

## Ruby trunk - Feature #14551

### What's missing to switch to Git instead of using Subversion?

02/25/2018 02:39 PM - Eregon (Benoit Daloze)

<b>Status:</b>	Rejected	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	hsbt (Hiroshi SHIBATA)	
<b>Target version:</b>		
<b>Description</b>		
<p>I know migrating to Git get asked at least every year at RubyKaigi, and this should be a motivation to change. <a href="#">#14470</a> is one example making me wonder why we still use such an ancient technology like Subversion.</p> <p>To be clear, this does NOT mean changing the issue tracker, that is orthogonal.</p> <p>1) I have heard some scripts might stop working if we switch to Git and nobody knows a list of those scripts. But, can't we just switch and adapt them as needed? As long as this is not close to a release, I expect there will be a couple days where scripts are fixed but after that pretty much everything should work just as well as before.</p> <p>2) Are there Ruby committers still using Subversion? I would guess the vast majority is using Git. Could they use <a href="https://help.github.com/articles/support-for-subversion-clients/">https://help.github.com/articles/support-for-subversion-clients/</a> maybe if they want to keep using Subversion?</p> <p>3) Does Redmine lack Git support? Is there an estimation of how much work it would take to add it?</p> <p>4) Something else?</p> <p>Advantages include:</p> <ul style="list-style-type: none"><li>• External people can contribute more directly with Pull Requests on GitHub, if they want (which is what most external contributors do). There is no need to squash their commits or manually checkout their branch, which delays integration significantly and loses the original author in the process (only mentioned as text while they really deserve to be marked as having contributed to Ruby).</li><li>• We use a non-obsolete technology and don't look like a language from the past.</li><li>• The setup for contributing to Ruby becomes much simpler (git clone, not a mix of SVN/git all the time)</li><li>• Many others advantages but listing them is not the point of this issue.</li></ul>		
<b>Related issues:</b>		
Related to Ruby trunk - Feature #11741: Migrate Ruby to Git from Subversion		<b>Rejected</b>
Related to Ruby trunk - Misc #10547: How to move the ruby project to git		<b>Rejected</b>

### History

**#1 - 02/25/2018 02:40 PM - Eregon (Benoit Daloze)**

- Related to Feature #11741: Migrate Ruby to Git from Subversion added

**#2 - 02/25/2018 02:40 PM - Eregon (Benoit Daloze)**

- Related to Misc #10547: How to move the ruby project to git added

**#3 - 02/25/2018 04:56 PM - shevegen (Robert A. Heiler)**

For point #1, perhaps the issue here could collect all files that would need modification (or a new file that handles git; though ideally, the existing files could be adapted to use both svn and git, so that if it is ever decided to change, the infrastructure would already work).

**#4 - 02/26/2018 12:37 AM - hsbt (Hiroshi SHIBATA)**

- Assignee set to hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

I have a plan to migrate git from subversion. But I have no enough time to figure out it. Because I need to update the infrastructure of distribution and mailing-list in this year.

**#5 - 02/26/2018 08:30 AM - naruse (Yui NARUSE)**

- Status changed from Assigned to Rejected

Duplicated with [#10547](#)

1) I have heard some scripts might stop working if we switch to Git and nobody knows a list of those scripts.

You can find some of those scripts in tool/ of ruby's repo.

2) Are there Ruby committers still using Subversion? I would guess the vast majority is using Git. Could they use <https://help.github.com/articles/support-for-subversion-clients/> maybe if they want to keep using Subversion?

Advantages include:

- External people can contribute more directly with Pull Requests on GitHub, if they want (which is what most external contributors do).

You need to read previous discussion in detail.

[#10547](#) has a point.

**#6 - 02/26/2018 12:28 PM - Eregon (Benoit Daloze)**

naruse (Yui NARUSE) wrote:

You need to read previous discussion in detail.  
[#10547](#) has a point.

What's the point?

That we need a planner showing the merits and listing issues?

The title of [#10547](#) sounds like that issue should be about listing issues to move to Git but as far as I see there is no list of issues there. This is why I filed this issue.

I think it would be valuable to answer my 4 questions above about what are the problems to migrate to Git. Some people might be willing to help once it's clear what needs to be done.

**#7 - 02/26/2018 01:39 PM - dsferreira (Daniel Ferreira)**

Eregon (Benoit Daloze) wrote:

Some people might be willing to help once it's clear what needs to be done.

If it is an extra pair of hands you are looking for I can volunteer.

A clean git development environment is something I'm looking forward in ruby for a long time. So count on me to help.

