Talon Voice Commands

August 31, 2020

M.A. Anwar

Contents

1 Introduction **2**

2 Commands **2**

2.1 Core . . . 3

2.1.1 keys.talon . . . 3

2.1.2 abbreviate.talon . . . 3

2.1.3 repeater.talon . . . 3

2.1.4 mouse.talon . . . 3

2.1.5 formatters.talon . . . 4

2.1.6 generic_editor.talon . . . 4

2.1.7 dictation.talon . . . 6

2.1.8 modes.talon . . . 6

2.2 Math . . . 7

2.2.1 lyx.talon . . . 7

2.3 Programming . . . 9

2.3.1 comment.talon . . . 9

2.3.2 operators.talon . . . 10

2.3.3 programming.talon . . . 10

3 Lists/Captures **11**

3.1 abbreviation . . . 11

3.2 any . . . 14

3.3 arrow . . . 14

3.4 format_text . . . 14

3.5 formatters . . . 14

3.6 function . . . 15

3.7 key . . . 15
3.8 letter . . . 15
3.9 letters . . . 15
3.10 mathfly_fractions . . . 16
3.11 modifiers . . . 16
3.12 number . . . 16
3.13 number_small . . . 16
3.14 ordinals . . . 16
3.15 spaceText . . . 16
3.16 special . . . 16
3.17 symbol . . . 17
3.18 text . . . 18
3.19 tex_greek_letters . . . 18
3.20 tex_maths . . . 19

1 Introduction

The goal of this document is to outline the functionality of different voice commands that will be used in an exam setting. Descriptions of commands primarily document the text outputted by a command rather than other functionality that that command may have. For example many of the programming commands may add a trailing space at the end of a word or reposition the cursor which provides utility when dictating with one's voice but may not necessarily have been documented.

2 Commands

Each sub-sub-heading in this section corresponds to the file with the same name in the repository. Some commands accept lists of possible inputs, such lists have been linked to and documented in the next section of the document. They are denoted by surrounding < > or { }. Most of these lists contain commands to press keys however some lists may be of interest such as the list of possible LaTeX commands. I have tried to provide a brief explanation of the functionality provided by each group of commands. Many listed commands may have some syntax associated with them, for example you will see symbols such as "| [] $ ^", these symbols identify whether saying certain words is optional or they identify a need to pause before or after saying a command and are of no consequence to the output of the command itself and as such the meaning of these symbols has not been documented.

2.1 Core

2.1.1 keys.talon

Provides the functionality of being able to press any key on a keyboard and inserting numbers.

Command	Function
<letter>	press a letter key
go <arrow>	press an arrow key
(ship | uppercase) <letters> [(lowercase | sunk)]	insert upper case letters
<symbol>	insert a symbol
<function>	press a function key
<special>	press a special key
<key>	press a key
<number>	insert a number

2.1.2 abbreviate.talon

Supports the insertion of common abbreviated forms of text. A typical example is abbreviating "integer" as "int" which is otherwise difficult to dictate because "int" is not a word that would be recognized by dictation.

Command	Function
(abbreviate|abreviate|brief) <abbreviation>	insert abbreviated form of word

2.1.3 repeater.talon

Provides functionality for repetition of commands. A typical example is "go right fifth" which will move the cursor to the right five times.

Command	Function
<ordinals>	repeat previous command specified amount of times
(repeat that|twice)	repeat previous command once
repeat that <number_small> [times]	repeat previous command specified amount of times

2.1.4 mouse.talon

Provides functionality to control the mouse.

Command	Function
righty	right click
<modifiers> touch	left click while pressing modifier
<modifiers> righty	right click while pressing modifier
(dubclick | duke)	double click
(tripclick | triplick)	triple click
wheel down	scroll down

touch	left click
wheel tiny [down]	scroll down a tiny amount
wheel up	scroll up
wheel tiny up	scroll up a tiny amount
wheel left	Scroll left
wheel tiny left	scroll left a tiny amount
wheel right	scroll right
wheel tiny right	scroll right a tiny amount
drag	use cursor to drag
* pop sound	left click

*The pop sound allows for controlling the mouse with noise.

