

The *Hansl-Gretl-Language* package

Program like a *pro* using the Sublime editor

Artur Tarassow

Otto GmbH & Co KG

Disclaimer

The opinions expressed in this presentation and on the following slides are solely those of the presenter and not necessarily those of Otto GmbH & Co KG.

Motivation

- ▶ I work on a daily base with multiple tools and languages – **incl. gretl/hansl.**
- ▶ What I need:
 - ▶ GOTO: functions, references
 - ▶ FIND & REPLACE: local, global
 - ▶ create, delete and rename files & folders
 - ▶ project management & window tiling
 - ▶ git support & integrated terminal
 - ▶ build systems + language support: SQL, Python, gretl etc.
 - ▶ snippets & code completion
 - ▶ UDF key-bindings
 - ▶ eye-relaxing environment

```
| - PROJECT
    | - gretl
        | - config
            config_1.inp
            config_2.inp
        | - data
        | - script
            job_1.inp
            job_2.inp
        | - src
            helper_1.inp
        | - sql
            query_1.sql
            query_2.sql
        | - output
    | - python
```

The sublime editor

My major decisive criteria:

“Where is it easiest to implement an own syntax support?”

- ▶ I've found that it's easiest for the sublime editor which uses the *Language Server Protocol*:
<https://github.com/sublimelsp/LSP>
- ▶ There exist many templates for other languages.
- ▶ How-to-start:
<https://github.com/sublimelsp/LSP#getting-started>



Figure 1: Market is full of both closed- and open-source IDEs

The editor

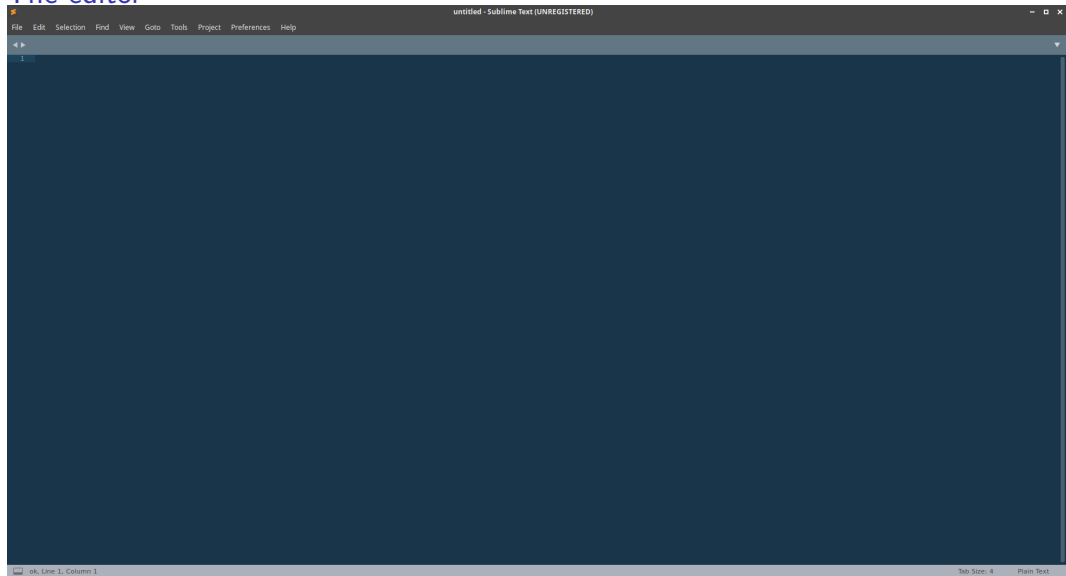


Figure 2: How it starts...

Figure 3: How it ends... after a few key-strokes

Some basic information on Sublime

- ▶ Lightweight cross-platform editor
- ▶ Add-ons make it a proper IDE (<https://packagecontrol.io/>)
- ▶ “Customize Anything”
- ▶ GOTO anything / definition (works reasonable out-of-box for gretl functions!)
- ▶ Among the top-5 editors
- ▶ homepage: <https://www.sublimetext.com/>
- ▶ Wikipedia: https://en.wikipedia.org/wiki/Sublime_Text
- ▶ Sublime version 4 has just been published! (new features: <https://youtu.be/O91i9rL8VxM>)
- ▶ You find support on anything in the web.

BUT, it's not free.

HOWEVER, one has a free indefinite preview without any functional limitations.

The “Hansl-Gretl-Language” package

- ▶ Support for *gretl*'s scripting language called *hansl*.
- ▶ The package's features are:
 - ▶ Syntax-highlighting
 - ▶ Completions
 - ▶ Snippets
 - ▶ Gretl build-systems (Batch mode, CLI, REPL)
- ▶ Version: v0.2.4
- ▶ Code: <https://github.com/atecon/Hansl-Gretl-Language>
- ▶ Package: <https://packagecontrol.io/packages/Hansl-Gretl-Language>
- ▶ Simple to install via Sublime's “Package Control” system.

