Principles of Open-Information Communities

Karl Fogel
kfogel@questioncopyright.org
http://subversion.tigris.org
“Producing Open Source Software”

producingoss.com
Ben Collins-Sussman and Brian Fitzpatrick:

“How Open Source Projects Survive Poisonous People (And You Can Too)”

http://video.google.com/videoplay?docid=-4216011961522818645

“What's In It For Me? Benefits from Open Sourcing Code”

http://youtube.com/watch?v=ZtYJoatnHb8
Your Volunteer Community?
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Jim Blandy <jimb@red-bean.com>
Ben Collins-Sussman <sussman@red-bean.com>
Karl Fogel <kfogel@red-bean.com>
Greg Stein <gstein@lyra.org>
Joe Orton <joe@light.plus.com>
Greg Hudson <ghudson@mit.edu>
Brian W. Fitzpatrick <fitz@red-bean.com>
Daniel Stenberg <daniel@haxx.se>
C. Michael Pilato <cmpilato@collab.net>
Kevin Pilch-Bisson <kevin@pilch-bisson.net>
Philip Martin <philip@codematters.co.uk>
Justin Erenkrantz <justin@erenkrantz.com>
Garrett Rooney <rooney@electricjellyfish.net>
Ben Collins <bcollins@debian.org>
Blair Zajac <blair@orcaware.com>
Sander Striker <striker@apache.org>
Nuutti Kotivuori <naked@iki.fi>
David Kimdon <dwhedon@debian.org>
Daniel Rall <drl@collab.net>
Mark Benedetto King <mbk@lowlatency.com>
Eric Gillespie <epg@pretzelnet.org>
Jani Averbach <jaa@iki.fi>
Julian Foad <julianfoad@bopenworld.com>
Tobias Ringström <tobias@ringstrom.mine.nu>
Josh Pieper <jbp@pobox.com>
John Szakmeister <john@szakmeister.net>
Erik Hülsmann <e.huelsmann@gmx.net>
Ben Reser <ben@reser.org>
D.J. Heap <djheap@gmail.com>
Max Bowsher <maxbl@ukf.net>
Peter N. Lundblad <peter@famlundblad.se>
Daniel Berlin <dberlin@dberlin.org>
David Anderson <david.anderson@natulite.net>
Ivan Zhakov <chemodax@gmail.com>
David James <davem@cs.toronto.edu>
Malcolm Rowe <malcolm-svn-dev@farside.org.uk>
Paul Burba <pburba@collab.net>
David Glasser <glasser@davidglasser.net>
Lieven Govaerts <lgo@mobsol.be>
Hyrum Wright <hyrum_wright@mail.utexas.edu>
Vlad Georgescu <vgeorgescu@gmail.com>
Kamesh Jayachandran <kamesh@collab.net>
Mark Phippard <mphippard@collab.net>
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it *easy* to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Step Function = High-Barrier Community
Linear Function = Low-Barrier Community
From: buildbot@mobsol.be
Subject: svn trunk r27236: FAIL (win32-xp VS2005)
To: svn-breakage@subversion.tigris.org
Date: Wed, 17 Oct 2007 07:39:13 +0200
Reply-To: dev@subversion.tigris.org
Reply-To: dev@subversion.tigris.org

Full details are available at:

Author list: blair.niqueco

Build Slave: djh-xp-vse2005

Subversion Buildbot
http://www.mobsol.be/buildbot/

Last 100 lines of the build log (step: Build ):

[Failure instance: Traceback (failure with no frames): twisted.internet.error.ConnectionLost: Connection to the other side was lost in a non-clean fashion.]
Subversion

Build win32-xp_VS2005:#2553

Reason:

SourceStamp:
- Branch: trunk
- Changes: see below

Buildslave: djh-xp-vse2005

Results:
failed failed slave lost

Resubmit Build:
This tree was built from a specific set of source files, and can be rebuilt exactly

Your name: [Redacted]
Reason for re-running build: [Redacted]

Rebuild
Steps and Logfiles:

1. **svn** [update r27236]
   1. **stdio**
2. **Build** ['..\svnbuild.bat %\%(branch)\' failed slave lost]
   1. **stdio**
   2. **interrupt**
3. **Test fsfs+ra_local** []
4. **Publish logs** []
5. **Cleanup** []

Blamelist:

1. blair
2. niqueco

All Changes

1. Changed by: **blair**
   Changed at: **Wed 17 Oct 2007 06:30:40**
   Branch: **trunk**
   Revision: **27232**

   Changed files:
   - **subversion/svn/copy-cmd.c**

   Comments:
2. Changed by: niqueco
   Changed at: Wed 17 Oct 2007 06:28:30
   Branch: trunk
   Revision: 27231