2.1.5 formatters.talon

Provides functionality to insert text "as-is" and to insert text modified by formatters. Text modified by formatters can be useful for both prose and when writing code. An example is the formatter "sentence" to insert text with the first word capitalized which is useful when writing prose. Another example when writing code is the formatters "snake" i.e. the command, "snake variable name" which would output "variable_name". Without this formatter the example would otherwise have to be dictated as "phrase variable underscore phrase name" which quickly becomes quite cumbersome to dictate. An example of the output of each formatter has been provided at formatters.

Command	Function
(say | speak | phrase) <text>$	insert phrase
(say | speak | phrase) <text> over	insert phrase
<format_text>$	insert formatted text
<format_text> <symbol>$	insert formatted text followed by symbol
format_text> over	insert formatted text
phrase <text>$	insert phrase
phrase <text> <symbol>$	insert phrase followed by symbol
phrase <text> over	insert phrase
(say | speak) <spaceText>$	insert phrase with space at the end
(say | speak) <spaceText> over	insert phrase with space at the end
format help	list formatters with examples of use
format recent	list recently formatted phrases
format repeat <number>	repeat recently formatted phrase by number representing how recent it was
format copy <number>	copy recently formatted text to clipboard
^nope that$	delete formatted text
^nope that was <formatters>$	delete formatted text and replace with new formatted text

2.1.6 generic_editor.talon

Provides utility commands to support text editing.

Command	Function

go word left	move cursor one word left
go word right	move cursor one word right
go left	move cursor left
go right	move cursor right
go up	move cursor up
go down	move cursor down
go line start	move cursor to start of line
go line end	move cursor to end of line
go way left	move cursor to the leftmost position
go way right	move cursor the rightmost position
go way down	move cursor to the bottom of the document
go way up	move cursor to the top of the document
go page down	move page down
go page up	move page up
select all	select all text
select left	select to the left
select right	select to the right
select up	select upwards
select down	select downwards
select word left	select one word left
select word right	select one word right
select way left	select everything towards the beginning of the line
select way right	select everything towards the end of the line
select way up	select everything towards the beginning of the document
select way down	select everything towards the end of the document
(indent less | out dent)	decrease indenting
clear line	delete line
clear left	delete one unit to the left
clear right	delete one unit to the right
clear up	delete line above
clear down	delete line below
clear word left	delete toward to the left
clear word right	delete toward to the right
clear way left	delete to the beginning of the line
clear way right	delete to the end of the line
clear way up	delete until the beginning of the document
clear way down	delete until the end of the document
copy all	copy everything
copy left	copy to the left of the cursor
copy right	copy to the right of the cursor
copy up	copy to the beginning of the document
copy down	copy to the end of the document
copy word left	copy one word to the left
copy word right	copy one word to the right
cut everything	cut everything from the document
cut word left	cut one word of the left
cut word right	cut one word to the right
lend	move cursor to the beginning of the line
bend	move cursor to the end of the line

2.1.7 dictation.talon

By default Talon interprets anything spoken as a command. Sometimes it is useful for dictation to be immediately interpreted as text rather than as a command. The following commands serve this functionality under dictation mode.

Command	Function
<text>	output spoken text
enter	press enter key
period	.
(comma | kama):	,
question mark	?
(bang | exclamation [mark])	!
dash	-
colon	:
go up <number_small> lines	move cursor up specified amount of lines
go down <number_small> lines	move cursor down specified amount of lines
go left <number_small> words	move cursor left specified amount of words
go right <number_small> words	move cursor right specified amount of words
select left <number_small> words	select left specified amount of words
select right <number_small> words	select right specified amount of words
select left <number_small> characters	select left specified amount of characters
select right <number_small> characters	select right specified amount of characters
clear left <number_small> words	delete specified amount of words to the left
clear right <number_small> words	delete specified amount of words to the right
clear left <number_small> characters	delete specified amount of characters to the left
clear right <number_small> characters	delete specified amount of characters to the right
format selection <formatters>$	format selected text according to specified formatter
scratch that	delete previous utterance

2.1.8 modes.talon

Provides commands to switch between different modes and enable different functionality. By default Talon functions under command mode.

