

VIM-плагин для SQL и для Perl vispen

Вадим Коновалов



CPAN ID: VKON

Конференция Perl-Conf.Ru/24



Обзор

- 1 Мотивация
 - Плагин vispen 
 - Издалека
- 2 Суть плагина
 - Какие узлы у движка
- 3 Примеры использования
 - мини REPL
 - чуть-чуть перл
 - чуть-чуть SQL
 - содержательный SQL
- 4 А можно другой редактор?
- 5 Планы (TODO)



О плагине

vispen - аббревиатура от vi-sql-perl-engine

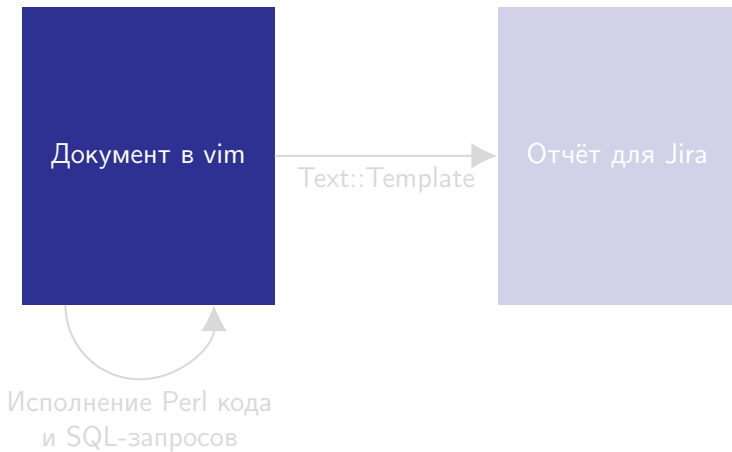
- на гитхаб выложен недавно
⇒ логотип может поменяться и, возможно, название.
- однако подход используется давно
в 2013 использовался в среде mainframe, тогда назывался 有-M-ё
- SQL-блоку - 2 года
- представляет собой 1 файл 600 строк, плюс возможно файл настроек.



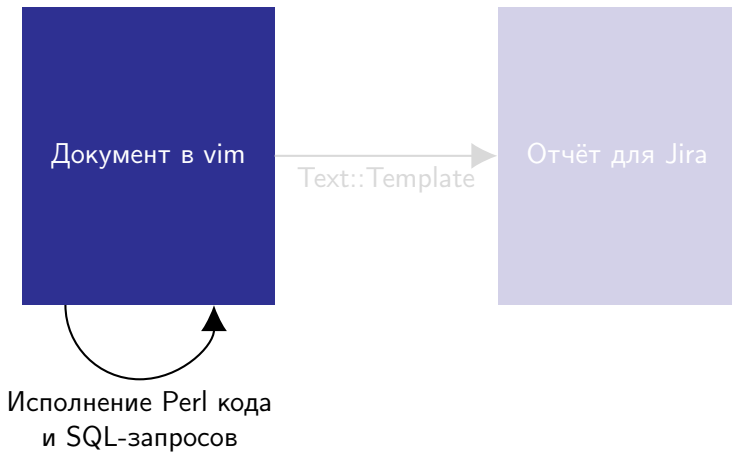
Чем может быть полезен этот плагин

- Во-первых, это REPL для Perl-a
- ..., а что более важно - это возможность задокументированно выдавать правильные последовательности shell-команд
- ..., а что ещё более важно - это помощь в создании отчётов, (помощь в JIRA-синтаксисе, либо отчёты в свободном формате)
- И главное - это удобный SQL-клиент. Плагин - для этого.

