLines</i> ]	Unindents line or <i>NumLines</i> specified by syntax indent width or "shiftwidth" setting. You may specify multiple less thans to unindent by a multiple of the syntax indent. For example << unindents by syntax indent*2.
:[ <i>range</i> ]>[ <i>NumLines</i> ]	Indents line or <i>NumLines</i> specified by syntax indent width or "shiftwidth" setting. You may specify multiple greater thans to indent by a multiple of the syntax indent. For example >> indents by syntax indent*2.
:bd[ <i>delete</i> ]	When in "One file per window" mode close window and buffer. When in "Multiple files share window" mode, close buffer.
:bn[ <i>ext</i> ]	Next buffer. When in "One file per window" mode, switches to next window. When in "Multiple files

share window" mode, switches to next buffer within the current window.

:bp[ <i>revious</i> ]	Previous buffer. When in "One file per window" mode, switches to previous window. When in "Multiple files share window" mode, switches to previous buffer within the current window.
:bufdo <i>cmd</i>	Execute <i>cmd</i> for each buffer.
:b[ <i>uffer</i> ] [ <i>buffer-id</i> ]	Edits the file corresponding to the buffer id specified.
:buffers	List buffers.
:cd [ <i>path</i> ]	Change directory to <i>path</i> if given or display current directory.
:clo[ <i>se</i> ]	When in "One file per window" mode close window and buffer. When in "Multiple files share window" mode, close buffer.
:[ <i>range</i> ]co[ <i>py</i> ] <i>destLine</i>	Copy line(s) after <i>destLine</i>
:[ <i>range</i> ]d[ <i>delete</i> ]	Delete line(s) specified
:e[ <i>dit</i> ] <i>filename</i>	Open file specified
:f[ <i>ile</i> ]	Displays file info which includes filename, current line number, and number of lines in the file
:[ <i>range</i> ]g[ <i>lobal</i> ] / <i>pattern</i> / <i>cmd</i>	Mark lines with occurrences of <i>pattern</i> in the line range specified. Apply <i>cmd</i> to all marked lines. <i>pattern</i> defaults to Vim regular expression syntax (see ???). <i>cmd</i> defaults to ":print" if not specified.
:[ <i>range</i> ]g[ <i>lobal</i> ]! / <i>pattern</i> / <i>cmd</i>	Same as : <b>global</b> except lines without occurrences of <i>pattern</i> are marked.
:h[ <i>elp</i> ] [ <i>vim-help-item</i> ]	Provides help on the <i>vim-help-item</i> specified. Currently supports EX commands and range specifiers. For example, : <b>help</b> :s will display help on the : <b>substitute</b> EX command. Help with no parameters displays start of Vim emulation keys section which provides a link to a list of the supported EX commands.
:[ <i>range</i> ][ <i>oin</i> ] [ <i>NumLines</i> ]	Join <i>NumLines</i> specified to current line. If a range is specified, <i>NumLines</i> is ignored and lines in range are joined to first line of range.
:k[ <i>a-z</i> ]	Set bookmark with the specified name. Note that in normal mode " goes to the next bookmark which is different than gvim.
:[ <i>range</i> ]l[ <i>ist</i> ] [ <i>NumLines</i> ]	Displays the line(s) specified with a \$ at the end of each line. Same as : <b>print</b> but appends \$ to the end of each line.
:[ <i>range</i> ]m[ <i>ove</i> ] <i>destLine</i>	Moves line(s) after <i>destLine</i>
:n[ <i>ext</i> ]	Switches to next buffer.
:noh[ <i>lsearch</i> ]	Clears highlighted search strings.
:[ <i>range</i> ]nu[ <i>umber</i> ] [ <i>NumLines</i> ]	Displays the line(s) specified with a line number at the beginning of each line. Same as : <b>print</b> but line number is displayed at the beginning of each line.
:p[ <i>rint</i> ] [ <i>NumLines</i> ]	Displays the line(s) specified.
:[ <i>range</i> ]pu[ <i>t</i> ] [ <i>x</i> ]	Paste clipboard or text specified after the last line or range specified. <i>x</i> can either be a clipboard(register) name (a-z0-9+), ="text", ="\text\\"", or = <i>mathematical_expression</i> (ex. =0x4a+0x20).
:q[ <i>uit</i> ]	When in "One file per window mode" close window and buffer. When in "Multiple files share window" mode, close tile without closing buffer or close buffer if only one tile left.
:q[ <i>uit</i> ]!	Same as : <b>quit</b> but doesn't prompt to save changes.
:qa[ <i>ll</i> ]	Close all buffers
:[ <i>range</i> ]r[ <i>ead</i> ] <i>file</i>	Insert <i>file</i> specified after the last line of range specified.
:[ <i>range</i> ]r[ <i>ead</i> ] ! <i>external-program</i>	Insert output from <i>external-program</i> specified after the last line of range specified.
:red[ <i>o</i> ]	Undoes an undo operation.
:reg[ <i>isters</i> ]	Display all named and unnamed clipboards (registers).
:rew[ <i>ind</i> ][!]	Revert the current buffer to the contents on disk. When ! is specified, buffer is reverted without prompting whether to discard changes.

