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

symbol =	clam	!	rack	]
	lat	@	slash	/
	numb	#	beck	“
	dall | dollar	$	till	~
	per	%	sem	;
	crat	^	col	:
	amp	&	cam | comma	,
	star	*	doot	.
	laip	(	loos	<
	rye	)	groot	>
	plus	+	quest	?
	nus	-	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 *phonetic*s 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 *positive*?	⟨space⟩ n times
loon *positive*?	⟨enter⟩ n times
choook *positive*?	⟨backspace⟩ n times
chaiw *positive*?	⟨delete⟩ n times
gloof *positive*?	⟨left⟩ n times
graif *positive*?	⟨right⟩ n times
goop *positive*?	⟨up⟩ n times
gnaith *positive*?	⟨down⟩ n times
toob *positive*	⟨tab⟩ n times
speece	no space
capi	capitalize
coomel	camel, i.e. no space and capitalize

Common positions

common-rng =	tisk |	between cursor and mouse
	tat	selected region (between cursor and mark)
common-pos =	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 designators

struct-simple =		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)

struct-complex =		may span lines
term	|	an identifier or block
inner	|	the inside of enclosing block
block	|	an enclosing block
defi	|	a definition
senten	|	sentence
para	|	paragraph
buffer		text in buffer

struct =	*struct-simple* | *struct-complex* whole structure
struct-partial=	(prin | prex) *struct* partial, to beginning (prin)/end (prex)
struct-rng=	*struct signed* ? | forwards (*positive*)/backwards (*negative*) *struct-partial*

Search range designators

search-rng =	(rorch |	backward
	sorch)	forward
	words	

Search position designators

search-pos =	pen?	penultimate
	(aift |	after and forward
	baif |	before and forward
	ooft |	after and backward
	boof)	before and backward
	words	
pos =	*common-pos* | *search-pos*	
stroop		end of argument, neutral word

Commands, except character commands and speece, terminate argument.

Cursor commands

movement =	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)

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

Text movement commands

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

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

Text deletion and copy-to-clipboard commands

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

del-copy struct-rng common-pos?	del/copy range at *pos*
del-copy (*common-rng* | *search-rng*)	del/copy range

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

Directional deletion

kaill *struct positive?positive*	nearest words forward
reese *struct positive?positive*	nearest words backward

Yank clipboard commands

yank *positive*?*common-pos*	insert nth entry from clipboard stack
pop *common-pos*	insert and pop clipboard stack

Changes

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

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

Paired delimiter commands

pair-type =	par	|	(. . .)	quote	|	" . . . "
	brace	|	{ . . . }	sing	|	' . . . '
	bracket	|	[. . .]	bing		` . . . '

pair-type pair	insert delimiters, around region if possible
pair-type nix	delete delimiters

Window management

window-op =	maxi	maximize window
	mini	minimize window, choose next
	vaix	choose other window
	voox	choose other window, reverse direction
	sploot	split current window

window-action =

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 *positive*?	|	make window bigger
shrink *positive*?	|	make window smaller
up	|	page up
down	|	page down
center	|	center window
head	|	current line to window head
bottom	|	current line to window bottom
positive | *negative*		down or up a fraction of page

window-op window-action	do op and action
gaix *positive*?	go to nth next window
goox *positive*?	go to nth 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*?

cross-ref =

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

LaTeX

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

pair-type +=	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 LaTeXcommands

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".