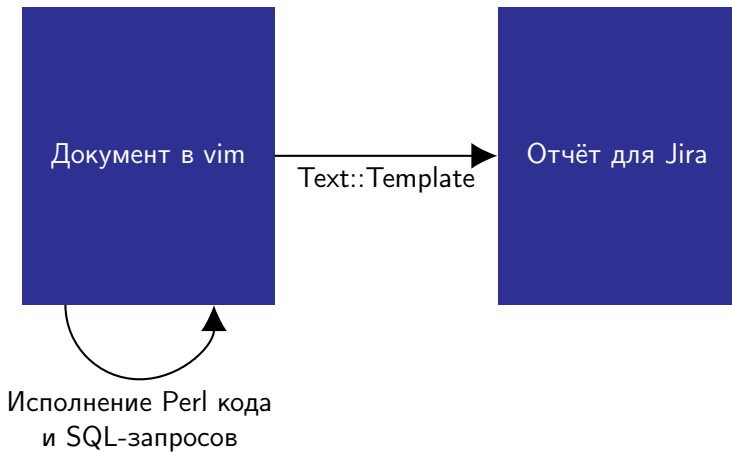
Задачи плагина



Задачи плагина



Задачи плагина



что под капотом и каков алгоритм действий

Сам плагин написан на Перле

- имеем perl🐪 и vim🚩 с поддержкой perl, версии любые
- `Text::Template` - ключевая часть. Очень.
- файл `vim-perl-sql.vim`
этот 1 файл и есть плагин. Формально он vimscript, но по сути он perl-файл
`perl -wc vim-perl-sql.vim`
github трактует `vim-perl-sql.vim` как перл-код
- можно ещё добавить несколько файлов, например `vim-perl-config.vim`, с настройками под себя либо с дополнительными функциями

(есть другие примеры, когда плагин пишется не на vimscript, YCM-youcompleteme и не только он).

Интересно, что с таким подходом получается более быстро работающий плагин, чем на vimscript.



Суть плагина (движок)

- Редактируем некий отчёт,
 - который затем пропускается через `Text::Template`¹
 - можно любой файл или даже `[No Name]` без сохранения.
- Всё, что выше курсора - прогоняется через `Text::Template`¹ это даёт возможность Perl выполнить нужный нам код
- выполнить текущую строку как Perl - `F7`, как SQL - `F9`
- блоки кода `=Perl/=Cut` и `=Sql/=Cut` игнорируются, но
- если курсор внутри - выполняется Perl-код или SQL-код
- блоки кода `=ie/=cut` игнорируются в момент "раскрытия" отчёта.
 - это позволяет внутри `=ie/=cut` иметь произвольное количество `=sql/=Cut`, которые никак не будут видны после `untemplate`
- блоки `Text::Template` - `{{{...}}}` под `no strict;` (умолчание `Text::Template`), блоки `=Perl/=Cut` - `use strict;`. Выглядит непоследовательно, но проблемой не является.

¹это опционально, регулируется опцией `$vim::untemplatep`



Какие узлы у движка

Блок,
инициализирующий
\$::dbh

Блок =ie/=cut для
работы с отчётами

SQL-запрос
в =Sql/=Cut

SQL-запрос в одной строке

несколько заготовок
перл-команд для
удобства конфига и тд

пункты меню плагина

The screenshot shows the Vim editor interface with a file named [No Name] - G2. The editor contains Perl and SQL code. A configuration menu is open, showing various settings for the plugin. The menu items are: config, untemplate, untemplate (1), execute_here, execute_here (1), execute_here (2), execute_here (5), delete_this_from_r, and suggest_columns. The menu is titled 'config' and has a 'tell method...' option. The code in the editor includes a Perl block for initializing \$::dbh, a Perl block for handling <F7> (perl execution), and a SQL block for handling <F9> (SQL execution). The SQL block includes a table with columns 'a' and 'b'.

```
[[{ # vi: syn=perl
unless ($dbh) {
    use DBI;
    $dbh = DBI->connect("dbi:SQLite:dbname=try-1.sqlite", "", "");
},
}]
=ie

# for <F7> - perl execution:
undef $::dbh
VIM::Msg('abcd', 'Comment')
VIM::Msg('efgh', 'ErrorMsg')

# конфиг:
$vim::anchorp = 0
$vim::width = 40

# for <F9> - SQL execution:
select 1 as n

n
1
=cut

=sql
select 1
union all select {{{2+3}}}
=cut

select "a" a,"b" as b union all select 'Вася', '{{{"a"x40}}}'
---select "a" a,"b" as b union all select 'Вася', 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa'
```

a	b
a	b
Вася	aa
я	aaaaa

все секции опциональны

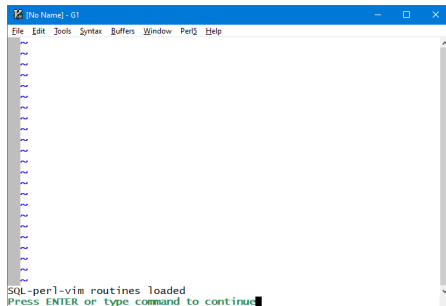
REPL - чуть подробнее

- REPL имеется у
 - Python
 - nodejs
 - julia
 - ...
 - Perl - на р5р предлагают workaround - `perl -de 0`, но признают его недостаточность
- ∃ notebook `Jupyter` для python, для Perl, у julia notebook называется `Pluton`

Плагин vispen занимает место где-то между обычными РЕПЛами и notebook, ближе к notebook.

