Ruby - Misc #16112

Reduce the possibility of "expand tabs" commit occurrences

08/19/2019 08:09 AM - k0kubun (Takashi Kokubun)

Status:	Closed	
Priority:	Normal	
Assignee:		

Description

Problem

- While we agreed to use only spaces for indentation of C code in [Bug #14246], we sometimes hit "expand tabs" commit when we just edit a part of existing lines with hard tab indentation.
- "expand tabs" commit bothers people when we need to perform a revert or a backport.
- However, because [Bug #14246] aimed to eventually make indentations consistent, we do not want to just drop "expand tabs".
  - One of the motivations to solve [Bug #14246] is that having hard tabs makes preprocessed MJIT header ugly and it makes debugging on GCC hard. As MJIT may introduce C-code inlining for sources outside vm.c in the future, we want to fix the issue in almost all C sources which can be run on runtime.

Possible Solutions

There would be some options to approach the problem. I'd like to hear opinions about these options.

1. Expand all tabs at once for all files managed by auto-style.

- In [Bug #14246], this was clearly objected for the reason "Indents should become consistent over time".
- In my understanding, not folliwng "Indents should become consistent over time" would be problematic mainly for polluting "git blame" and conflicts on backport.
  - For the first point, we can use -w option of git blame to ignore that.
  - For backport, it's a trade-off with many "expand tabs" commits. We need opinions about this from @usa (Usaku NAKAMURA) and @nagachika (Tomoyuki Chikanaga) .
- Also reverting a commit before the commit expanding all tabs would be bothering, for a short while.

1. Skip expand tabs for existing lines when indentation is not changed, and expand tabs only in newly-added lines.

- If editors are configured properly, "expand tabs" would not happen for new patches in this approach.
- Even in this approach, at least we will not go to the opposite direction of eventually achieving [Bug #14246]. So it seems acceptable.

1. Prepare a local pre-commit hook to expand tabs in newly-added or edited lines, and let people who do not want an "expand tabs" commit use it.
2. Make pull request CI fail when there's diff to be expanded, and let people who do not want "expand tabs" commit things from pull requests.

Related issues:

Related to Ruby - Bug #14246: Inconsistent C source code indentation	**Closed**

Associated revisions

Revision 15eaedf805fb2727c79a6c59af6d5f6c2a6d634b - 08/22/2019 12:24 PM - k0kubun (Takashi Kokubun)

Add misc/expand_tabs.rb ported from auto-style.rb

This is implemented to close [Misc #16112] because all other options got
at least one objection, and nobody has objected to this solution.

This code is a little complicated for the purpose, but that's just
because it includes some historical code for auto-style.rb:
https://github.com/ruby/ruby-commit-hook/blob/918a7c31b69ad2f7b125608c1c6a1f4fd01ec15a/bin/auto-style.rb

Please feel free to improve this file as you like.

[Misc #16112]

Revision 15eaedf805fb2727c79a6c59af6d5f6c2a6d634b - 08/22/2019 12:24 PM - k0kubun (Takashi Kokubun)

Add misc/expand_tabs.rb ported from auto-style.rb

This is implemented to close [Misc #16112] because all other options got
at least one objection, and nobody has objected to this solution.

This code is a little complicated for the purpose, but that's just
because it includes some historical code for auto-style.rb:
https://github.com/ruby/ruby-commit-hook/blob/918a7c31b69ad2f7b125608c1c6a1f4fd01ec15a/bin/auto-style.rb

Please feel free to improve this file as you like.

[Misc #16112]

Revision 15eaedf8 - 08/22/2019 12:24 PM - k0kubun (Takashi Kokubun)

Add misc/expand_tabs.rb ported from auto-style.rb

This is implemented to close [Misc #16112] because all other options got
at least one objection, and nobody has objected to this solution.

This code is a little complicated for the purpose, but that's just
because it includes some historical code for auto-style.rb:
https://github.com/ruby/ruby-commit-hook/blob/918a7c31b69ad2f7b125608c1c6a1f4fd01ec15a/bin/auto-style.rb

Please feel free to improve this file as you like.

[Misc #16112]

Revision 57119dd561418c917b885db5f5af7f129a96d1ec - 03/22/2020 07:37 AM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h

I can't live without this when using gdb or perf report.

See also: [Misc #16112]

Revision 57119dd561418c917b885db5f5af7f129a96d1ec - 03/22/2020 07:37 AM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h

I can't live without this when using gdb or perf report.

See also: [Misc #16112]

Revision 57119dd5 - 03/22/2020 07:37 AM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h

I can't live without this when using gdb or perf report.

