Caster Quick Reference

1. Numbers

numb <0...1m>	**print digits**
word number <0...9>	**print as words**

2. Alphabet[a]

arch	goof	mike	sierra	yankee
brov	hotel	novakeen	tango	zulu
char	india	oscar	uniform	
delta	julia	prime	victor	
echo	kilo	quebec	whiskey	
foxy	lima	romeo	x-ray	

3. Punctuation[d]

tabby	TAB	period | dot	.
shin tabby	SHIFT-TAB	bam [c]	.
ace	SPACE	slash	/
clamor	!	deckle	:
chocky	"	semper	;
quotes [b]	""	angle [bf]	<>
hashtag	#	[e] equals	=
dolly	$	questo	?
modulo	%	atty	@
ampersand	&	brax [bf]	[]
chicky | apostrophe	'	backslash	\
thin quotes [b]	''	carrot	^
prekris [bf]	()	underscore	_
starling	*	ticky	~
[e] plus	+	tickris [b]	``
comma	,	curly [bf]	{}
boom [c]	,	tilde	~
[e] minus	-	pipe (sim | symbol)	|

4. Comparisons

[is] greater than	>
[is] less than	<
[is] greater [than] [or] equal [to]	>=
[is] less [than] [or] equal [to]	<=
[is] equal to	==

5. Directions

sauce [d]	**up**
dunce [d]	**down**
lease [d]	**left**
ross [d]	**right**

6. Editing

clear [d]	**bkspc**
deli [d]	**del**
shock [d]	**enter**
cancel	**escape**
save	**ctrl - s**

7. Selection, movement

shackle	**select line**
shin [a] [b]	**shift - <dir>**
queue [a] [b]	**ctrl - shift - <dir>**
fly [a] [b]	**ctrl - <dir>**
splat [a] [b]	**ctrl - del/bkspc**
[a] wally	**home/end**
tell sink	**end + enter**
tell dock	**end + semicolon + enter**
kraken	**ctrl + space**
hug <enclose> [e]	**encloses target**

8. Selection, movement (shorter words)

latch	**end**
ratch	**home**
bird	**ctrl-left**
fird	**ctrl-right**
brick	**shift-left**
frick	**shift-right**
blitch	**ctrl shift - left**
flitch	**ctrl shift - left**

9. Modifier Keys [d]

(control \| fly) <Key>
(shift \| shin) <Key>
alt <Key>
(control shift \| que) <Key>
(control alt \| culter) <Key>
(shift alt \| alt shift) <Key>

11. Programming core[a]

enable <language>	**activate module**

iffae	lodge and
shells	lodge or
shell iffae	lodge not
switch	print to console
breaker	function
default	class
for each	add comment
for loop	long comment
do loop	value not
while loop	return
convert to integer	value true
convert to floating-point	value false
convert to string	

12. Text formatting
Capitalisation

yell	**SOME WORDS**
tie	**Some Words**
gerrish [b]	**someWords**
sing	**Some words**
laws	**some words**

Spacing

gum | gun	**somewords**
spine	**some-words**
snake [b]	**some_words**
pebble	**some.words**
incline	**some/words**
dissent | descent	**some\words**

Capitalisation and spacing can be combined into a single command. They may be suffixed by "bow" and then dictation to format (ex: "tie snake bow some words") to aid recognition.

---

[a]Prefix with "big" for capitals
[b]Left key is pressed after
[c]Space bar is pressed after
[d]Repeatable by saying number after
[e]Prefix with "long" to get spaces either side
[f]Left or right available by prefixing with "left" or "right"

---

[a]See the CCR languages quick reference in the docs for details of the language specific commands.
[b]gerrish defaults to gum, snake to laws

13. Format setting

set [big] format <capitalisation> <spacing> [bow]
Set a standard/[secondary] format

[big] format <dictation> ....
**Apply the standard/[secondary] format
to dictation**

14. Mouse replacement

kick .... **left click**
kick mid .... **middle click**
psychic .... **right click**
shift right click .... **shift right click**
shift click .... **shift click**
squat .... **left button down**
bench .... **left button up**
lean .... **right button down**
hoist .... **right button up**
colic .... **control left click**
scree [a] .... **scroll wheel**
curse .... **move cursor by pixels**
douglas .... **move cursor on grid**
rainbow .... **alternate grid**
legion .... **smart text selection**
left point .... **mouse to screen left**
center point ...... **mouse to screen center**
right point .... **mouse to screen right**

---
[a] Takes optional direction parameter, one of the direction keys from table (5)

15. Template-based navigation

jump in [b] ...... **move cursor inside next ([{<**
jump out [b] ...... **move cursor past next)]}>**
jump back [b] ... **move cursor inside prev)]}>**
fill <target> [b] .... **highlight <target> in line**

16. Window management

minimise .... **minimise window**
maximise .... **maximise window**
remax .... **force maximise**
dredge [a] .... **alt + tab**
switch window .... **display all windows**
move window .... **move with arrow keys**
window left .... **snap window to left**
window right .... **snap window to right**
show work .... **show active workspaces**
new work .... **create a new workspace**
close work .... **close the current workspace**
close all work .... **close all workspaces**
previous work [a] **go to the previous workspace**
next work [a] .... **go to the next workspace**
go work <n> ..... **go to workspace number n**
send work <n> . **send the current window to
workspace n**
move work <n> **move the current window to
workspace n**

---
[b] All of these are asynchronous, and can be cancelled with the word "cancel" if the search is taking too long. See *navigation.py* for a full list of targets.
[a] Takes an optional number parameter for repetition

17. Repetition

again do .... **repeat last dictation**
again <n> times ...... **repeat last dictation*n**
record from history ...... **open macro recorder**
complete .... **save macro**
delete recorded macros **delete recorded macros**

18. Aliasing

alias <dict> .... **alias highlighted text**
chain alias <dict> .... **alias as CCR**

19. Bring me

program/website/folder/file to bring me as
<key> .... **create a new bring me binding**
bring me <key> ...... **open bring me binding**
remove <key> from bring me ....
restore bring me defaults ....

20. Dragon

number/spell/dictation/normal/command
mode .... **switch dragon modes**

21. Update and Caster Management

clear caster log .... **Clears Log Window**
update caster .... **Updates Caster**
update dragonfly .... **Updates Dragonfly**
reboot dragon .... **Restarts Caster**
reboot windows speech recognition .... **Restarts
Caster**

22. Example command chains

shackle clear ....
Select and delete the current line

queue lease stoosh three ....
Store the previous word/variable name in position three

queue lease three hug brax ....
Enclose the previous three elements in square brackets

shackle cut dunce four spark ....
Cut the current line, move down four, paste

shin ross wally clear ....
Select to the end of the line, delete

ross wally clear four ....
Go to the end of the line, backspace four

snake variable name ....
variable_name

curly quotes arch ross deckle quotes brov ....
{"a":"b"}

backslash format command curly ....
\command{}

(python) tell sink print to console spark three ....
new line, print(<third clipboard position>)