How To Open Source Your In-House App

Karl Fogel

kfogel@civiccommons.org
How Open Source Projects Work

Karl Fogel

kfogel@civiccommons.org
producingoss.com

"Producing Open Source Software"
What does “open source” make you think?
• Open source software is commercial software
• Open source software is commercial software
• It's just not proprietary
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• No such thing as gov't-only OSS
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• No such thing as gov't-only OSS
• “Open source” same as “free software”
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• No such thing as gov't-only OSS
• “Open source” same as “free software”
• Copyleft vs non-copyleft (both qualify)
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• No such thing as gov't-only OSS
• “Open source” same as “free software”
• Copyleft vs non-copyleft (both qualify)
• Lots of religious wars in open source (e.g., AGPL)
• Open source software is commercial software
• It's just not proprietary
• No such thing as academic-only OSS
• No such thing as gov't-only OSS
• “Open source” same as “free software”
• Copyleft vs non-copyleft (both qualify)
• Lots of religious wars in open source (e.g., AGPL)
• But not about the definition.
• You open source your code, not your time.
What we've got.

- NWS weather feed parser (Ron)
Group breakouts.

- demo
- procure
- install
- use
- ask a question
- report a bug :-)
- contribute a patch
Principles of Participation

- Participants are people, not organizations
- 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 organizations
- 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 organizations
- 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
Investment vs. Reward

CIVIC COMMONS
SHARING TECHNOLOGY FOR THE PUBLIC GOOD
Step Function = High-Barrier Community
Linear Function = Low-Barrier Community
Principles of Participation

- Participants are people, not organizations
- 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
<table>
<thead>
<tr>
<th>Time</th>
<th>User</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:12</td>
<td>DannyB (n=DannyB@66.17.133.19)</td>
<td>has joined #svn-dev</td>
</tr>
<tr>
<td>10:12</td>
<td>ChanServ</td>
<td>gives channel operator status to DannyB</td>
</tr>
<tr>
<td>10:13</td>
<td>&lt;lgo&gt;</td>
<td>Why would svn switch require write access on the branch I'm switching too??</td>
</tr>
<tr>
<td>10:15</td>
<td>&lt;glasser&gt;</td>
<td>huh?</td>
</tr>
<tr>
<td>10:16</td>
<td>&lt;glasser&gt;</td>
<td>where are you seeing that?</td>
</tr>
<tr>
<td>10:16</td>
<td>&lt;lgo&gt;</td>
<td>glasser: no, don't bother, misinterpreted a result of a test I'm writing.</td>
</tr>
<tr>
<td>10:16</td>
<td>&lt;lgo&gt;</td>
<td>pfew</td>
</tr>
<tr>
<td>10:16</td>
<td>DannyB_ (n=<a href="mailto:DannyB@56.sub-75-197-117.myvzw.com">DannyB@56.sub-75-197-117.myvzw.com</a>)</td>
<td>has joined #svn-dev</td>
</tr>
<tr>
<td>10:17</td>
<td>&lt;glasser&gt;</td>
<td>yay</td>
</tr>
<tr>
<td>10:17</td>
<td></td>
<td>-- DannyB_ has quit (Client Quit)</td>
</tr>
<tr>
<td>10:19</td>
<td></td>
<td>-- DannyB has quit (Read error: 104 (Connection reset by peer))</td>
</tr>
<tr>
<td>10:19</td>
<td>&lt;CIA-7&gt;</td>
<td>cmpilato * r27220 /trunk/subversion/ (libsvn_ra_neon/fetch.c libsvn_ra_serf/getlocationsegments.c): Clean up some subpools after we're done with them. ...</td>
</tr>
<tr>
<td>10:20</td>
<td>&lt;jeremyw&gt;</td>
<td>Not being a dav expert, why would I want to use neon over serf or vice versa?</td>
</tr>
</tbody>
</table>
### 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>
<th>Action</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>
<td>Delete</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn_opt_parse_path and canonical paths</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>David Glasser</td>
<td>svn commit: r27224 - trunk/subversion/tests/cmdline</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>David Glasser</td>
<td>svn commit: r27222 - branches/svnpatch-diff/subversion/libsvn_client</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn_opt_parse_path peg revision inconsistency</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Vlad Georgescu</td>
<td>Authz inconsistency in svn_repos_get_logs()</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Miller, Eric</td>
<td>[Issue 2972] - Error processing command 'set-timestamp'</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Vlad Georgescu</td>
<td>Authz inconsistency in svn_repos_get_logs()</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td><a href="mailto:svnlg0@mobsol.be">svnlg0@mobsol.be</a></td>
<td>svn commit: r27190 - in trunk/subversion: libsvn_ra_neon tests/cmdline</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>C. Michael Pilato</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Brian W. Fitzpatrick</td>
<td>svn checkpoint: r27139 - checkpoints</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Erik Huelsmann</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Justin Erenkrantz</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Karl Fogel</td>
<td>[RFC] Distinguish Between Paths/File For Summarized Diff Output</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Blair Zajac</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Ben Collins-Sussman</td>
<td>svn checkpoint: r27213 - checkpoints/relative-externals</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Kamesh Jayachandran</td>
<td>[PATCH] XFail test to prove issue #2969</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
<tr>
<td>Kamesh Jayachandran</td>
<td>svn merge question</td>
<td>2007-10-16</td>
<td>Delete</td>
</tr>
</tbody>
</table>
Principles of Participation

- Participants are people, not organizations
- 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
- Others
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 notifications.
- 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. :-)

Writing log messages
- Credit
- Patch submission guidelines
- Filing bugs / issues
- Issue triage
- Commit access
- The configuration/build system under unix
- How to release a distribution tarball
- Release numbering, compatibility, and deprecation
- Stabilizing and maintaining releases
- Signing source distribution packages
- Localization (i18n)
dev@subversion.tigris.org. :-)

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 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 ra_svn network protocol, which is a custom protocol on port 3699 (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 \n | 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 been p

for now. 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 work 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 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 used as any form of program output.