See also: [Misc #16112]

Revision 9ebf74fd7843c34eda59f228fc34ab73f2cc458d - 03/22/2020 06:50 PM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h (#2975)

This is necessary to avoid converting a hard tab to just 1 space in
preprocessor to generate rb_mjit_header.h, which is helpful when using
gdb or perf report.

See also: [Misc #16112]

This reverts commit 91acdd17c4b4bb69a8fa3ada46e09dad46b9362e.
Fixed permission failure on Travis, encoding, and added SKIPPED_FILES.

Revision 9ebf74fd7843c34eda59f228fc34ab73f2cc458d - 03/22/2020 06:50 PM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h (#2975)

This is necessary to avoid converting a hard tab to just 1 space in
preprocessor to generate rb_mjit_header.h, which is helpful when using
gdb or perf report.

See also: [Misc #16112]

This reverts commit 91acdd17c4b4bb69a8fa3ada46e09dad46b9362e.
Fixed permission failure on Travis, encoding, and added SKIPPED_FILES.

Revision 9ebf74fd - 03/22/2020 06:50 PM - k0kubun (Takashi Kokubun)

Expand tabs for rb_mjit_header.h (#2975)

This is necessary to avoid converting a hard tab to just 1 space in
preprocessor to generate rb_mjit_header.h, which is helpful when using
gdb or perf report.

See also: [Misc #16112]

This reverts commit 91acdd17c4b4bb69a8fa3ada46e09dad46b9362e.
Fixed permission failure on Travis, encoding, and added SKIPPED_FILES.

History

#1 - 08/19/2019 08:38 AM - k0kubun (Takashi Kokubun)

- Description updated

#2 - 08/19/2019 08:42 AM - mame (Yusuke Endoh)

My (weak) vote goes to 1, or 0 (just drop "expand tabs" :-).

#3 - 08/19/2019 09:12 AM - shyouhei (Shyouhei Urabe)

Option 4 is not a lethal tab killer, but nevertheless a good thing to have.

#4 - 08/19/2019 09:39 AM - k0kubun (Takashi Kokubun)

- Description updated

#5 - 08/19/2019 06:46 PM - alanwu (Alan Wu)

- File Screen Shot 2019-08-19 at 2.43.06 PM.png added

I would love it if (1) can happen. Editors like Sublime Text, VSCode and Atom cannot distinguish between indent size and tab size, so they have trouble handling the current style. I use Sublime and have to set my indentation size to 8 to get it to render tabs correctly. Consequently, I have to indent manually by pressing space four times. There are plugins that somewhat improve the situation, but I wasn't able to find anything that provides a first class experience.

Imagine greeting someone opening Ruby's source for the first time with a sub-par editing experience. It's certainly not very welcoming.

Mostly a minor annoyance, to be sure, and perhaps C programmers don't use these editors. However, it's something worth considering.

#6 - 08/20/2019 08:26 AM - byroot (Jean Boussier)

I second what Alan said, editing Ruby source with my editor is a nightmare.

On another note, the main objection for getting rid of tabs once and for all seem to be git blame, so I wonder wether git filter-branch was considered.

It would allow to fix the formatting while keeping a clean history, the downside being that all commit shas would change.

#7 - 08/21/2019 01:12 PM - k0kubun (Takashi Kokubun)

> I would love it if (1) can happen.
>
> Mostly a minor annoyance, to be sure, and perhaps C programmers don't use these editors. However, it's something worth considering.
>
> I second what Alan said, editing Ruby source with my editor is a nightmare.

Despite the Vim's default behavior which shows the mixed indentation well, my Vim configuration had shown it weirdly (well, I couldn't find the reason to keep the config now, so I fixed it today actually), and I've put .vimrc.local whenever I git-clone ruby.git to correct that. It'd be bothering for people having such configuration in both Emacs and Vim too.

> I wonder wether git filter-branch was considered.
>
> It would allow to fix the formatting while keeping a clean history, the downside being that all commit shas would change.

Making most of shas different would be more problematic than polluting git blame (without -w) because we can't update all of the references to commits in the past. We'd have a control of commit messages and redmine contents, but there are many third-party projects linking Ruby commits, tweets, blog posts, conference talks, etc.

#8 - 08/21/2019 01:25 PM - k0kubun (Takashi Kokubun)

- Related to Bug #14246: Inconsistent C source code indentation added

#9 - 08/21/2019 01:59 PM - k0kubun (Takashi Kokubun)

There's Option 5; altering commits in pre-receive hook on server side. But I think successful git push should never make future git pull fail. It is a very surprising behavior and would confuse committers. So I'm rejecting this option by myself.

By the way, my personal position is taking either 1 or 2, and optionally implementing 3 and/or 4 as well. So I'm supporting all of option 1, 2, 3, and 4.