   Changed files:
   - subversion/po/es.po
   Comments:

3. Changed by: blair
   Changed at: Wed 17 Oct 2007 06:38:45
   Branch: trunk
   Revision: 27233

   Changed files:
   - subversion/svn/changelist-cmd.c
   - subversion/svn/diff-cmd.c
   - subversion/svn/info-cmd.c
   - subversion/svn/lock-cmd.c
   - subversion/svn/log-cmd.c
   - subversion/svn/propdel-cmd.c
   - subversion/svn/propgt-cmd.c
   - subversion/svn/proplist-cmd.c
   - subversion/svn/propset-cmd.c
   - subversion/svn/revert-cmd.c
   - subversion/svn/status-cmd.c
   - subversion/svn/unlock-cmd.c
4. Changed by: blair
   Changed at: Wed 17 Oct 2007 06:57:24
   Branch: trunk
   Revision: 27234

   Changed files:
   
   - subversion/include/svn_opt.h

   Comments:

5. Changed by: blair
   Changed at: Wed 17 Oct 2007 07:00:14
   Branch: trunk
   Revision: 27235

   Changed files:
   
   - subversion/include/svn_opt.h

   Comments:

6. Changed by: blair
   Changed at: Wed 17 Oct 2007 07:17:49
   Branch: trunk
   Revision: 27236

   Changed files:
   
   - subversion/include/svn_opt.h
   - subversion/libsvn_subr/opt.c
### BuildBot: Subversion - Iceweasel

#### Subscribe to FAILED builds feed: RSS ATOM

<table>
<thead>
<tr>
<th>Subversion</th>
<th>build successful</th>
<th>build successful</th>
<th>none</th>
<th>build successful</th>
<th>build successful</th>
</tr>
</thead>
<tbody>
<tr>
<td>last build</td>
<td>building ETA in 2647 secs at 00:48:36</td>
<td>building ETA in 3597 secs at 01:04:26</td>
<td>offline 815 pending</td>
<td>offline 31 pending</td>
<td>waiting next at 14:00:00 [50131 secs]</td>
</tr>
</tbody>
</table>

#### Current activity

**Tue 16 Oct 2007**

<table>
<thead>
<tr>
<th>Time</th>
<th>Changes</th>
<th>Command Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:27:31</td>
<td>x86-macosx-gnu_shared win32-xp V32005</td>
<td><code>./svncheck.sh fsfs</code> <code>./svnbuild.sh</code> <code>./svnbuild.bat</code> <code>update r27224</code> <code>./svncheck.bat</code> <code>./svnbuild.bat</code> <code>update r27224</code> <code>Build 3231</code></td>
</tr>
<tr>
<td>22:21:02</td>
<td>cacknin</td>
<td><code>./svncheck.sh fsfs</code> <code>./svnbuild.sh</code> <code>./svnbuild.bat</code> <code>update r27224</code></td>
</tr>
</tbody>
</table>

**22:02:24**

<table>
<thead>
<tr>
<th>Command Lines</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>./svncheck.sh fsfs</code> <code>./svnbuild.sh</code> <code>./svnbuild.bat</code> <code>update r27224</code></td>
</tr>
</tbody>
</table>

### Done
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
DannyB (n=DannyB@66.17.133.19) has joined #svn-dev
ChanServ gives channel operator status to DannyB
Why would svn switch require write access on the branch I'm switching too??
huh?
where are you seeing that?
glasser: no, don't bother, misinterpreted a result of a test I'm writing.
pfew
DannyB- (n=DannyB@56.sub-75-197-117.myvzw.com) has joined #svn-dev
yay
DannyB- has quit (Client Quit)
DannyB has quit (Read error: 104 (Connection reset by peer))
Cmplilato * r27220 /trunk/subversion/
(libsvn_ra_neon/fetch.c libsvn_ra_serf/getlocationsegment.c): Clean up some subpools after we're done with them...
Not being a dav expert, why would I want to use neon over serf or vice versa?
### Mailing list details

**List address**: dev@subversion.tigris.org

**List description**: The development mailing list for Subversion. Non-development questions should be posted to 'users' instead; that includes build problems, configuration issues, etc, as well as usage questions.

