Introduction to Vim 8.0

VimConf 2016

2016-11-05 Ken Takata

Who am I?

- Name: K.Takata
- Twitter: @k_takata
- GitHub: k-takata
- Web: http://k-takata.o.oo7.jp/

K.T

- Maintainer of Onigmo. (The regex engine of Ruby 2.0 and later)
- One of the most active Vim contributor in Japan.

Contents

- New features of Vim 8.0
- History of Vim
- Contributions of Japanese people to Vim

Congratulations

- Vim 8.0 was released on Sep. 12, 2016.
  - Three years from the last minor release. (7.4)
  - Ten years from the last major release. (7.0)
- More than 2300 patches from nearly 200 contributors were merged after 7.4.

V(im)8を称えよ!

New Features of Vim 8.0

- Vim script enhancements
  - Asynchronous I/O support (channels), Jobs, Timers
  - Partials, Lambda and Closure
  - 64-bit Numbers
  - Window IDs
  - New style tests
- Packages
- Wrapping lines with indent (‘breakindent’)
- Windows: DirectWrite support
- GTK+ 3 support

See `:help version8`

New Features of Vim 8.0 (contd.)

- Most of the new features are enhancement of Vim script.
- New features for end users are not so much.

- So at first Bram had been going to release it as 7.5, but finally he decided to release it as 8.0, because it is easier to understand for end users when a plugin requires new features from Vim 8.0.

Asynchronous I/O support (Channels)

- Vim can now exchange messages with other processes in the background.
- You can send a message to a server, then you can get the results from it asynchronously.
- You can use sockets or standard I/O.
- You can use JSON, linewise or raw format.

See `:help channel-demo`

Jobs and Timers

- Vim can now start a job, communicate with it and stop it.
- Channels are used to communicate with the job.

See `:help job_start()`

- Vim can now use timers.
- They can fire once or periodically.

See `:help timer_start()`

Partials, Lambda and Closure

- A Funcref optionally binds a Dictionary and/or arguments.
  This is called a Partial.
- Useful for timers and other callback functions.

See `:help Partial`

- A short way to create a function has been added: {args -> expr}.
- This is useful for functions such as `filter()` and `map()`.
- Written by Yasuhiro Matsumoto (mattn), vim-jp members and me.

See `:help lambda` and `:help closure`

64-bit Numbers

- Vim now uses 64-bit Numbers on almost platforms.
- Useful when handling filesizes (> 2GiB), timestamps and other large values.
- Written by mattn and me.

See `:help Number`

Window IDs

- Previously windows could only be accessed by their number. And every time a window would open, close or move that number changes.
- Each window now has a unique ID, so that they are easy to find.
- Useful for plugin writers.

See `:help win_getid()` and `:help win_id2win()`

New style tests

- This is for Vim developers.
- So far writing tests for Vim has not been easy.
- Vim 8 adds assert functions and a framework to run tests.
- If you write a patch for Vim, you should use them.

See `:help test-functions`

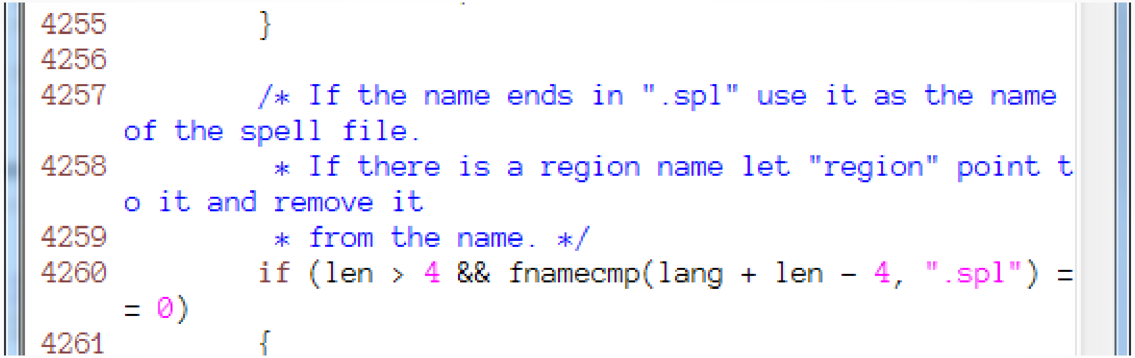
Packages

- Vim now has minimal support for managing plugins.
- If you unpack a plugin in a package directory, Vim can load it automatically (or on demand).
- This can be a replacement for pathogen.
- If you want to download plugins from GitHub and etc., maybe you still need to use a plugin manager like Vundle, NeoBundle, vim-plug, etc.