Command	Function
talon sleep	put talon on standby
talon wake	enable talon
dictation mode$	enable dictation mode
command mode$	enable command mode
force java$	enable programming commands
clear language modes$	disable programming commands

2.2 Math

2.2.1 lyx.talon

Provides commands used to dictate math with LyX, a document processor that utilizes LaTeX.

Where a command does not accept input but it is useful to have text or a number to illustrate the output of the command, the text "sample" or the variables n, m have been used. On the other hand where a command accepts input, the variables x, y have been used.

Command	Function
<tex_maths>	insert associated LaTeX symbol
greek <tex_greek_letters>	insert associated Greek letter
<number> <mathfly_fractions>	insert a fraction, e.g "one half " corresponds to $\frac{1}{2}$
matrix one by <number>	insert $1 \times x$ matrix surrounded with square brackets
matrix <number> by one	insert $x \times 1$ matrix surrounded with square buckets
matrix <number> by <number>	insert $x \times x$ matrix surrounded with square brackets
cross <number> by <number>	insert $x \times x$ metric surrounded with vertical lines
fraction	$\frac{n}{m}$
over	$\frac{n}{m}$, used when the numerator has already been dictated
(super script | to the power)	allow writing a superscript
sub script	allow writing a subscript
squared	n^2
cubed	n^3
(prekris | parens | brackets | parents)	(n)
(brax | square bracket)	$[n]$
curly [brackets]	$\{n\}$
absolute	$\|n\|$
add matrix row	add an additional row to a matrix
(delete | remove) matrix row	remove a row from a matrix
add matrix column	add an additional column to a matrix
(delete | remove) matrix column	remove a column from a matrix
accent tilde	$\tilde{n}$
accent dote	$\dot{n}$
accent double	(n)
accent bar	$\bar{n}$
accent vector	$\vec{n}$
blank summation	$\sum_{m}$
summation	$\sum_{m}^{n}$
big union	$\bigcup_{n}$
blank product	$\prod_{m}$
product	$\prod_{n}$
limit	$\lim_{n}$
blank limit	lim
label above	$\overset{n}{m}$
label below	$\underset{m}{n}$

prime	$'$
degrees	$^\circ$
exponential	$\exp(n)$
expectation	$E(n)$
variance	$Var(n)$
real numbers	$\mathbb{R}$
complex numbers	$\mathbb{C}$
integer numbers	$\mathbb{Z}$
rational numbers	$\mathbb{Q}$
natural numbers	$\mathbb{N}$
new file	open a new file in LyX
open file	open "open file" dialogue
save as	open "save file as" dialogue
math	enter math mode
display mode	enter display mode
normal mode	enter normal mode
view PDF	view the PDF
update PDF	compile the PDF
next tab	go to the next tab
(prior | previous) tab	go to the previous tab
close tab	close the tab
move line up	move one lineup
move line down	move one line down
insert (in line formula | in line)	insert in line formula
insert (numbered formula)	insert numbered formula
insert (display formula | display)	insert display formula
insert equation array	insert equation array
insert (AMS align environment | AMS align)	insert AMS align environment
insert AMS align at [environment]	insert AMS align environment
insert AMS flalign [environment]	insert AMS flalign environment
insert (AMS gathered environment | AMS gather)	insert AMS gathered environment
insert (AMS multline [environment]| multiline)	insert AMS multline environment
insert array [environment]	insert array environment
insert (cases [environment] | piecewise)	insert cases environment
insert (aligned [environment] | align)	insert aligned environment
insert aligned at [environment]	insert aligned environment
insert gathered [environment]	insert gathered environment
insert split [environment]	insert split environment
insert delimiters	insert delimiters
insert matrix	insert matrix
insert macro	insert macro
insert [bulleted] list	insert bulleted list
insert numbered list	insert numbered list
insert description	insert description
insert part	insert part
insert (section | heading)	insert section heading
insert sub (section | heading)	insert sub section heading
insert sub sub (section | heading)	insert sub sub section heading
insert paragraph	insert paragraph