**Total messages**: 131269

Messages 1 through 783 of 783 shown | Show paged

<table>
<thead>
<tr>
<th>Author</th>
<th>Subject</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charles Acknin</td>
<td>svn commit: r27222 - branches/svnpatch-diff/subversion/libsvn_client</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn_opt_parse_path and canonical paths</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>David Glasser</td>
<td>svn commit: r27224 - trunk/subversion/tests/cmdline</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>David Glasser</td>
<td>svn commit: r27222 - branches/svnpatch-diff/subversion/libsvn_client</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn_opt_parse_path peg revision inconsistency</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Vlad Georgescu</td>
<td>Authz inconsistency in svn_repos_get_logs()</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Miller, Eric</td>
<td>[Issue 2972] - Error processing command 'set-timestamp'</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Vlad Georgescu</td>
<td>Authz inconsistency in svn_repos_get_logs()</td>
<td>2007-10-16</td>
</tr>
<tr>
<td><a href="mailto:svnlo@mobisobl.be">svnlo@mobisobl.be</a></td>
<td>svn commit: r27190 - in trunk/subversion: libsvn_ra_neon tests/cmdline</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>C. Michael Pilato</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Brian W. Fitzpatrick</td>
<td>svn checkpoint: r27193 - checkpoints</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Erik Huelsmann</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Justin Erenkrantz</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>[RFC] Distinguish Between Paths/File For Summarized Diff Output</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Ben Collins-Sussman</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Kamesh Jayachandran</td>
<td>[PATCH] XFail test to prove issue #2969</td>
<td>2007-10-16</td>
</tr>
<tr>
<td>Kamesh Jayachandran</td>
<td>svn merge question</td>
<td>2007-10-16</td>
</tr>
</tbody>
</table>
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Hacker's Guide to Subversion

If you are contributing code to the Subversion project, please read this first.

$LastChangedDate: 2007-10-09 11:10:32 -0700 (Tue, 09 Oct 2007) $

Table of Contents

- Participating in the community
- Theory and documentation
- Code to read
- Directory layout
- Coding style
- Secure coding guidelines
- Destruction of stacked resources
- Document everything
- Using page breaks
- Error message conventions
- Other conventions
- APR pool usage conventions
- APR status codes
- Exception handling
- Automated tests
- Writing test cases before code
- Debugging the server
- Tracking down memory leaks
- Writing log messages
- Signature
Participating in the community

Although Subversion is originally sponsored and hosted by CollabNet (http://www.collab.net), it's a true open-source project under a BSD-style license. A number of developers work for CollabNet, some work for other large companies (such as RedHat), and many others are simply excellent volunteers who are interested in building a better version control system.

The community exists mainly through mailing lists and a Subversion repository. To participate:

Go to http://subversion.tigris.org/ and

- Join the "dev", "svn", and "announce" mailing lists. The dev list, dev@subversion.tigris.org, is where almost all discussion takes place. All development questions should go there, though you might want to check the list archives first. The "svn" list receives automated commit email.

- Get a copy of the latest development sources from http://svn.collab.net/repos/svn/trunk/.

  New development always takes place on trunk. Bugfixes, enhancements, and new features are backported from there to the various release branches.

There are many ways to join the project, either by writing code, or by testing and/or helping to manage the bug database. If you'd like to contribute, then look at:

- The bugs/issues database http://subversion.tigris.org/project_issues.html

- The bite-sized tasks page http://subversion.tigris.org/project_tasks.html

To submit code, simply send your patches to dev@subversion.tigris.org. No, wait, first read the rest of this file, then start sending patches to dev@subversion.tigris.org. :-)

Done
To help manage the issues database, read over the issue summaries, looking and testing for issues that are either invalid, or are duplicates of other issues. Both kinds are very common, the first because bugs often get unknowingly fixed as side effects of other changes in the code, and the second because people sometimes file an issue without noticing that it has already been reported. If you are not sure about an issue, post a question to dev@subversion.tigris.org. (“Subversion: We're here to help you help us!”)

Another way to help is to set up automated builds and test suite runs of Subversion on some platform, and have the output sent to the svn-breakage@subversion.tigris.org mailing list. See more details at http://subversion.tigris.org/servlets/ProjectMailingListList in the description for svn-breakage list.

Theory and documentation

1. Design
   
   A design spec was written in June 2000, and is a bit out of date. But it still gives a good theoretical introduction to the inner workings of the repository, and to Subversion’s various layers.

2. API Documentation

   The API documentation is generated by Doxygen from the C header files and gives detailed information about all functions, types and so available in the Subversion API.

3. Delta Editors

   Karl Fogel wrote a chapter for O’Reilly’s 2007 book Beautiful Code: Leading Programmers Explain How They Think covering the design and of Subversion’s delta editor interface.

4. Network Protocols

   The WebDAV Usage document is an introduction to Subversion's DAV network protocol, which is an extended dialect of HTTP and uses beginning with "http://" or "https://".