Активизируем плагин

<F11>



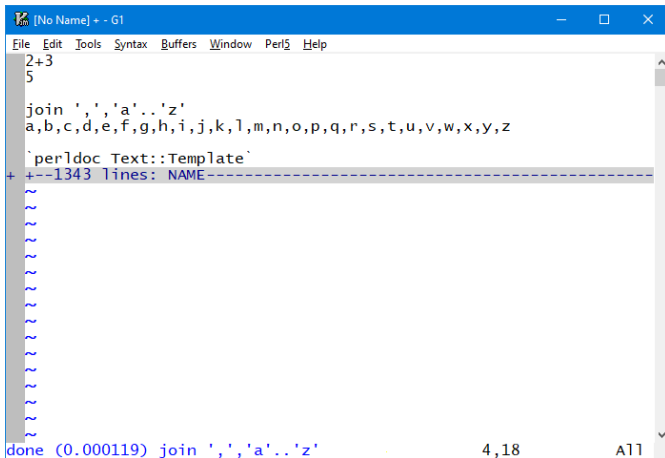
Готов REPL.

2+3<F7>

join ',','a'..'zz'



чуть-чуть перл



```
[No Name] + - G1
File Edit Tools Syntax Buffers Window Perl5 Help
2+3
5

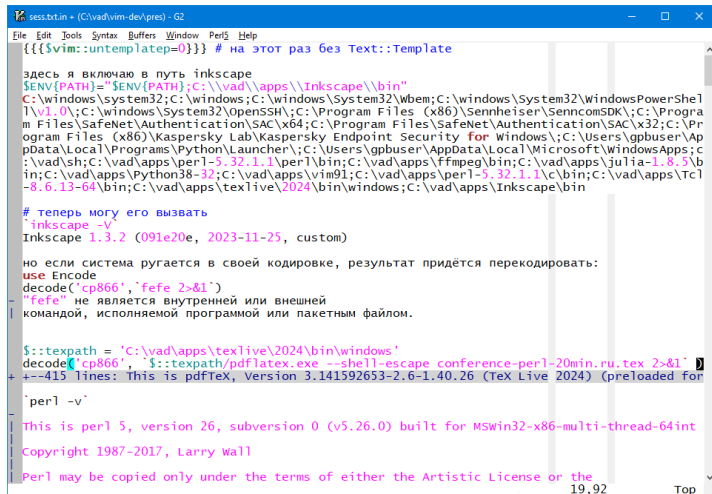
join ',', 'a'..'z'
a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z

`perldoc Text::Template`
+--1343 lines: NAME-----
~
~
~
~
~
~
~
~
~
~
~
~
done (0.000119) join ',', 'a'..'z' 4,18 A11
```

удобно посмотреть доку `perldoc Text::Template`, выполнить какие-то арифметические действия, и т.д.



REPL как документированная сессия



```
sess.txt.in - (C:\vad\vim-dev\pres) - G2
File Edit Tools Syntax Buffers Window Perl5 Help
{{{${vim::untemplatep=0}}}} # на этот раз без Text::Template

здесь я включаю в путь inkscaper
$ENV{PATH}="$ENV{PATH};C:\\vad\\apps\\Inkscape\\bin"
C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\WindowsPowerShell
\v1.0;C:\windows\System32\OpenSSH\;C:\Program Files (x86)\Sennheiser\SenncomSDK\;C:\Program
Files\SafeNet\Authentication\SAC\x64;C:\Program Files\SafeNet\Authentication\SAC\x32;C:\Pr
ogram Files (x86)\Kaspersky Lab\Kaspersky Endpoint Security for Windows\;C:\Users\gpbuser\Ap
pData\Local\Programs\Python\Launcher\;C:\Users\gpbuser\AppData\Local\Microsoft\WindowsApps;c
:\vad\sh;C:\vad\apps\perl-5.32.1.1\perl\bin;C:\vad\apps\ffmpeg\bin;C:\vad\apps\julia-1.8.5\b
in;C:\vad\apps\Python38-32;C:\vad\apps\vim91;C:\vad\apps\perl-5.32.1.1\c\bin;C:\vad\apps\Tcl
-8.6.13-64\bin;C:\vad\apps\texlive\2024\bin\windows;C:\vad\apps\Inkscape\bin

# теперь могу его вызвать
`inkscaper -v`
Inkscape 1.3.2 (091e20e, 2023-11-25, custom)

но если система ругается в своей кодировке, результат придётся перекодировать:
use Encode
decode('cp866', `fefe 2>&1`)
- "fefe" не является внутренней или внешней
командой, исполняемой программой или пакетным файлом.

$::texpath = 'C:\vad\apps\texlive\2024\bin\windows'
decode('cp866', '$::texpath\pdflatex.exe --shell-escape conference-perl-20min.ru.tex 2>&1` )
+---415 lines: This is pdfTeX, Version 3.141592653-2.6-1.40.26 (TeX Live 2024) (preloaded for
`perl -v`

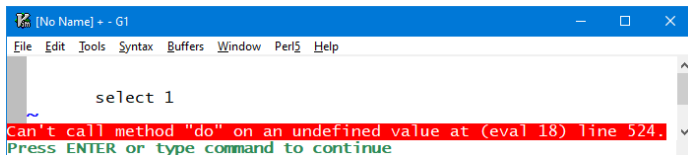
This is perl 5, version 26, subversion 0 (v5.26.0) built for MSWin32-x86-multi-thread-64int
Copyright 1987-2017, Larry Wall

Perl may be copied only under the terms of either the Artistic License or the
19,92 Top
```