**:sb[uffer]** [buffer-id]

Split window horizontally. If [buffer-id] is specified, the file corresponding to [buffer-id] is opened.

**:sb[uffer]** [file]

Split window horizontally. If [file] is specified, file is opened.

patte  
str  
in

Replace occurrences of *pattern* with occurrences of *string* in the line *range* specified. By default, *pattern* is interpreted as a Vim regular expression (see ???). *options* is a string of one or more options.

Options	Description
<b>c</b>	Confirm each substitution. 'y' to substitute match. 'l' to substitute match and then quit, 'n' to skip match, <b>Esc</b> to quit substituting, 'a' to substitute remaining matches, and 'q' to quit substituting.
<b>g</b>	Replace all occurrences in the line. With this option, only the first occurrence in each line is matched.
<b>i</b>	Case insensitive matching.
<b>l</b>	Case sensitive matching.
<b>e</b>	Case sensitive matching.
<b>&lt;</b>	If found, place cursor at beginning of word.
<b>&gt;</b>	If found, place cursor at end of word.
<b>r</b>	Interpret search <i>pattern</i> as a SlickEdit® regular expression.
<b>l</b>	Interpret <i>pattern</i> as a Perl regular expression.
<b>~</b>	Interpret <i>pattern</i> as a Vim regular expression.
<b>n</b>	Interpret <i>pattern</i> as literal text (plain text search) and not any regular syntax.
<b>u</b>	Interpret <i>pattern</i> as a Perl regular expression. Unix syntax regular expressions are no longer supported.
<b>p</b>	Print current line after substitution.
<b>w</b>	Limit search to words. Used to search for variables.
<b>v</b>	Preserve case. When specified, each occurrence found is checked for all lowercase, all uppercase, first word capitalized, or mixed case. The replace string is converted to the same case as the occurrence found except when the occurrence found is mixed case (possibly multiple capitalized words). In this case, the replace string is used without modification.

*options*

**:se[t]**

Display list of supported option settings.

**:se[t]** *option=value*

Set *option* to *value*.

**:sh[ell]**

Runs the default shell. The shell command executed is specified in the 'shell' option.

**:sp[lit]** [file], **:sbuffer** [file]

Split window horizontally. If [file] is specified, file is opened.

**:[range]t**

Synonym for copy.

**:ta[g]** [tag]

If *tag* is specified, navigates to tag specified and pushes a bookmark at the previous location. Otherwise, bookmark is popped and cursor is placed on the previous location.

**:u[ndo]** [tag]

Undo last change.

**:[range]v[lobal]** /*pattern*/ [*cmd*]

Same as **:global!**.

**:ve[rsion]**

Displays product help about information.

**:vs[plit]** [file]

Split window vertically. If [file] is specified, file is opened.

**[range]:w[rite]!**

Save lines specified to current buffer. If ! is specified, prompting to replace existing file is suppressed.

**:[range]w[rite]!** file

Save specified lines to file specified. If ! is specified, prompting to replace existing file is suppressed.

**:[range]w[rite]!** >>file

Append specified lines to file specified. If ! is specified, prompting to replace existing file is suppressed.

**:wa[ll]**

Write all buffers

**:wq**

Save the current buffer and quit.

**:wqa[ll]**

Save and quit all buffers

**:xa[ll]!**

Save all changed buffers and exit

**:x**

Same as **:wq**.

**:[range]y[ank]** [x]

Copy specified line(s) to the clipboard(register) specified.

**:[range]z** [NumLines]

Displays *NumLines* that follow the last line of the *range* specified.