See `:help packages`

Wrapping lines with indent (‘breakindent’)

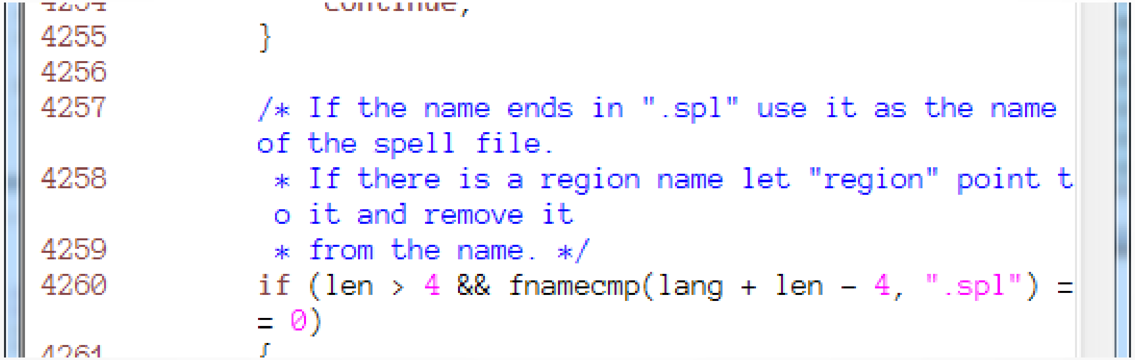
Off



```
        }

        /* If the name ends in ".spl" use it as the name
of the spell file.
         * If there is a region name let "region" point t
o it and remove it
         * from the name. */
        if (len > 4 && fnamecmp(lang + len - 4, ".spl") =
= 0)
        {
```

On



```
            continue;
        }

        /* If the name ends in ".spl" use it as the name
        of the spell file.
         * If there is a region name let "region" point t
         o it and remove it
         * from the name. */
        if (len > 4 && fnamecmp(lang + len - 4, ".spl") =
        = 0)
        {
```

From my slides
of VimConf 2013

Wrapping lines with indent (contd.)

- The 'breakindent' option has been added to be able to wrap lines without changing the amount of indent.

- The first version was written in 2004.
- It took ten years to be merged. (There were some problems; coding style problems, lack of tests.)
- Tests are very important to be merged.

See `:help ’breakindent’`

DirectWrite support on Windows

- Better text rendering on Windows
  - Better rendering quality
  - Better multilingual support
- Written by Taro Muraoka (koron) and me.
- See also: https://www.kaoriya.net/blog/2013/05/05/ (in Japanese)

See `:help ’renderoptions’`

- TODO: Using Direct2D
  - Currently it uses GDI compatible mode. So it is slower than using Direct2D.
    - Copy the background to a buffer, render with DirectWrite, then copy back to GDI buffer.
  - Color emojis are not supported on GDI compatible mode.

GTK+ 3 support

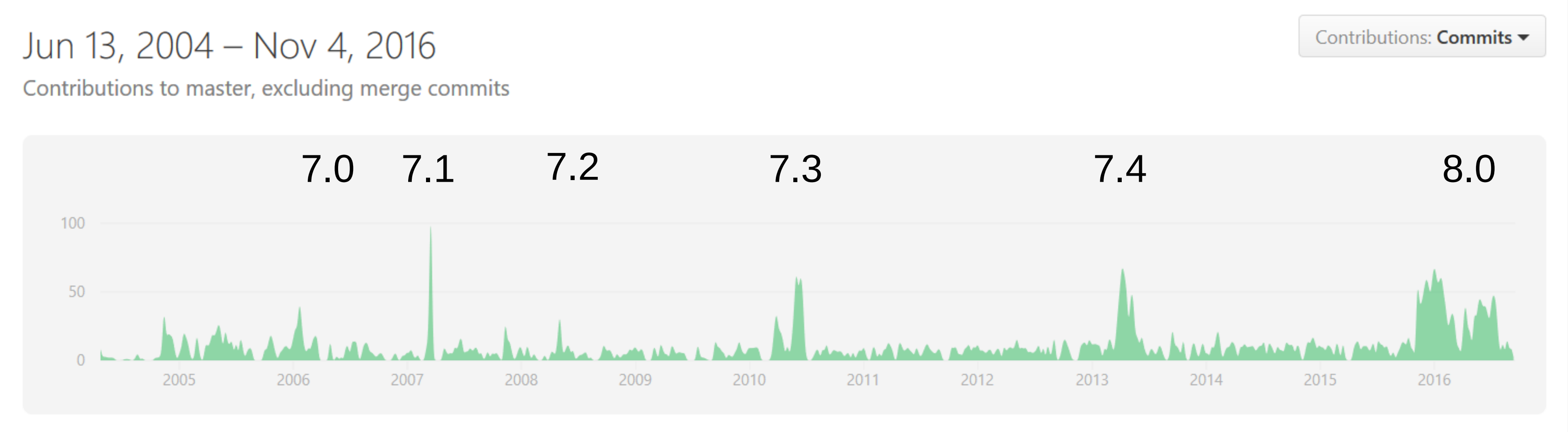
- The GTK+ 3 GUI works just like GTK+ 2.
- Written by Kazunobu Kuriyama (nuko8).