under (<letter>)	insert the specified letter as a subscript
under (<number>)	insert the specified number as a subscript
<letter> under (<number>)	insert the specified letter with the specified number as a subscript
super (<number>)	insert the specified number as a superscript
super (<letter>)	insert the specified letter as a superscript
transpose	T
trace	tr
line right	draw a vertical line through a matrix right of the cursor
line left	draw a vertical line through a matrix left of the cursor
inverse	$^{-1}$
operation	$\overset{n}{\sim}$
operations	$\overset{n}{\underset{m}{\sim}}$
row <number>	R_1
not equal	$\neq$
<letter> by <letter>	x x y
hat <letter>	$\vec{x}$
zero [matrix | vector]	$\vec{0}$
state null	null
state column	col
call	$\mathcal{N}$, used to write calligraphic letters
oh	0
math transformation	$T : \mathbb{R}^n \longrightarrow \mathbb{R}^m$
state span	sp
state done	■
one to one	1-1
determine <letter>	$\det(X)$
determine	$\det(n)$
cases	$\begin{cases} n & m \end{cases}$
huge zero	$\Huge 0$
part real	Re
part imagine	Im
argument	Arg

2.3 Programming

2.3.1 comment.talon

Provides functionality to facilitate inserting comments when writing code.

Command	Function
comment	insert //
comment line	insert // at the beginning of line
comment <text>$	insert // along with text

2.3.2 operators.talon

Provides functionality to facilitate inserting operators when writing code.

Command	Function
op subscript	[]
op (equals | assign)	=
op (minus | subtract)	-
op (minus | subtract) equals	-=
op (plus | add)	+
op (plus | add) equals	+=
op (times | multiply)	*
op (times | multiply) equals	*=
op divide	/
op divide equals	/=
op mod	%
op mod equals	%
(op (power | exponent) | to the power [of])	**
(op | is) equal	==
(op | is) not equal	!=
(op | is) (greater | more)	>
(op | is) (less | below) [than]	<
(op | is) greater [than] or equal	>=
(op | is) less [than] or equal	<=
(op | logical) and	&&
(op | logical) or	||

2.3.3 programming.talon

Provides functionality to facilitate inserting common statements/expressions used when writing code particularly geared towards Java for now.

Command	Function
state if	if
state else	else
state else if	else if
state self	self
self dot	self.
state while	while
state for	for
state switch	switch
state case	case
state do	do
state return	return
state import	import
state class	class
state (no | nil | null)	null
push <symbol>	move to end of line, insert symbol and go to next line

append <symbol>	move to end of line and insert symbol
kick	insert comma followed by space

3 Lists/Captures

3.1 abbreviation

Input	Abbreviated Form
address	addr
administrator	admin
advance	adv
advanced	adv
alternative	alt
application	app
applications	apps
argument	arg
arguments	args
attribute	attr
attributes	attrs
authenticate	auth
authentication	auth
binary	bin
button	btn
centimeter	cm
char	chr
character	char
class	cls
command	cls
comment	cmt
compare	cmp
config	cfg
configuration	cfg
context	ctx
control	ctrl
copy	cpy
database	db
debug	dbg
define	def
definition	def
description	desc
develop	dev
development	dev
dictation	dict
dictionary	dict
direction	dir
directory	dir
document	doc

