

ShortTalk: Quick Reference

Nils Klarlund (©Carnegie Mellon University, 2004)

Numerals

positive = ane | twain | traio | fairn | faif
negative = oon | twoon | truo | foorn | foof
signed = *positive* | *negative*

Punctuation and special

letter = alpha | bravo | charlie | delta | echo | foxtrot | golf
 | hotel | india | juliett | kilo | lima | mike | november
 | oscar | papa | quebec | romeo | sierra | tango
 | uniform | victor | whiskey | x-ray | yankee | zulu

<i>symbol</i> =	clam		!	rack]
	lat		@	slash		/
	numb		#	beck		“
	dall dollar		\$	till		~
	per		%	sem		;
	crat		^	col		:
	amp		&	cam comma		,
	star		*	doot		.
	laip		(lang		<
	rye)	rang		>
	plus		+	quest		?
	noos		-	score		-
	eke		=	hive		=
	bar		—	sing		,
	lace		{	quote		”
	race		}	bing		‘
	lack		[

phonetic = *symbol* | *letter* | spooce | toob | loon

word is an English word or a *character command*, which is

phonetic (*phonetic signed*)⁺

Words may be followed by a *common-pos*; in that case, they are inserted at *common-pos*, not at cursor. The *phonetics* loon or spooce can only occur by themselves or with a *signed*. toob can be combined only with loon or a *signed*.

Keys, capitalization, and spacing

spooce <i>positive</i> ?	<space> <i>n</i> times
loon <i>positive</i> ?	<enter> <i>n</i> times
choook <i>positive</i> ?	<backspace> <i>n</i> times
chawi <i>positive</i> ?	<delete> <i>n</i> times
gloof <i>positive</i> ?	<left> <i>n</i> times
graif <i>positive</i> ?	<right> <i>n</i> times
goop <i>positive</i> ?	<up> <i>n</i> times
gnaith <i>positive</i> ?	<down> <i>n</i> times
toob <i>positive</i>	<tab> <i>n</i> times
speece	no space
capi	capitalize
coomel	camel, i.e. no space and capitalize

Common positions

<i>common-rng</i> =	tisk	between cursor and mouse
	tat	selected region (between cursor and mark)
<i>common-pos</i> =	loost	where last touched before cursor
	lairk	where mark is
	hare	at cursor
	tair	at mouse, don't move cursor
	gook	at mouse, move cursor

Structural designers

<i>struct-simple</i> =	contained in line
char	character
stretch	whitespace
ting	non-whitespace
word	word
eed	identifier (hyphenated words)
chunk	filenames, email addresses,...
tier	a line (w/o. terminator)
line	a line (w. terminator)

<i>struct-complex</i> =	may span lines
term	an identifier or block
inner	the inside of enclosing block
block	an enclosing block
defi	a definition
senten	sentence
parra	paragraph
buffer	text in buffer

<i>struct</i> =	<i>struct-simple</i> <i>struct-complex</i>	whole structure
<i>struct-rng</i> =	<i>struct signed</i> ?	forwards (<i>positive</i>)/backwards (<i>negative</i>)

Search range designers

<i>search-rng</i> =	(rorch	backward
	sorch)	forward
	words	

Search position designers

<i>search-pos</i> =	pen?	penultimate
	(aift	after and forward
	baif	before and forward
	ooft	after and backward
	boof)	before and backward
	words	

<i>pos</i> =	<i>common-pos</i> <i>search-pos</i>
stroop	end of argument, neutral word

Commands, except character commands and *speece*, terminate argument.

Cursor commands

<i>movement</i> =	nairx	go to beginning of structures [“next”] and
	noorx	go to end of structures
	skaip	to end of structures (forward)
	skoop	to end of structures (backward)

<i>movement struct positive?</i>	move
<i>struct positive?</i>	move to beginning of structure (forward)
<i>struct negative?</i>	move to beginning of structure (backward)
<i>movement character positive?</i>	move to character occurrences
(ghin ex) <i>struct</i>	beginning or end (exit) of current structure
(ghin ex) <i>block positive?</i>	leave blocks, go to beginning or end
go <i>search-pos</i> ...	go to first match
mairk	set mark explicitly
gairk <i>positive?</i>	unravel: go to <i>n</i> th last mark
goork	opposite of gairk, go to unravelled mark
goost	go to loost

Text movement commands

grab on? <i>struct-rng</i>	copy text at mouse to cursor
pull on? <i>struct-rng</i>	move text at mouse to cursor
paste on? <i>struct-rng</i>	copy text at cursor to mouse
push on? <i>struct-rng</i>	move text at cursor to mouse
swap <i>struct-rng</i> ? <i>common-pos</i> ?	swap between cursor and position
trans <i>struct-rng</i>	transpose occurrence and following

Note: “on” means: replace the single *struct* at destination.