History of Vim

Date	Version	Description
Nov. 2, 1991	1.14	First public release 25 years ago
Dec. 14, 1993	2.0	The first release using the name Vi IMproved.
Aug. 12, 1994	3.0	Support for multiple windows
May 29, 1996	4.0	Graphical user interface
Feb. 19, 1998	5.0	Syntax highlighting, basic scripting
Sep. 26, 2001	6.0	Folding, plugins, multi-language, etc.
May 7, 2006	7.0	Spell checking, code completion, tab pages, current line and column highlighting, undo branches, and more
May 12, 2007	7.1	Bug fixes, new syntax and runtime files, etc.
Aug. 9, 2008	7.2	Floating point support in scripts, new syntax files, etc.
Aug. 15, 2010	7.3	Lua and Python3 support, Blowfish encryption, persistent undo/redo
Aug. 10, 2013	7.4	A new, faster regular expression engine.
Sep. 12, 2016	8.0	Asynchronous I/O support, jobs, lambdas, etc.

From Wikipedia (some dates are different from vim_faq)

Commits graph of Vim



Jun 13, 2004 – Nov 4, 2016

Contributions to master, excluding merge commits

From https://github.com/vim/vim/graphs/contributors

Last patch numbers

Version	Last patch number
7.0	243
7.1	330
7.2	446
7.3	1314
7.4	2367
8.0	On going (60 for now)

Contributors to Vim

- Vim 7.4.001 to 7.4.2367 patch contributor ranking
https://gist.github.com/k-takata/628ff1103f92b9135041a15c43c85b32

No.	Patches	Name (Country)
1	210	Christian Brabandt (Germany)
2	174	Ken Takata (Japan)
3	103	Hirohito Higashi (Japan)
4	92	Yasuhiro Matsumoto (Japan)
5	86	Dominique Pelle (France)
6	45	Yegappan Lakshmanan (?)
7	39	Kazunobu Kuriyama (Japan (maybe))
8	34	Mike Williams (?)
9	32	ZyX (Russia)
10	28	Ozaki Kiichi (Japan)

Contributors to Vim (contd.)

- Nearly 200 people contributed to Vim within 7.4.001 to 7.4.2367 from all over the world including Japan.

Contributions of Japanese people to Vim

- Vim-jp plays a very big role for this.
  - Many patches and bug reports have been submitted to vim_dev from vim-jp members.

What is vim-jp?

- Vim-jp is a community for Japanese vimmers. It aims to collect all resources of Vim related to Japan/Japanese.
- Started on Sep. 15, 2011. (Five years ago.)
- Official Web Site: http://vim-jp.org/
- Issue Tracking System: https://github.com/vim-jp/issues/issues
  - Discuss Vim’s issues, create and review patches for Vim, etc.
- Reading vimrc: http://vim-jp.org/reading-vimrc/
  - Read vimrcs on every Saturday night. (11PM JST)

Achievements of vim-jp and Japanese people

- Before vim-jp (2011), 19 patches from six Japanese people were merged.
- On 2012, 59 patches from nine Japanese people.
  - Data from Taro Muraoka’s slides for ujihisa.vim #3. http://koron.github.io/ujihisa-201211/
- After Vim 7.4, about 560 patches from about 25 Japanese people were merged in three years. (About 186 patches per year) (Note that not all people are member of vim-jp.)
- Number of patches becomes ten times in five years.