   The SVN Protocol document contains a formal description of Subversion’s SVN network protocol, which is a custom protocol on port 369 (default), whose URLs begin with "svn://" or "svn+ssh://".

5. User Manual

   Version Control with Subversion is a book published by O'Reilly that shows in detail how to effectively use Subversion. The text of the book is free, and is actively being revised. On-line versions are available at http://svnbook.red-bean.com. The XML source and translations to other languages are maintained in their own repository at http://svn.red-bean.com/svnbook.
...skip 43 pages...
Announcement. Include this file in a mail, typically in reply to the announcement of the unofficial tarball.

If you've downloaded and tested a .tar.bz2 file, it is possible to sign a .tar.gz file with the same contents without having to download and test it separately. The trick is to extract the .bz2 file, and pack it using gzip like this:

```
    bzip2 -cd subversion-1.3.0-rc4.tar.bz2  \
         | gzip -9n > subversion-1.3.0-rc4.tar.gz
```

The resulting file should be identical to the file generated by the release manager, and thus can be signed as described above. To verify that the are identical, you may use either the MD5 checksums or the release manager's signature, both of which should be provided with the tarballs.

Localization (i10n)

Translation has been divided into two domains. First, there is the translation of server messages sent to connecting clients. This issue has not yet been worked on. Second there is the translation of the client and its libraries.

The gettext package provides services for translating messages. It uses the xgettext tool to extract strings from the sources for translation. This works by extracting the arguments of the _() and N_() macros. The former is used in context where function calls are allowed (typically anything except static initializers). The latter is used whenever _() isn't. Strings marked with N_() need to be passed to gettext translation routines whenever referenced in code. For an example, look at how the header and footer are handled in subversion/svn/help-cmd.c.

When using direct calls to gettext routines (gettext or dgettext), keep in mind that most of Subversion code is library code. Therefore the default behavior is not necessarily Subversion's own domain. In library code you should use the dgettext versions of the gettext functions. The domain name is defined by the PACKAGE_NAME define.

All required setup for localization is controlled by the ENABLE_NLS conditional in svn_private_config.h (for *nix) and svn_private_config.hw (for Windows). Be sure to put

```
    #include "svn_private_config.h"
```

as the last include in any file which requires localization.

Also note that return values of _() and gettext() calls are UTF-8 encoded; this means that they should be translated to the current locale being written to, or as any form of program output.

Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on
  the heap, instead of just using a pointer to the loop variable
  (which changes each time through the loop).
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

**Found by: Barry Scott <barry@barrys-emacs.org>**
**Patch by: Stefan Sperling <stsp@elego.de>**

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).
Contributors as of r27762

The following list of contributors and their contributions is meant to help us keep track of whom to consider for commit access. The list was generated from "svn log" output by contribulyze.py, which looks for log messages that use the special contribution format.

Please do not use this list as a generic guide to who has contributed what to Subversion! It omits existing full committers, for example, because they are irrelevant to our search for new committers. Also, it merely counts changes, it does not evaluate them. To truly understand what someone has contributed, you have to read their changes in detail. This page can only assist human judgement, not substitute for it.

1. Madan U S <madan@collab.net> (madanus) [51 patches, 19 non-patches] (partial committer)
2. Bhuvaneswaran <bhuvan@collab.net> (bhuvan) [37 patches, 4 non-patches] (partial committer)
3. Joe Swatosh <joe.swatosh@gmail.com> (joeswatosh) [17 patches, 24 non-patches] (partial committer)
4. Senthil Kumaran S <senthil@collab.net> (stylesen) [28 patches, 2 non-patches] (partial committer)
5. Stefan Küng <tortoisesvn@gmail.com> [18 patches, 21 non-patches]
6. Kouhei Sutou <kou@cozmixng.org> (kou) [10 patches, 23 non-patches] (partial committer)
7. Giovanni Bajo <rasky@develer.com> (giovannibajo) [6 patches, 26 non-patches] (partial committer)
8. S. Ramaswamy <srsy70@gmail.com> [16 patches, 3 non-patches]
9. Jelmer Vernooij <jelmer@samba.org> (jrvernooij) [15 patches] (partial committer)
10. Peter Samuelson <peter@p12n.org> [11 patches, 8 non-patches]
133. Norbert Unterberg <nepo@gmx.net> [2 patches, 1 non-patch]
134. Tim Van Holder <tim.van.holder@telenet.be> [2 patches, 1 non-patch]
135. phpxcache [1 patch, 3 non-patches]
136. Ariel Arjona <beerfrick@gmail.com> (beerfrick) [2 patches] (partial committer)
137. Art Haas <ahaas@neosoft.com> [2 patches]
138. Artem Egorkine <arteme@gmail.com> [2 patches]
139. Artem Khodush <greenkaa@mail.ru> [2 patches]
140. Augie Fackler <durin42@gmail.com> [2 patches]
141. Barry Scott <bbarry@barrys-emacs.org> [4 non-patches]
142. Bill Wohler <wohler@newt.com> [1 patch, 2 non-patches]
143. Bruce A. Mah <bmah@packetdesign.com> [2 patches]
144. Craig Peterein <craig@peterein.org> [2 patches]
145. Daniel Patterson <danpat@danpat.net> (danpat) [2 patches] (partial committer)
146. Daniel Westermann-Clark <dwc@tigris.org> [2 patches]
147. David Mahakian <tmoli42@gmx.net> [2 patches]
148. David Waite <mass@akuma.org> (mass) [2 patches] (partial committer)
149. Edmund Horner <ejrh00@gmail.com> [2 patches]
Found  
[r27619, r27593]  
<table>
<thead>
<tr>
<th>Review</th>
<th>Suggested</th>
</tr>
</thead>
<tbody>
<tr>
<td>r7349</td>
<td>r27618</td>
</tr>
</tbody>
</table>