Text deletion and copy-to-clipboard commands

<i>del-copy</i> =	rem	copy and delete (“cut”)
	smack	delete, but no copy (“delete”)
	save	copy (“copy”)
	choose	highlight (“select”), no copy

<i>del-copy struct-rng common-pos?</i>	del/copy range at <i>pos</i>
<i>del-copy (common-rng search-rng)</i>	del/copy range

Note: “choose tat” highlights range between cursor and mark.

Directional deletion

kaill <i>struct positive?positive</i>	nearest words forward
reese <i>struct positive?positive</i>	nearest words backward

Yank clipboard commands

yank <i>positive?common-pos</i>	insert <i>n</i> th entry from clipboard stack
poop <i>common-pos</i>	insert and pop clipboard stack

Changes

(caip <i>struct-rng?</i>)	capitalize
laiw <i>struct-rng?</i>	lowercase
aipper <i>struct-rng?</i>	uppercase
hive <i>struct-rng?</i>	hyphenate
speece	delete spaces
space	normalize to one space
fix	apply spacing heuristics
chaiw	delete
choock	backspace
join	join line with next
lindent <i>struct-rng?</i>	left indent range
rindent <i>struct-rng?</i>	right indent range
comment <i>struct-rng?</i>	comment range
fill <i>struct-rng?</i>	fill
pound <i>struct-rng?</i>)	compound, remove spaces
<i>pos</i>	

If *struct-rng* is provided, then *pos* is optional with default “hare.”

Paired delimiter commands

<i>pair-type</i> =	par		(...)	quote		”...”
	brace		{...}	sing		’...’
	bracket		[...]	bing		‘...’

<i>pair-type</i> pair	insert delimiters, around region if possible
<i>pair-type</i> nix	delete delimiters

Window management

<i>window-op</i> =	maxi	maximize window
	mini	minimize window, choose next
	vaix	choose other window
	voox	choose other window, reverse direction
	splot	split current window

<i>window-action</i> =		
menu		show buffer menu
open		open a file
save		save file
write		write file (save as)
insert		insert file
kaill		kill buffer
direct		directory listing
switch		most other recently shown file
grow <i>positive?</i>		make window bigger
shrink <i>positive?</i>		make window smaller
up		page up
down		page down
center		center window
head		current line to window head
bottom		current line to window bottom
<i>positive</i> <i>negative</i>		down or up a fraction of page

window-op *window-action* do op and action

gaix *positive?* go to *n*th next window
 goox *positive?* go to *n*th previous window

color = red | blue | green | brown | purple | orange | yellow | pink

window-op *color* do op, select view per color

Cross-referencing commands

cross-ref-cmd = *window-op* *cross-ref* *common-pos?*

<i>cross-ref</i> =	
help	editor help
f-help	function help (programming languages)
v-help	variable help (programming languages)
f-def	function definition (programming languages)
v-def	variable definition (programming languages)
i	follow indexed reference
info	Emacs info
apropos	Emacs apropos
command apropos	Emacs command apropos

Browsing

browse *common-pos* invoke browser on URL at position
 browse direct invoke external file browser

Repetition of last command

goink *positive?* repeat last command a number of times

Repetition of command sequence

vox rec record a macro
 goink stop recording and play
 vox stop stop recording
 vox play play previous macro

E-mail support

vox read mail switch to inbox
 vox send mail send current message
 vox reply reply to message
 vox follow-up reply to all
 vox forward forward message

Specialized support

pleat dynamic completion
 c-pleat choose dynamic completion
 vox quit listen-quit
 go line... goto a specified line number
 meta toob meta-tab, complete according to mode
 meta X. meta-X, execute command
 meta N. meta-N, next
 meta P. meta-P, previous

Lisp

eval express eval-expression
 eval (*common-pos* | *struct-complex*) evaluate expression
 eval print eval-print-last-sexp

L^AT_EX

m-tech... insert command
 m-no tech *common-pos?* delete macro
 e-tech... insert environment
 e-change tech ... change environment
 e-no tech *common-pos?* delete environment tags
 e-end tech *common-pos?* close environment

<i>pair-type</i> +=	tech-quote	“...”
	tech-sing	’...’
	tech-brace	{...}
	tech-angle	\<... \>

tech label reftex-label
 tech ref reftex-reference
 tech cite reftex-citation
 tech e.m. “\emph{...}”
 tech ... various other common L^AT_EX commands

XML

snex... insert begin and end tag
 dent snex... insert indented begin and end tag
 snoox... insert empty element tag
 x-change tag ... change tag name
 x-end tag *common-pos?* close element
 x-no tag *common-pos?* delete begin and end tag
 x-quote “' | '”
 x-angle “< | >”
 x-cdata “<![CDATA[|]]>”;

Notation “?” means optional, “+” means “one or more”.