To help the discussion, let me summarize my current understanding of pros/cons in each option:

0. Disable "expand tabs" in auto-style commit hook

Supporters: @mame (Yusuke Endoh), @jeremyevans

Pros

- No noisy commits just for styling will be made, which makes reverts and backports easier.

Cons

- Since we've seen "expand tabs" commits, taking this option means to possibly start increasing the number of inconsistent indentations, effectively reopening [Bug #14246].

1. Expand all tabs at once for all files managed by auto-style.

Supporters: @alanwu (Alan Wu), @byroot (Jean Boussier), @k0kubun (Takashi Kokubun)

Pros

- It would reduce the number of problems for editors like Sublime Text, VSCode, and Atom.
- More tools will be able to flawlessly show indentation of CRuby's code / patch (Compiler's preprocessed results used in MJIT, Slack text snippet, Code listing of some blog services)
- All other options are very unlikely to achieve the above benefits within coming years.

Cons

- Pollute git blame result when -w option is not specified.
- Tab-indented existing patches and reverts/backports will face conflicts.

2. Apply "expand tabs" only for newly-added lines

Supporters: @k0kubun (Takashi Kokubun)

Pros

- We're very unlikely to see "expand tabs" for new patches written with a proper editor config.

Cons

- We'll see "expand tabs" for old patches, or code written with an outdated editor config.
- It slows down the progress towards consistent indentation, while it does not reverse or stop it unlike the option 0.

3. Prepare a local pre-commit hook to expand tabs in newly-added or edited lines.

Supporters: @jeremyevans, @k0kubun (Takashi Kokubun)

Pros

- People using this tool will not see "expand tabs" for their own commits, and not be bothered for reverting them.

Cons

- This cannot be always applied. People not using this tool will make "expand tabs" commits.

4. Make pull request CI fail when there's diff to be expanded

Supporters: @shyouhei (Shyouhei Urabe), @k0kubun (Takashi Kokubun)

Pros

- Patches merged from GitHub pull requests will not make "expand tabs" commits.

Cons

- Direct pushes to master and pull requests branched from older commits without the CI setting will make "expand tabs" commits.

#10 - 08/21/2019 02:51 PM - jeremyevans0 (Jeremy Evans)

Here are my opinions on the options:

1. I'm in favor of this. I certainly encourage the use of spaces over tabs, but auto commits for indentation changes are unnecessary and I don't think the benefit outweighs the cost.
2. I'm against this as it will make merging old patches more difficult, and that is something I've been doing a lot of in my bug triaging work.
3. I'm against this. If we do keep the "expand tabs", then it should be applied to modified lines as well as new lines, so that we move in the direction of fewer tabs.
4. I'm in favor of this and would use it, since it is easier for me than trying to expand the tabs manually.
5. I'm against this as it places an additional burden on contributors.
6. I'm against this for the same reason as k0kubun (Takashi Kokubun).

#11 - 08/21/2019 03:18 PM - k0kubun (Takashi Kokubun)

Thanks for your comment, Jeremy. Your opinion makes sense except:

> 1. I'm against this. If we do keep the "expand tabs", then it should be applied to modified lines as well as new lines, so that we move in the direction of fewer tabs.

Can you tell me what problem you want to solve by not taking this option, compared to the current situation (not other options)? I do think both the current situation and option 2 solve the problem that "we move in the direction of fewer tabs", so I just failed to interpret your reasoning.

In my understanding, option 2 is a weaker version or a subset of option 0. Therefore, if you agree with option 0 but disagree with option 2, you or I may not be understanding how the other thinks.

Whenever we make a commit which adds new lines, which is the case for many of our commits, option 2 will increase the ratio of space-indented lines in the repository. In that sense it helps "we move in the direction of fewer tabs" for sure. Some commits are not reducing the number of tab-indented lines, but is that different from the current "expand tabs"?

By combining option 3 with the current "expand tabs" behavior (changing indentation of changed lines to spaces as well, probably you felt it's not the case with option 2? If so, it's not my intention) and option 2 with the weaker "expand tabs" behavior, I believe we can move in the direction of fewer tabs for sure.

#12 - 08/21/2019 03:32 PM - k0kubun (Takashi Kokubun)

- Description updated

#13 - 08/21/2019 03:55 PM - jeremyevans0 (Jeremy Evans)

k0kubun (Takashi Kokubun) wrote:

> Thanks for your comment, Jeremy. Your opinion makes sense except:
>
> > 1. I'm against this. If we do keep the "expand tabs", then it should be applied to modified lines as well as new lines, so that we move in the direction of fewer tabs.
>
> Can you tell me what problem you want to solve by not taking this option, compared to the current situation (not other options)? I do think both the current situation and option 2 solve the problem that "we move in the direction of fewer tabs", so I just failed to interpret your reasoning.