**r27619** | djames | 2007-11-06 11:14:32 -0600 (Tue, 06 Nov 2007)

* autogen.py: Add helpful note when ctypesgen subdirectory is missing.

Found by: Barry Scott <barry@barrys-emacs.org>

**r27618** | djames | 2007-11-06 11:10:05 -0600 (Tue, 06 Nov 2007)

* autogen.py: Use apr-1-config to calculate include paths and link paths for apr-util. Add new '--apu-config' option for specifying path to apru-config script.

Suggested by: Barry Scott <barry@barrys-emacs.org>

**r27593** | hwright | 2007-11-05 09:10:13 -0600 (Mon, 05 Nov 2007)

In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl_copy): Allocate the beg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time)

http://svn.collab.net/viewcvs/svn?rev=275936&view=rev
Revision 27593

Jump to revision: 27593

Author: hwright
Date: Mon Nov 5 15:10:13 2007 UTC (6 days, 13 hours ago)

Log Message:
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl_copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).

Changed paths:

<table>
<thead>
<tr>
<th>Path</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>trunk/subversion/svn/copy-cmd.c</td>
<td>modified, text changed</td>
</tr>
</tbody>
</table>

svnadmin@svn.collab.net

Powered by ViewVC 1.0.4
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl__copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).
45. John Peacock <jpeacock@rowman.com> (jpeacock) [5 patches, 2 non-patches] (partial committer)

46. Luke Call <lsuvkne@onemodel.org> [6 patches]

47. Dan Christian <dchristian@google.com> [5 patches, 1 non-patch]

48. David Summers <david@summersoft.fay.ar.us> (dws) [4 patches, 3 non-patches] (partial committer)

49. Jens Peters <jpeters7677@gmx.de> [4 patches, 3 non-patches]

50. Joshua Varner <jlvarner@gmail.com> (jlvarner) [4 patches, 3 non-patches] (partial committer)

51. Nicolás Lichtmaier <nick@reloco.com.ar> (niqueco) [5 patches, 1 non-patch] (partial committer)

52. Rafael Garcia-Suarez <raphel.garcia-suarez@hexaflux.com> [5 patches, 1 non-patch]

53. Stefan Sperling <stsp@elego.de> [5 patches, 1 non-patch]

54. Bruce Atherton <bruce@callendish.com> [5 patches]

55. Colin Watson <cjwaton@flatline.org.uk> [3 patches, 4 non-patches]

56. G.J. Doornink <dgk@xs4all.nl> [5 patches]

57. Geoff Richards <qef@ungwe.org> [5 patches]

58. Gábor Szabó <szabgab@gmail.com> [5 patches]

59. Jeremy Whitlock <jcscoobyrs@gmail.com> [5 patches]

60. Juanma Barranquero <jmbaranquero@wke.es> [5 patches]

61. Kai Nacke <kai.nacke@redstar.de> (knacke) [5 patches] (partial committer)
Stefan Sperling <stsp@elego.de>

**Patch**

r27593, r27495, r27316, r25707, r25578, r27301

---

**r27593** | hwright | 2007-11-05 09:10:13 -0600 (Mon, 05 Nov 2007)

In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barry@barrys-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl_copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).

---

**r27495** | cmplato | 2007-10-30 14:54:26 -0500 (Tue, 30 Oct 2007)

Teach `svnlook diff` about the `--extensions` parameter (just like the `svn diff` subcommand already accepts).