Package Control

BROWSE

Hansl-Gretl-Language

by [atecon](#) 

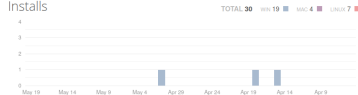
Sublime support for hansl scripting language: syntax-highlighting, auto-complete, snippets

LABELS language syntax, snippets, completions

Details

VERSION 0.2.0
HOMEPAGE [github.com](https://github.com/atecon/Hansl-Gretl-Language)
ISSUES [github.com](#)
MODIFIED 6 months ago
LAST SEEN 2 hours ago
FIRST SEEN 12 months ago

Installs



The REPL

Gretl can be executed in various modes:

- 1) Batch mode
- 2) CLI/ REPL

REPL: read-eval-print loop

==> Execute a hansl script and switch to CLI model without closing gretl

```
repl_example.inp x  Gretl CLI x
Expected a function return type, found 'type'
? open denmark
Opening a new data file closes the present one. Proceed? (y/n) y

Read datafile /usr/local/share/gretl/data/misc/denmark.gdt
periodicity: 4, maxobs: 55
observations range: 1974:1 to 1987:3

Listing 5 variables:
0) const  1) LRM  2) LRY  3) IBO  4) IDE

? print dataset -o --range=1:4

      LRM      LRY      IBO      IDE
1974:1  11.63255  5.903658  0.1547356  0.0940
1974:2  11.60415  5.873820  0.1779912  0.0955
1974:3  11.58152  5.837818  0.1705647  0.0955
1974:4  11.60185  5.812255  0.1522273  0.0955

? eval cos(33)
-0.013276747
? ols LRM const LRY IBO IDE

Model 1: OLS, using observations 1974:1-1987:3 (T = 55)
Dependent variable: LRM

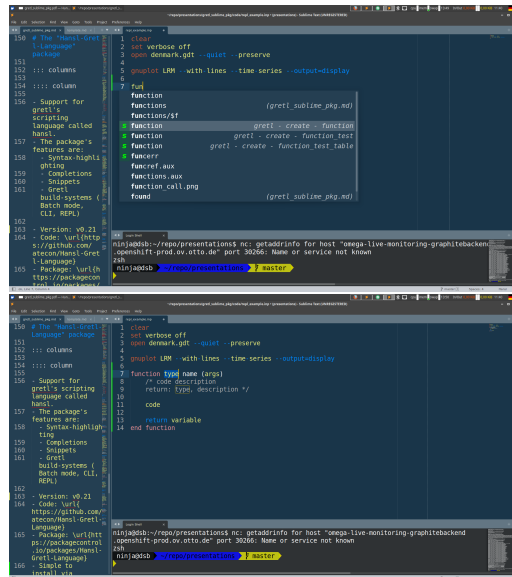
      coefficient  std. error  t-ratio  p-value
-----
const      4.39447    0.581120    7.562    7.07e-10 ***
LRY        1.29580    0.0939831   13.79    7.81e-19 ***
IBO        -2.61631    0.328191   -7.972    1.61e-10 ***
IDE         0.618564   0.691102    0.8950    0.3750

Mean dependent var  11.75441  S.D. dependent var  0.152357
Sum squared resid  0.092526  S.E. of regression  0.042594
R-squared          0.926185  Adjusted R-squared  0.921843
F(3, 51)          213.3069  P-value(F)          7.65e-29
Log-likelihood     97.61737  Akaike criterion    -187.2347
Schwarz criterion  -179.2054  Hannan-Quinn        -184.1297
rho               0.606885  Durbin-Watson        0.745003
```

Snippets

- ▶ We all use certain short fragments of text again and again...
- ▶ Snippets are smart templates that will insert text for you and adapt it to their context.
- ▶ Snippets are simple to program in Sublime – for any language or purpose.
- ▶ The “Hansl-Gretl-Language” package ships various snippets for the user (<https://github.com/atecon/Hansl-Gretl-Language/tree/master/Snippets>).