Patches from vim-jp members

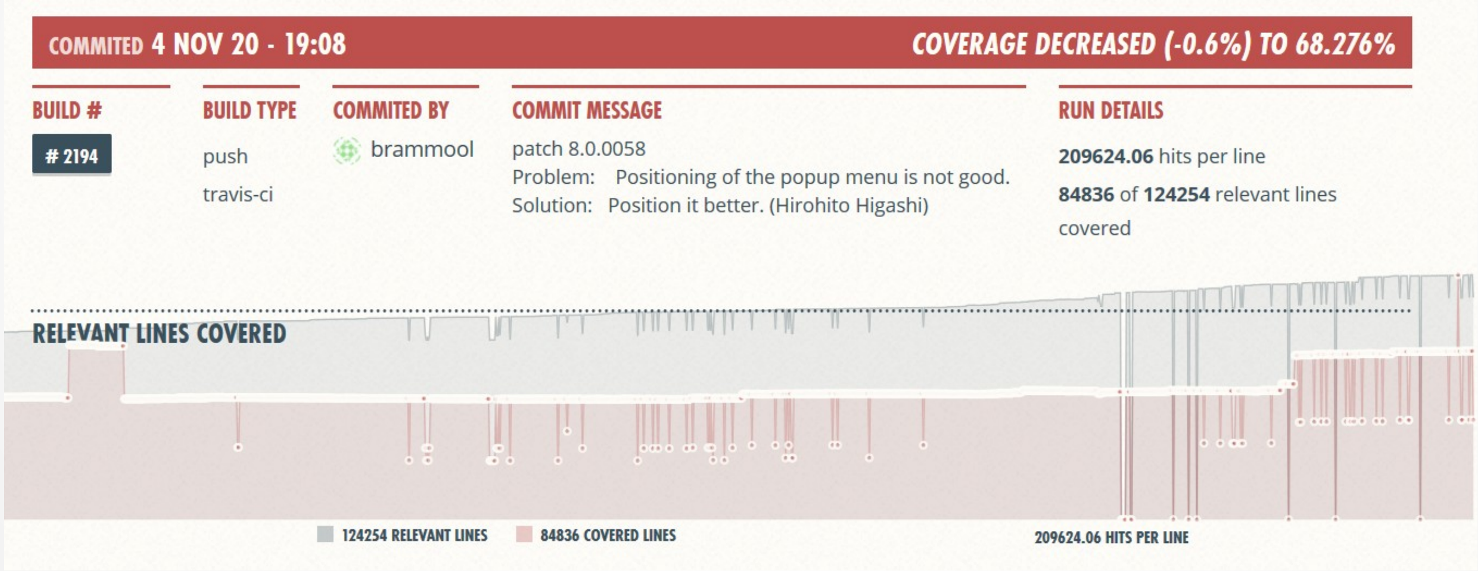
- Channels, jobs and timers support on Windows.
- Lambda and closure.
- 64-bit Number support.
- DirectWrite support.
- Many bug fixes related to Japanese and multibyte.
- And etc.

Moving to GitHub

- Last year (2015), the official repository of Vim moved from Google Code to GitHub. (Because Google Code shut down.)
  - There were two alternatives: GitHub and Bitbucket.
- Vim-jp members think that success of vim-jp might have influence to the decision. See Taro’s slides for VimConf 2015:
  https://drive.google.com/file/d/0ByQIX4Ls1SHlaGFJamN6bnZiUE0/view
- Vim official repository has CI on Travis-CI and AppVeyor. This is also based on vim-jp’s work: https://github.com/vim-jp/vim-ci
- Now Vim has nightly builds for Windows (based on my work): https://github.com/vim/vim-win32-installer

Test Coverage

- One year ago the test coverage of Vim was about 58%.
- Now it becomes about 68%. https://coveralls.io/github/vim/vim



COMMITED 4 NOV 20 - 19:08

COVERAGE DECREASED (-0.6%) TO 68.276%

BUILD #

# 2194

BUILD TYPE

push

travis-ci

COMMITED BY

brammool

COMMIT MESSAGE

patch 8.0.0058
Problem: Positioning of the popup menu is not good.
Solution: Position it better. (Hirohito Higashi)

RUN DETAILS

209624.06 hits per line

84836 of **124254** relevant lines covered

Test Coverage (contd.)

- Vim uses Coveralls for showing coverage data.
- This is also based on vim-jp’s work. So it can be said that vim-jp also contributed indirectly to keep the quality of Vim’s source codes.

- Bram also cares about the coverage.
- Unfortunately sometimes Coveralls cannot show very large source files. Eval.c (the Vim script engine) was very large and it couldn’t show on Coveralls. So he divided eval.c into several files. Now we can see it on Coveralls.

Conclusions

- Vim 8.0 was released on September with many new features mainly in enhancements of Vim script.
- Many people from all over the world contributed to it.
- Japanese people and vim-jp also played a big role for it.

Vim-jp is waiting your contributions!