documents	docs
double	dbl
dupe	dup
duplicate	dup
dynamic	dyn
enumerate	enum
environment	env
escape	esc
example	ex
exception	exc
execute	exec
expression	exp
extend	ext
extension	ext
framework	fw
function	func
image	img
information	info
initialize	init
initializer	init
instance	inst
integer	int
iterate	iter
java archive	jar
javascript	js
jump	jmp
keyboard	kbd
keyword arguments	kwargs
keyword	kw
kilogram	kg
kilometer	km
language	lng
latex	tex
library	lib
markdown	md
message	msg
milligram	mg
millisecond	ms
miscellaneous	misc
mount	mnt
nano second	ns
number	num
object	obj
okay	ok
package	pkg
parameter	param
parameters	params
pico second	ps
pixel	px

point	pt
pointer	ptr
position	pos
previous	prev
property	prop
public	pub
python	py
query string	qs
random	rnd
receipt	rcpt
reference	ref
references	refs
register	reg
regular expression	regex
regular expressions	regex
represent	repr
representation	repr
request	req
return	ret
revision	rev
ruby	rb
service pack	sp
session id	sid
shell	sh
source	src
special	spec
specific	spec
specification	spec
specify	spec
standard in	stdin
standard out	stdout
standard	std
string	str
structure	struct
synchronize	sync
synchronous	sync
system	sys
table of contents	toc
table	tbl
technology	tech
temporary	tmp
text	txt
token	tok
user	usr
utilities	utils
utility	util
value	val
variable	var
verify	vrfy

versus	vs
window	win

3.2 any

Any key from the following: arrow, letter, special, function, special.

3.3 arrow

Input	Output
down	down
left	left
right	right
up	up

3.4 format_text

One or more formatters followed by text.

3.5 formatters

The below table includes the name of each formatter and a sample of the text that it would output when applied to text.

Name of Formatter	Sample Output
alldown	this is a test
allcaps	THIS IS A TEST
camel	thisIsATest
dotted	this.is.a.test
dubstring	“this is a test”
dunder	__this__isatest
kebab	this-is-a-test
packed	this::is::a::test
padded	this is a test
sentence	This is a test
slasher	/this/is/a/test
smash	thisisatest
snake	this_is_a_test
string	’this is a test’
title	This is a Test

3.6 function

The function keys, F1, F2, ...

3.7 key

Any modifiers with any.

3.8 letter

Input	Output
air	a
bat	b
cap	c
drum	d
each	e
fine	f
gust	g
harp	h
sit	s
jury	j
crunch	k
look	l
made	m
near	n
odd	o
pit	p
quench	q
red	r
sun	s
trap	t
urge	u
vest	v
whale	w
tax	t
yell	y
zip	z

3.9 letters

Any combination of letter.

3.10 mathfly_fractions

Provides functionality to output fractions from spoken text such as "one fifth" which would be output as $\frac{1}{5}$.

3.11 modifiers

Input	Output
alt	alt
command	cmd
control	ctrl
option	alt
shift	shift
super	super

3.12 number

Provides the functionality to output the digits of a given number, e.g. 1000, 1000000 etc.

3.13 number_small

Provides the functionality to output the digits of a given small number (this is used because it provides faster output than number).

3.14 ordinals

Typically used for repetition, examples are second, third, fourth, fifth etc.

3.15 spaceText

Outputs given text with a space at the end.

3.16 special

Command	Key Pressed
end	end
enter	enter
escape	escape
home	home
insert	insert
pagedown	pagedown

pageup	pageup
space	space
tab	tab
delete	backspace
forward delete	delete

3.17 symbol

Command	Output
back tick	‘
comma	,
dot	.
period	.
semi	;
semicolon	;
quote	’
L square	[
left square	[
square	[
R square	]
right square	]
forward slash	/
slash	/
backslash	\
minus	-
dash	-
equals	=
plus	+
question mark	?
tilde	~
bang	!
exclamation points	!
dollar	$
dollar sign	$
down score	_
underscore	_
colon	:
paren	(
L paren	(
left paren	(
right paren	)
left brace	{
R brace	}
right brace	}
angle	<
left angle	<
less than	<
rangle	>

R angle	>
right angle	>
greater than	>
star	*
asterisk	*
pound	#
hash	#
hash sign	#
number sign	#
percent	%
percent sign	%
caret	^
at sign	@
and sign	&
ampersand	&
amper	&
pipe	|
dubquote	“
double quote	“

3.18 text

Outputs given text.