Basically, I am in favor of removing the "expand tabs" commits (stop trying to enforce whitespace). However, if we do keep the "expand tabs" commits (keep trying to enforce whitespace), I think we should continue to apply the the enforcement to both modified and new lines. The only reason to enforce whitespace is to eventually get to a point where whitespace is mostly consistent, and if we do not enforce it for modified lines, I don't think we will get there.

> In my understanding, option 2 is a weaker version or a subset of option 0. Therefore, if you agree with option 0 but disagree with option 2, you or I may not be understanding how the other thinks.

I can see what you are saying. And in a linear system, it makes sense. However, in my viewpoint, this is a non-linear system, and it is possible for a middle point (option 2) to be suboptimal to either end point (option 0 and keeping things as they are).

> Whenever we make a commit which adds new lines, which is the case for many of our commits, option 2 will increase the ratio of space-indented lines in the repository. In that sense it helps "we move in the direction of fewer tabs" for sure. Some commits are not reducing the number of tab-indented lines, but is that different from the current "expand tabs"?

It is true that the ratio will be increased. However, in absolute numbers, it does not necessarily move in the direction of fewer tabs (unless tab lines are deleted).

All this being said, I'm not strongly against option 2. I'm more against 4, 5, and 1.

> By combining option 3 with the current "expand tabs" behavior (changing indentation of changed lines to spaces as well, probably you felt it's not the case with option 2? If so, it's not my intention) and option 2 with the weaker "expand tabs" behavior, I believe we can move in the direction of fewer tabs for sure.

I like option 3 because it is optional and something I would use. I don't personally like using tabs.

#14 - 08/22/2019 05:03 AM - duerst (Martin Dürst)

k0kubun (Takashi Kokubun) wrote:

- It would reduce the number of problems for editors like Sublime Text, VSCode, and Atom.

I'm surprised some of the currently most popular editors can't handle such situations. I'm not using one of these editors, but would like to check how the editors I use behave. For this, it would be good to know a file (and some relevant line numbers) that currently has both tab-indented and space-indented lines. Can somebody give an example?

#15 - 08/22/2019 11:24 AM - k0kubun (Takashi Kokubun)

> I like option 3 because it is optional and something I would use. I don't personally like using tabs.

Okay, I'm going to implement this one because nobody has objected to it and all other options had objections from somebody.

> For this, it would be good to know a file (and some relevant line numbers) that currently has both tab-indented and space-indented lines. Can somebody give an example?

It was not my opinion. Perhaps you overlooked the original comment from https://bugs.ruby-lang.org/issues/16112#note-5 because it already did that thing. The attached image seems to be ruby.c.

#16 - 08/22/2019 12:30 PM - k0kubun (Takashi Kokubun)

- Status changed from Open to Closed

Applied in changeset git|15eaedf805fb2727c79a6c59af6d5f6c2a6d634b.

Add misc/expand_tabs.rb ported from auto-style.rb

This is implemented to close [Misc #16112] because all other options got
at least one objection, and nobody has objected to this solution.

This code is a little complicated for the purpose, but that's just
because it includes some historical code for auto-style.rb:
https://github.com/ruby/ruby-commit-hook/blob/918a7c31b69ad2f7b125608c1c6a1f4fd01ec15a/bin/auto-style.rb

Please feel free to improve this file as you like.

[Misc #16112]

#17 - 08/22/2019 01:13 PM - k0kubun (Takashi Kokubun)

This ticket was resolved with 15eaedf805fb2727c79a6c59af6d5f6c2a6d634b.

In addition to that, since people have often objected to "expand tabs" against files which are not related to VM / MJIT (debugging MJIT needs to see preprocessed C code, and tab is converted to 1 space. So it's really helpful if MJIT-related code is space-indented), I also modified auto-style.rb https://github.com/ruby/ruby-commit-hook/commit/ec980bc6e46156e8ab3a96b32084154ff3acb64f to guard only MJIT-related files and some other files which were already indented consistently.
Newly introducing tab indentation to files which were originally indented with only spaces would be clearly violating [Misc #14246], and it should be prohibited "so that we move in the direction of fewer tabs" like @jeremyevans said.

#18 - 05/15/2022 02:58 PM - byroot (Jean Boussier)

> cons: Pollute git blame result when -w option is not specified.

Not sure if this will ever be reconsidered, but just in case:

Git 2.23.0 added an --ignore-revs-file switch for this purpose: https://www.git-scm.com/docs/git-blame/2.23.0

And GitHub added support for it recently as well:
https://docs.github.com/en/repositories/working-with-files/using-files/viewing-a-file#ignore-commits-in-the-blame-view

Files

Screen Shot 2019-08-19 at 2.43.06 PM.png	48.8 KB	08/19/2019	alanwu (Alan Wu)