Principles of Participation

- Participants are people, not organizations
- 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).
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]
<table>
<thead>
<tr>
<th></th>
<th>Name</th>
<th>Email/Username</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>133</td>
<td>Norbert Unterberg</td>
<td><a href="mailto:nepo@gmx.net">nepo@gmx.net</a></td>
<td>2 patches, 1 non-patch</td>
</tr>
<tr>
<td>134</td>
<td>Tim Van Holder</td>
<td><a href="mailto:tim.van.holder@telenet.be">tim.van.holder@telenet.be</a></td>
<td>2 patches, 1 non-patch</td>
</tr>
<tr>
<td>135</td>
<td>phpxcache</td>
<td></td>
<td>1 patch, 3 non-patches</td>
</tr>
<tr>
<td>136</td>
<td>Ariel Arjona</td>
<td><a href="mailto:beerfrick@gmail.com">beerfrick@gmail.com</a> (beerfrick)</td>
<td>2 patches (partial committer)</td>
</tr>
<tr>
<td>137</td>
<td>Art Haas</td>
<td><a href="mailto:ahaas@neosoft.com">ahaas@neosoft.com</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>138</td>
<td>Artem Egorkine</td>
<td><a href="mailto:arteme@gmail.com">arteme@gmail.com</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>139</td>
<td>Artem Khodush</td>
<td><a href="mailto:greenkaa@mail.ru">greenkaa@mail.ru</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>140</td>
<td>Augie Fackler</td>
<td><a href="mailto:durin42@gmail.com">durin42@gmail.com</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>141</td>
<td>Barry Scott</td>
<td><a href="mailto:bary@barys-emacs.org">bary@barys-emacs.org</a></td>
<td>4 non-patches</td>
</tr>
<tr>
<td>142</td>
<td>Bill Wohler</td>
<td><a href="mailto:wohler@newt.com">wohler@newt.com</a></td>
<td>1 patch, 2 non-patches</td>
</tr>
<tr>
<td>143</td>
<td>Bruce A. Mah</td>
<td><a href="mailto:bmah@packetdesign.com">bmah@packetdesign.com</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>144</td>
<td>Craig Peterein</td>
<td><a href="mailto:craig@peterein.org">craig@peterein.org</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>145</td>
<td>Daniel Patterson</td>
<td><a href="mailto:danpat@danpat.net">danpat@danpat.net</a> (danpat)</td>
<td>2 patches (partial committer)</td>
</tr>
<tr>
<td>146</td>
<td>Daniel Westermann-Clark</td>
<td><a href="mailto:dwc@tigris.org">dwc@tigris.org</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>147</td>
<td>David Mahakian</td>
<td><a href="mailto:tmoli42@gmx.net">tmoli42@gmx.net</a></td>
<td>2 patches</td>
</tr>
<tr>
<td>148</td>
<td>David Waite</td>
<td><a href="mailto:mass@akuma.org">mass@akuma.org</a> (mass)</td>
<td>2 patches (partial committer)</td>
</tr>
<tr>
<td>149</td>
<td>Edmund Horner</td>
<td><a href="mailto:ejrh00@gmail.com">ejrh00@gmail.com</a></td>
<td>2 patches</td>
</tr>
</tbody>
</table>
Barry Scott <barry@barrys-emacs.org>

<table>
<thead>
<tr>
<th>Found</th>
<th>Review</th>
<th>Suggested</th>
</tr>
</thead>
<tbody>
<tr>
<td>r27619, r27593, r7349</td>
<td></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 peg 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=27593&view=rev
Revision 27593

Jump to revision:

27593

Go

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:

Path                                      Details
-----------------------------------------   
trunk/subversion/svn/copy-cmd.c           modified, text changed

svnadmin@svn.collab.net
Powered by ViewVC 1.0.4
Principles of Participation

- Participants are people, not organizations
- 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
Optimize Toward Action

- Commit access for *anyone*
- Shared mailing list
- Visible results quickly
“Audit trail” means “No regrets”.
Recovery is better than protection.
Principles of Participation

- Participants are people, not organizations
- 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
Your Volunteer Community?
References

• wiki.civiccommons.org/Releasing_Open_Source
• wiki.civiccommons.org/Open_Source_Development_Guidelines
• wiki.civiccommons.org/Choosing_a_License
• wiki.civiccommons.org/Legal_Policy (big)
• civiccommons.org/2011/05/it-dashboard-six-weeks-in
• civiccommons.org/2011/01/be-open-from-day-one
• See the rest of CivicCommons.org too!
• cio-nii.defense.gov/sites/oss/index.shtml
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
PANTALONS SUPERIEUR

LES JEANS LEVI'S SONT DISPONIBLES ICI