Hugo Coto Flórez (hugo.coto@outlook.com)

Vicel Reference Manual

Contents

Abstract

Vicel is a free open source TUI spreadsheet editor. You can read, modify and save data stored in rows and columns. It aims to be an alternative to privative non gratis well known Microsoft program, for non professional usage (as it is quite simple, al least for now).

About development

For now it is being developed by Hugo Coto as a side project. He do that because for college reasons he has to use the Windows-only mouse-centered similar and he almost went ill. The plan for the future is reach a stable version with all and no more than the useful and needed features to have an usable program for non professional usage. What I mean with no professional is that it's not planned to support any graphs neither economic formulas or such specific things. It's true that with builtins someone can adapt vicel to his own necessities.

Open Vicel

As vicel is a TUI program you have to start it from the command line. If you installed it in ~/.local/bin then you can start it as a normal terminal tool:

```
vicel filename [options]
```

If the filename is omitted it opens an unnamed sheet. Note that this sheet can't be saved. Also, you can't open more than one file at the same time.

Options

The options supported are the following:

Command	Description
`-m, --use-mouse`	Enable mouse support
`-D, --debug`	Enable debug output
`-c, --config-file`	Set custom file path

,

For example, if you want to open the file ./SHEETS/TABLE.CSV with a config file in ./CONFIG/VICEL.TOML, the command line should look like that:

```
vicel sheets/table.csv -c config/vicel.toml
```

Moving around

As a vim enthusiast, movement is keyboard centered, and use the vim default HJKL. Every action can be prefixed with a number, so it would be executed that amount of times. The following table describes the basic movement.

Command	Description
`h`, `l`, `j`, `k`	Move cursor left, right, down, up
`$`	Go to last cell of the current row
`^`	Go to first cell of the current row
`gg`	Go to first cell of the current column
`G`	Go to last cell of the current column
`g0`	Same as ^ and `gg`

,

Write or edit

For write text in a cell, move the cursor there an press `i`. A text input box would be open at the cell. After write, press ENTER to save it. The data type would be automatically calculated by the program. Every number, with or without a fractional part separated by a dot would be interpreted as a NUMBER. If the text written starts with a equal sign it would be interpreted as a formula. Other formats would be set to TEXT.

Command	Description
i	insert/modify text
d	delete cell content
v	toggle cell selection

,

The valid types are described in the following table by it's formal representation.

Type	Formal
NUMBER	[0-9]+(“.”[0-9]+)?
FORMULA	“=” FORMULA BODY
TEXT	!NUMBER && !FORMULA

,

Formula

Formulas are expressions that evaluate to a valid value. They start with a equal sign. The function body have to contain a valid expression.

Type	Description	Example
Literal	Number, text, identifier or range	see below

Type	Description	Example
Number	As cell type NUMBER	5987, 45.6
Text	Alphas or text surrounded by '	hello, '5.9'
Identifier	Cell reference by name as COLROW	A0, b5, ZZ98
Range	Cell range as ID:ID	A0:A2, A7:C8
Arithmetic operators	Evaluate arithmetic expressions	+, -, /, *, ^
Comparison operators	Compare two expressions	>, <, >=, <=, ==, !=, !
functions	Reserved names that convert some input in some output, with the form NAME(ARGS,...)	sum(A0,A1)

Todo: expand formula reference

,

Advanced write: write and move
There is a builtin feature to automatically move before insert text. It is useful if you need to input a big amount of data in a given direction. The idea is to prefix the following commands with a number, to do it for a given number of times.

Command	Description
gih, gij, gik, gil	insert text and move in the given direction

,

Modify sheet structure

There is some commands to add/delete rows and columns. Note that formula identifier would not change on row/col insertion/deletion.

Command	Description
`g#`	Add row/column: see below
`gd#`	Delete row/column: see below
`gj`	Add a new row after the cursor
`gl`	Add a new column after the cursor
`gk`	Add a new row before the cursor
`gh`	Add a new column before the cursor
`gJ`	Add a new row at the end
`gL`	Add a new column at the end
`gK`	Add a new row at the start
`gH`	Add a new column at the start
`gdj`	Delete row and move up
`gdl`	Delete column and move right
`gdk`	Delete row and move down
`gdh`	Delete column and move left

Expand cells

There is a feature to get the cell value given the previous one and a direction. Numbers add 1 and formula identifiers recalculate depending on the direction. They can avoid modification if `$` is used before identifier letter (for freeze column) or number (for freeze row). The mappings to do this are described in the following table.

Command	Description
`J`, `K`, `H`, `L`	Expand current cell down, up, left, right (and move)

Copy - Paste

As a vim user, you would want to copy-paste things around. Unfortunately, it's only possible to copy a single cell value and paste it in a single cell.

Command	Description
p	paste
y	yank (copy)

Other commands

There are another useful commands, described below.

Command	Description
q	Save and quit
w	Write (save)
r	Re-render the screen
Ctrl-c	Quit without save