**#8 - 02/28/2018 05:43 AM - duerst (Martin Dürst)**

Two points that would be worse for git, the first is general, the second is personal:

- 1) Revisions become meaningless hashes; it's totally impossible to look at a revision id and know where it fits in relation to other revision ids.
- 2) Forking, making a change, and sending it back as a pull request is easy. But staying in sync over the long term is not. As an example, <https://github.com/duerst/ruby> says "This branch is 8734 commits behind ruby:trunk". I have asked people from github (incl. Aaron Paterson) to add a "sync me up" button besides this message, but that hasn't happened yet, even if to me this seems to be a non-brainer. And if I try to fork again, it tells me that I already have a fork.

**#9 - 02/28/2018 02:08 PM - nobu (Nobuyoshi Nakada)**

duerst (Martin Dürst) wrote:

- 1) Revisions become meaningless hashes; it's totally impossible to look at a revision id and know where it fits in relation to other revision ids.

The revision shown by ruby -v can be managed by adding a tag REV\_xxxxx, where xxxxx is the svn revision of the tagged commit. See VCS::GIT.get\_revisions in tool/vcs.rb.

- 2) Forking, making a change, and sending it back as a pull request is easy. But staying in sync over the long term is not. As an example, <https://github.com/duerst/ruby> says "This branch is 8734 commits behind ruby:trunk". I have asked people from github (incl. Aaron Paterson) to add a "sync me up" button besides this message, but that hasn't happened yet, even if to me this seems to be a non-brainer. And if I try to fork again, it tells me that I already have a fork.

I experienced it today too.

But it is an issue of github, not git itself.

Note that git doesn't equal github.

I think probably few committer are against moving to git, but some are against moving to github because of their policy.

#### #10 - 02/28/2018 06:22 PM - normalperson (Eric Wong)

[duerst@it.aoyama.ac.jp](mailto:duerst@it.aoyama.ac.jp) wrote:

Two points that would be worse for git, the first is general, the second is personal:

1) Revisions become meaningless hashes; it's totally impossible to look at a revision id and know where it fits in relation to other revision ids.

"git describe" can help give you a rough age iff we tag off trunk (or merge the release branches back to trunk); but it currently doesn't work for our repository.

But being a decentralized system, git history is non-linear and reliance on serial numbers can be misleading. Decentralization is an alien concept to a lot of people, so they rely on centralized messaging systems (e.g. GitHub, Redmine instances) to make sense of it :<

In contrast, git and Linux kernel development messaging is peer-to-peer: developers all email each other directly. The mailing lists are merely an agreed-upon logging/broadcast channel.

2) Forking, making a change, and sending it back as a pull request is easy. But staying in sync over the long term is not. As an example, <https://github.com/duerst/ruby> says "This branch is 8734 commits behind ruby:trunk". I have asked people from github (incl. Aaron Paterson) to add a "sync me up" button besides this message, but that hasn't happened yet, even if to me this seems to be a non-brainer. And if I try to fork again, it tells me that I already have a fork.

You can use "git rebase" (or "git merge(\*)") locally for that and push to a new branch. Doing it via a web interface falls down once you hit conflicts and need to fix+test locally. There's no need to rely on proprietary tools at all.

(\*) I do caution against criss-cross merging because it makes history confusing to follow; rebase is often nicer because it allows you to clean up the history for presentation. Generally, merges should be done by the designated upstream and resolved there, not by downstream.

#### #11 - 03/01/2018 09:40 AM - Eregon (Benoit Daloze)

1) Revisions become meaningless hashes; it's totally impossible to look at a revision id and know where it fits in relation to other revision ids.

Looking at the committer date is enough to know the relation if we rebase (which is similar to what is done with subversion).

2) Forking, making a change, and sending it back as a pull request is easy. But staying in sync over the long term is not. As an example, <https://github.com/duerst/ruby> says "This branch is 8734 commits behind ruby:trunk". I have asked people from github (incl. Aaron Paterson) to add a "sync me up" button besides this message, but that hasn't happened yet, even if to me this seems to be a non-brainer. And if I try to fork again, it tells me that I already have a fork.

For committers, there is no need to use GitHub. So committers would just push to the official ruby git repository, just like with SVN.

For pull requests on GitHub for non-committers which want to use GitHub, it shouldn't matter whether their fork is up to date or not. In a normal setup, there is the official ruby repository as the upstream/origin remote, and their fork as an extra remote. Then new branches should be pushed on their fork for pull requests, but should be rebased on top of the official ruby repository's master branch, not the fork's.