результат команды помещается в fold

чуть-чуть SQL

```
select 1<F9>
```



Ошибка из-за того, что для выполнения SQL надо в `$::dbh` иметь нужный хэндлер.

Самое простое - нажать `<F9>` на пустой строке. Появится несколько строк кода, инициализирующих `$::dbh`

теперь SQL-запрос будет работать



```
[No Name] -- GS
File Edit Tools Syntax Buffers Window Perl5 Help
{[[[ # vi: syntax=perl
unless ($dbh) {
    use DBI;
    $dbh = DBI->connect("dbi:SQLite:dbname=try-1.sqlite", "", "");
}
}}]
$vim::width = 23;
23
select 1 kon1, '{{{{'a'x60}}}' kon2, 'str' kon3
---select 1 kon1, 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa' kon2, 'str' kon3

kon2 kon3
1
1 aaaaaaaaaaaaaaaaaaaaaa s
1 aaaaaaaaaaaaaaaaaaaaaa t
1 aaaaaaaaaaaaaaaaaaaaaa r

=sql
select 1 kon1, '{{{{'a'x60}}}' kon2, 'str' kon3
=Cut
[code:sql]
select 1 kon1, 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa' kon2, 'str' kon3
[code]
| |kon1|kon2|kon3|
| 1|aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa|str|

=sql имя1/f=x
select 1 kon1, '{{{{'a'x60}}}' kon2, 'str' kon3
union all select 1,2,3
=Cut
[code:sql]
select 1 kon1, 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa' kon2, 'str' kon3
union all select 1,2,3
[code]
kon1 : 1
kon2 : aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
kon3 : str
kon1 : 1
kon2 : 2
kon3 : 3
[code]

use Data::Dumper;
Dumper \%$VAR1 = {
    $VAR1 = {
        'имя1' => [
            [
                '1',
                'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa',
                'str'
            ],
            [
                '2',
                '2',
                '3'
            ]
        ]
    };
};
```



Форматы вывода SQL

- ASCII +-+
- HTML
- Синтаксис JIRA wiki
- Outwiker wiki
- построчно
- суммарно (сами строки не выводятся, только "итого")

Ограничений по диалектам SQL нет.

Насколько матёрый/глучный этот плагин

Всё работает, очень стабильно, НО

- при выполнении перл-кода можно столкнуться с тем, что не будут повторно выполняться `use Модуль;`, и если модуль поменялся, он не пересчитывается.
 - можно "заставить" пересчитать модуль при помощи `BEGIN{delete $INC{Модуль.pm}}`
 - не надо увлекаться `=Perl/=Cut`, достаточно `=Sql/=Cut` и однострочных perl-коамнд типа `'make 2>&1'`
- что-то может пойти не так на стадии `untemplate`, если имеется нетривиальный код в `{{{...}}}`
 - НО тогда виноват не плагин, а отчёт/сессия
 - тогда следует отключить `untemplate`, будет проще :)



Итого: очень надёжный



А можно другой редактор?

Сам плагин написан на Перле, поэтому - да, можно другой редактор.
Теоретически - например - sublime - через модуль `Python::Object`
Кто возьмётся делать - мне скажите.



Итого

vispen умеет:

- удобно послать SQL-запрос и получить результат
 - при этом в SQL можно вставлять перл-код и результат получать в перл-переменные
- помогать формировать Отчёт (r - Reporting)
- даёт функциональность REPL, которого в перле мало
- больше даже похоже на юпитер/перл но под vim



Направления опережающего развития (AKA TODO)

- доделать документацию на github - `README.md` - WIP
- другой редактор?
- подсветка синтаксиса внутри `=sql/=Cut`, `=perl/=Cut`
- тесты тесты тесты, как обычно
- убрать велосипед в пользу `Term::Table`



Вопросы, ссылки I

Вопросы

- ?

Ссылки

- <https://github.com/vadrer/vispen>
- vim.org, perl.com, sqlite.org, ...