* subversion/svnlook/main.c
  (options_table): Add `--extensions (-x)` support.
  (cmd_table): Teach the 'diff' subcommand to accept `--extensions (-x)`.
  (svnlook_opt_state): Add 'extensions' member.
  (svnlook_ctxt_t): Add 'diff_options' member.
  (print_diff_tree): Pass requested diff extensions to the diff library (using svn_diff_file_diff_2 now).
  (get_ctxt_baton): Populate context baton's diff_options member.
  (main): Parse and handle new `--extensions (-x)` option.
Revision 27593

Jump to revision: 27593

Author: hwright
Date: Mon Nov 5 15:10:13 2007 UTC (6 days, 13 hours ago)

Log Message:
In the command line client, don't reference a pointer to a local variable outside of its declared scope.

Found by: Barry Scott <barrystack-emacs.org>
Patch by: Stefan Sperling <stsp@elego.de>

* subversion/svn/copy-cmd.c
  (svn_cl_copy): Allocate the peg revision for each copy source on the heap, instead of just using a pointer to the loop variable (which changes each time through the loop).

Changed paths:

<table>
<thead>
<tr>
<th>Path</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>trunk/subversion/svn/copy-cmd.c</td>
<td>modified, text changed</td>
</tr>
</tbody>
</table>

svnadmin@svn.collab.net

Powered by ViewVC 1.0.4
569. Travis Cripps <tcripps@mac.com> [1 non-patch]
570. Troy Straszheim <troy@resophonic.com> [1 non-patch]
571. Ulf Ejlertsson <ulf.ejlertsson@fmail.com> [1 non-patch]
572. Ulf Lamping <ulf.lamping@web.de> [1 non-patch]
573. Vladimir Marek <vlmarek@volny.cz> [1 non-patch]
574. Yves Bergeron <yves.bergeron@shq.gouv.qc.ca> [1 non-patch]
575. Zing <zing+svndev@fastmail.fm> [1 non-patch]
576. danielsh@tigris.org [1 non-patch]
577. faxmodem@free.fr [1 non-patch]
578. kbrannen@gte.net [1 non-patch]
579. loren jan wilson <loren@solar.uchicago.edu> [1 non-patch]
580. mattdoran <mattdoran@bigpond.com> [1 non-patch]
581. olczyk@interaccess.com [1 non-patch]
582. pfleg [1 non-patch]
583. rengmann@lycos.de [1 non-patch]
584. sanity [1 non-patch]
585. stuart updegrave <stuart@voice.byz.org> [1 non-patch]
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
“Producing Open Source Software”
“Producing Open Source Software”

Open Licence → Open Community

producingoss.com
Producing Open Source Software is a book about the human side of open source development. It describes how successful projects operate, the expectations of users and developers, and the culture of free software. It is available in bookstores and from the publisher (O'Reilly Media), or you can browse or download it here.

Producing Open Source Software is released under an open copyright that allows everyone to share and modify the book freely. The latest version is always here. The online version is the same as the commercially available print version — in other words, you can buy a printed copy and know that it's up-to-date. When and if there are significant differences, we will list them here.

- Read online as multiple HTML pages (faster to load)
- Read online as a single HTML page (slower to load)
- Download PDF
- Download all formats at once (XML, HTML, PDF)
- Download RocketBook
Producing Open Source Software is a book about the human side of open source development. It describes how successful projects operate, the expectations of users and developers, and the culture of free software. It is available in bookstores and from the publisher (O'Reilly Media), or you can browse or download it here.

Producing Open Source Software is released under an open copyright that allows everyone to share and modify the book freely. The latest version is always here. The online version is the same as the commercially available print version — in other words, you can buy a printed copy and know that it's up-to-date. When and if there are significant differences, we will list them here.

- Read online as multiple HTML pages (faster to load)
- Read online as a single HTML page (slower to load)
- Read reviews of this book
- Order it from O'Reilly Media
- Visit the book's version control repository
- Download PDF
- Download all formats at once (XML, HTML, PDF)
- Download RocketBook
Translations of *Producing Open Source Software*

People are already at work translating *Producing Open Source Software* into other languages. If you'd like to start or help with a translation, please [let me know](mailto:producingoss.com). We'll give you commit access to the book's repository and put you on a mailing list with other translators. Our goal is to make it as easy as possible for you to translate, and we'll continue to host the translation here after it's done.