3.19 tex_greek_letters

Command	Output
alpha	α
beater	β
gamma	γ
delta	δ
epsilon	υ
zita	θ
eater	η
theta	θ
iota	ι
kappa	κ
lambda	λ
mu	μ
new	ν
zee	ξ
pie	π
row	ρ
sigma	σ
tau	τ
upsilon	υ

phi	ϕ
chi	χ
sigh	ψ
omega	ω

3.20 tex_maths

Where a command does not accept input but it is useful to have text or a number to illustrate the output of the command, the text "sample" or the variables n, mhave been used. On the other hand where is useful to illustrate the output of a command that accepts input, the variables x, y have been used.

Command	Output
fraction	$\frac{n}{m}$
summation	$\sum_{n}^{m}$
product	$\prod_{n}^{n}$
limit	$\lim$
generic root	$\sqrt[n]{n}$
integral	$\int$
double integral	$\iint$
triple integral	$\iiint$
times	$\times$
divide	$\div$
C dot	$\cdot$
plus or minus	$\pm$
partial	∂
infinity	∞
binomial	$\binom{m}{n}$
vector nabia	∇
accent hat	$\hat{n}$
accent tilde	$\tilde{n}$
accent dot	$\dot{n}$
accent double dot	$\ddot{n}$
accent bar	$\bar{n}$
accent vector	$\vec{n}$
sine	$\sin$
cosine	$\cos$
tangent	$\tan$
secant	$\sec$
cosecant	$\csc$
cotangent	$\cot$
arc sine	$\arcsin$
arc cosine	$\arccos$
arc tan	$\arctan$
hyperbolic sine	$\sinh$
hyperbolic cosine	$\cosh$
hyperbolic cotangent	$\coth$
hyperbolic tangent	$\tanh$

degree	$\deg$
determinant	$\det$
dimension	$\dim$
natural log	$\ln$
logarithm	$\log$
maximum	$\max$
minimum	$\min$
modulus	$\mod$
infimum	$\inf$
supremum	$\sup$
probability	$\Pr$
there exists	$\exists$
member [of]	$\in$
for all	$\forall$
[is] not equal [to]	$\neq$
greater or equal	$\geq$
less or equal	$\leq$
left arrow	$\leftarrow$
right arrow	$\rightarrow$
up arrow	$\uparrow$
down arrow	$\downarrow$
left right arrow	$\leftrightarrow$
maps to	$\mapsto$
oh plus	$\oplus$
oh times	$\otimes$
big oh plus	$\bigoplus$
big oh times	$\bigotimes$
dot dot dot	$\ldots$
diagonal dots	$\ddots$
horizontal dots	$\cdots$
vertical dots	$\vdots$
empty set	$\emptyset$
subset	$\subset$
superset	$\supset$
strict subset	$\subsetneq$
strict superset	$\supsetneq$
intersection	$\cap$
union	$\cup$
GCD	$\gcd$
cat hom	$\hom$
kernel	$\ker$
unit	n
unit two	$n\,\mathrm{m}$
unit fraction	n/m
text fraction	$\frac{n}{m}$
display fraction	$\dfrac{n}{m}$
text binomial	$\binom{n}{m}$

display binomial	$\binom{n}{m}$
long right arrow	$\longrightarrow$
mag	$\|$
low dots	$\ldots$
equivalent	$\leftrightarrow$
medium space	$n\,m$
proper subset	$\subseteq$
stop	$\cdot$
member	$\in$
normal text	sample
long equivalent	$\longleftrightarrow$
overline	$\overline{n}$