Details: <https://www.sublimetext.com/docs/completions.html#snippets>



```
1 clear
2 set verbose off
3 open denmark.gdt --quiet --preserve
4
5 gplot LRM --with lines --time series --output=display
6
7 fun
function
functions/$f
function
gretl - create - function
function
gretl - create - function_test
funcerr
funcrref.aux
functions.aux
function_call.png
found
(gretl_subline_pkg.md)

ninjadbsi:~/repo/presentations$ nc: getaddrinfo for host "omega-live-monitoring-graphitebacken
openshift-prod.ov.otto.de" port 30266: Name or service not known
$sh
ninjadbsi ~/repo/presentations > master
```

```
1 clear
2 set verbose off
3 open denmark.gdt --quiet --preserve
4
5 gplot LRM --with lines --time series --output=display
6
7 function type name (args)
8 /* code description
9 return: type, description */
10
11 code
12
13 return variable
14 end function

ninjadbsi:~/repo/presentations$ nc: getaddrinfo for host "omega-live-monitoring-graphitebacken
openshift-prod.ov.otto.de" port 30266: Name or service not known
$sh
ninjadbsi ~/repo/presentation > master
```

Snippets

<snippet>

<content><![CDATA[

function \${1:type} \${2:name} (\${3:args})

/* \${4:code description}

return: \${5:type}, \${6:description} */

\${7:code}

return \${8:variable}

end function

]]></content>

<!-- Optional: Set a tabTrigger to define how to trigger the snippet -->

<tabTrigger>function</tabTrigger>

<!-- Optional: Set a scope to limit where the snippet will trigger -->

<!-- <scope>source.inp</scope> -->

<!-- Optional: Description to show in the menu -->

<description>gretl - create - function</description>

</snippet>

Plotting

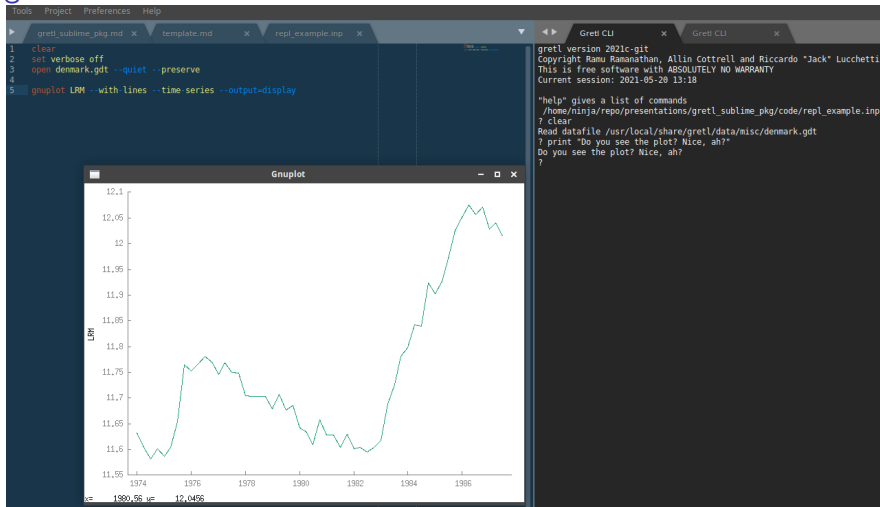


Figure 4: Plot appears instantaneously in REPL model.

Parallel execution of gretl jobs

Each executed gretl script opens a new gretlcli tab.

Hence...

- 1) Execute script SCRIPT with parameter vector A → opens a new gretlcli instance
- 2) Execute script SCRIPT with parameter vector B → opens a new gretlcli instance

.

.

.

They all run in parallel.

Known issues and missings

Known issues and missings

- ▶ The syntax highlighting is far from being perfect.

Anyone with professional regex-knowledge here in the room? ;-)

- ▶ Auto-indentation not supported, yet – but shouldn't be too complicated.
- ▶ Auto-completion of functions:
Since v0.2.3 much better completion-support for hansl commands but not yet for functions).
THIS WILL CHANGE SOON! ;-)

Wish-list

I really would like to have the possibility to have a sidebar showing gretl objects similar to **gretl's "icon view"**.

In Microsoft's (open-source) vscode editor you have, for instance, for Python a Julia an implemented **variable explorer** and **data viewer**.

Live session

Let's do a brief live session!

Please interrupt me about 4 minutes before timeout.

Recommended packages

The Sublime package repository server is: <https://packagecontrol.io>

There you can explore thousands of packages which can be installed easily through Sublime's package manager by:

ctrl+shift+p -> Package Control: Install Package

1. BracketHighlighter
2. All Autocomplete
3. Origami
4. Pandoc
5. Pretty JSON
6. rainbow_csv
7. SideBarEnhancements
8. Terminus
9. TodoReview
10. AdvancedNewFile
11. Compare Side-By-Side


Other resources

There is really nothing better than *OdatNurd's* youtube channel:

https://www.youtube.com/channel/UCJAB_XF3kAMqwF85y0hxcXQ

Thanks for your attention!

Email me: atecon@posteo.de

Follow me on  : <http://www.twitter.com/atecon1>

Follow me on  : <http://www.github.com/atecon>