- **Deutsch (German)**
  Manuel Barkhau, Sebastian Menge

- **日本語 (Japanese)**
  前木 正弘 (TAKAGI Masahiro), Takaoka Yoshinari

- **Español (Spanish)**
  Rafael Martilotti, Alejandro Ayuso, Francisco Urbano García, José Manuel Puerta Peña, Pedro Andrés Bonilla Polo, Christian López Espinola

- **Français (French)**
  Etienne Savard, Bertrand Florat
Producing Open Source Software

How to Run a Successful Free Software Project

Karl Fogel

Copyright © 2005, 2006, 2007 Karl Fogel, under a Creative Commons Attribution-ShareAlike (3.0) license

Dedication

This book is dedicated to two dear friends without whom it would not have been possible: Karen Underhill and Jim Blandy.

Table of Contents

Preface
  Why Write This Book?
  Who Should Read This Book?
Sources
Acknowledgments
オープンソースソフトウェアのつくりかた

フリーソフトウェアプロジェクトを成功させるコツ

Fogel Karl [FAMILY Given]

（著者）

高木 正弘 [FAMILY Given]

（翻訳者）

Takaoka Yoshinari [FAMILY Given]

（翻訳者）

製作著作 © 2005, 2006, 2007 Karl Fogel, 高木正弘, Yoshinari Takaoka, under a CreativeCommons Attribution-ShareAlike (表示・継承) license (3.0)

謝辞
Produktion von Open Source Software

Wie man ein erfolgreiches freies Software Projekt führt

Karl Fogel

(Author)

Manuel Barkhaus

(Deutsche Übersetzung)

Sebastian Menge

(Deutsche Übersetzung)


Widmung
Producir Software de Código Abierto

Como Llevar a Buen Puerto un Proyecto de Código Libre

Karl Fogel

(Author)

Rafael Martilotti

(Translator)

Alejandro Ayuso

(Translator)

José Manuel Puerta Peña

(Translator)

Pedro Andrés Bonilla Polo

(Translator)
Produire un Logiciel Libre

Comment Diriger avec Succès un Projet de Logiciel Libre

Karl Fogel
(Auteur)
Étienne Savard
(Traducteur)
Bertrand Florat
(Traducteur)

Copyright © 2005, 2006, 2007 Karl Fogel, Étienne Savard, Bertrand Florat, sous une licence CreativeCommons Paternité - Partage des Conditions Initiales à l'Identique (3.0)

Dédicace

Ce livre est dédié à deux amis cherches qui il n’aurait pu être possible: Karen Underhill et Jim.
Optimize Toward Action

- Commit access for anyone
- Shared mailing list
- Visible results quickly
オープンソースソフトウェアのつくりかた

フリーソフトウェアプロジェクトを成功させるコツ

Fogel Karl [FAMILY Given]

(著者)

高木 正弘 [FAMILY Given]

(翻訳者)

Takaoka Yoshinari [FAMILY Given]

(翻訳者)

製作著作 © 2005, 2006, 2007 Karl Fogel, 高木正弘, Yoshinari Takaoka, under a CreativeCommons Attribution-ShareAlike (表示・継承) license (3.0)

謝辞

貢献を持続するためには、変更や更新が必要です。この作品を再使用、改変または拡散することを検討していただければ幸いです。
親友であるKaren UnderhillとJim Blandyに本書をささげます。ふたりがいなければ、本書を完成させることはできなかったでしょう。

目次

序文
この本を書いた理由は？
どんな人たちに読んでほしい？
情報源
謝辞
免責
1. 導入
歴史
独占的なソフトウェアとフリーソフトウェア
意識的な抵抗
無意識な抵抗
「フリーソフト」、「オープンソース」の違い
現状
2. さあ始めましょう
手持ちのもので始めよう
名前の決定
明確な目標を掲げる
フリーであることを宣言する
機能一覧・要件一覧
開発の進捗状況
ダウンロード
バージョン管理システムやバグトラッカーへのアクセス
連絡手段
開発者向けのガイドライン
ドキュメント
ドキュメントの公開方法
パートナー

ドキュメントの公開方法
開発者向けドキュメント
使用例とスクリーンショット
公開場所

ライセンスの選択と適用
「何でもできる」ライセンス
GPL
ライセンスを適用する方法
うまく引っ張っていく
個人的な議論を避ける
炎上を阻止する
きちんととしたコードレビューの習慣
もとどど非公開だったプロジェクトをオープンにするときには、変化の大きさに気をつけよう

広報

3. 技術的な問題
プロジェクトに必要なもの
メールリスト
スパム対策
投稿のフィルタリング
アーカイブでのメールアドレスの処理
識別しやすいヘッダ
Reply-to はどうすべきか
私のふたつの夢
アーカイブ
ソフトウェア
バージョン管理
バージョン管理に関する用語集
バージョン管理システムの選択
バージョン管理システムの使用法
すべてをバージョン管理する
すべてのバージョン管理する
ウェブで閲覧できるようにする
コミットメール
プルリクの活用
情報の一元管理
承認
バグ追跡システム
議論の場としてメーリングリストを使う
バグ追跡システムをあらかじめフィルタする
IRC/リアルタイムに行なわれるチャットシステム
ポット
IRCの会話を保存する
Wiki
ウェブサイト
ツールが一致したホスティングサイト
ホスティングサイトを選ぶ
匿名性とプロジェクト参加
4. プロジェクトの政治構造と社会構造
優しい独裁者
誰がよく「優しい独裁者」になれるか？
合意に基づく民主主義
バージョン管理を行なうと堅くならずに対応
合意に至らなければ投票する
いつ投票を行うべきか？
誰が投票するのか？
世論調査と投票
拒否権
全てを記録しておく
5. Money
Types of Involvement
Hire for the Long Term
Hire for the Long Term
Appear as Many, Not as One
Be Open About Your Motivations
Money Can't Buy You Love
Contracting
  Review and Acceptance of Changes
    Case study: the CVS password-authentication protocol
Funding Non-Programming Activities
  Quality Assurance (i.e., Professional Testing)
  Legal Advice and Protection
  Documentation and Usability
  Providing Hosting/Bandwidth
Marketing
  Remember That You Are Being Watched
  Don't Bash Competing Open Source Products

6. コミュニケーション
書いたことがすべて
構成や体裁
中身
口調
何が失礼にあたるのか
顔
陥りがちな罠
目的のない投稿をしない
生産的なスレッドとそうでないスレッド
簡単な議題ほど長引く
宗教論争を回避する
"口やかましい少数派"について
扱いにくい人たち
扱いにくい人たちへの対応
実例
高木化への対応
表列
巨大化への対応
アーカイブを目に付きやすくする方法
全リソースをアーカイブと同様に扱う
しきたりの成文化
バグ追跡システムでは議論しない
宣伝・広報
セキュリティ脆弱性の告知
バグ報告を受ける
大至急それを修正する
CAN/CVE 番号
事前通知
修正を一般に公開する
7. パッケージの作成、リリース、日々の開発
リリースに番号を付ける
リリース番号の構成要素
単純なやり方
奇数/偶数 に意味を持たせるやり方
リリースプランチ
リリースプランチの使い方
リリースを安定させるプロセス
リリースオーナーによる独裁
リリースに含める変更を投票で決める
リリースを安定させるプロセスを管理する
リリースマネージャー
パッケージング
パッケージのフォーマット
Name and Layout
To capitalize or not to capitalize
Pre-releases
Compilation and Installation
Binary Packages

Done
Compilation and Installation
Binary Packages
Testing and Releasing
Candidate Releases
Announcing Releases
Maintaining Multiple Release Lines
Security Releases
Releases and Daily Development
Planning Releases

8. Managing Volunteers
Getting the Most Out of Volunteers
Delegation
  Distinguish clearly between inquiry and assignment
  Follow up after you delegate
  Notice what people are interested in
Praise and Criticism
Prevent Territoriality
The Automation Ratio
  Automated testing
Treat Every User as a Potential Volunteer
Share Management Tasks as Well as Technical Tasks
  Patch Manager
  Translation Manager
  Documentation Manager
  Issue Manager
  FAQ Manager

Transitions
Committers
  Choosing Committers
  Revoking Commit Access
  Partial Commit Access
  Dormant Committers
Dormant Committers
Avoid Mystery
Credit
Forks
Handling a Fork
Initiating a Fork

9. Licenses, Copyrights, and Patents
Terminology
Aspects of Licenses
The GPL and License Compatibility
Choosing a License
   The MIT / X Window System License
   The GNU General Public License
   Is the GPL free or not free?
What About The BSD License?
Copyright Assignment and Ownership
   Doing Nothing
   Contributor License Agreements
   Transfer of Copyright
Dual Licensing Schemes
Patents
Further Resources
A. Free Version Control Systems
B. Free Bug Trackers
C. Why Should I Care What Color the Bikeshed Is?
D. Example Instructions for Reporting Bugs
E. Copyright
Principles of Participation

- Participants are people, not companies
- Make it *rewarding* to do things right
- Make it easy to do things right
- Make decisions in public
- Document the culture
- Don't work – amplify other people's work
- No obstacles: act now, audit later
Recovery is better than protection.
Principles of Open-Information Communities

Karl Fogel
kfogel@questioncopyright.org