

Ruby - Bug #14607

Fix use of the rb_profile_frames start parameter

03/15/2018 02:55 PM - dylants (Dylan Thacker-Smith)

Status:	Closed	
Priority:	Normal	
Assignee:	ko1 (Koichi Sasada)	
Target version:		
ruby -v:		
		Backport: 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN
Description		
rb_profile_frames was always behaving as if the value given for the start parameter was 0.		
The reason for this was that it would check if (start > 0) { then continue without updating the control frame pointer or anything other than decrementing start.		
This bug applies to all branches under normal maintenance, from ruby 2.3 to trunk.		

Associated revisions

Revision d676ad1050c0f570d6a2d1f983c1223611d45570 - 04/26/2018 10:49 PM - tenderlovmaking (Aaron Patterson)

Fix use of rb_profile_frames start parameter

rb_profile_frames was always behaving as if the value given for the start parameter was 0.

The reason for this was that it would check if (start > 0) { then continue without updating the control frame pointer or anything other than decrementing start.

[ruby-core:86147] [Bug #14607]

Co-authored-by: Dylan Thacker-Smith Dylan.Smith@shopify.com

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63265 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d676ad10 - 04/26/2018 10:49 PM - tenderlovmaking (Aaron Patterson)

Fix use of rb_profile_frames start parameter

rb_profile_frames was always behaving as if the value given for the start parameter was 0.

The reason for this was that it would check if (start > 0) { then continue without updating the control frame pointer or anything other than decrementing start.

[ruby-core:86147] [Bug #14607]

Co-authored-by: Dylan Thacker-Smith Dylan.Smith@shopify.com

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@63265 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 2b96737636e1c96fedda83895ef32e19a914e310 - 12/28/2023 04:58 PM - Dylan Thacker-Smith

Fix use of the rb_profile_frames start parameter

Previously, it was decrementing the start argument until it reached zero without actually changing the control frame pointer.

[Bug #14607]

Revision 2b96737636e1c96fedda83895ef32e19a914e310 - 12/28/2023 04:58 PM - Dylan Thacker-Smith

Fix use of the rb_profile_frames start parameter

Previously, it was decrementing the start argument until it reached

zero without actually changing the control frame pointer.

[Bug #14607]

Revision 2b967376 - 12/28/2023 04:58 PM - Dylan Thacker-Smith

Fix use of the rb_profile_frames start parameter

Previously, it was decrementing the start argument until it reached zero without actually changing the control frame pointer.

[Bug #14607]

History

#1 - 04/26/2018 10:49 PM - tenderlovmaking (Aaron Patterson)

- Status changed from Open to Closed

Applied in changeset trunk|r63265.

Fix use of rb_profile_frames start parameter

rb_profile_frames was always behaving as if the value given for the start parameter was 0.

The reason for this was that it would check if (start > 0) { then continue without updating the control frame pointer or anything other than decrementing start.

[\[ruby-core:86147\]](#) [Bug #14607]

Co-authored-by: Dylan Thacker-Smith Dylan.Smith@shopify.com

#2 - 04/27/2018 12:15 AM - tenderlovmaking (Aaron Patterson)

- Status changed from Closed to Open

ko1 said I shouldn't have committed the patch, so I reverted. Sorry!

#3 - 04/27/2018 12:30 AM - ko1 (Koichi Sasada)

I need to remember why such special (additional) calculation is done, so I left this ticket.
I need to remember...

#4 - 08/04/2020 07:57 PM - dylants (Dylan Thacker-Smith)

- File deleted (fix-use-of-the-rb_profile_frames-start-parameter.patch)

The original patch has a merge conflict. However, I have opened a pull request with the fix for this issue (<https://github.com/ruby/ruby/pull/2713>) that has been rebased to resolve the merge conflict.

#5 - 04/20/2021 09:59 PM - dylants (Dylan Thacker-Smith)

I need to remember why such special (additional) calculation is done

I'm not sure what you mean by additional calculation. It is decrementing start when non-zero as expected to loop over that number of frames, it just was missing the corresponding update to cfp.

Could this get another look?

#6 - 06/09/2022 06:12 AM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to ko1 (Koichi Sasada)

I assume that "additional calculation" means the code if (start > 0) { start--; continue; }, but the "additional calculation" seems to make no sense at all to me either.

```
for (i=0; i<limit && cfp != end_cfp;) {  
  if (VM_FRAME_RUBYFRAME_P(cfp)) {  
    if (start > 0) {  
      start--;  
      continue;  
    }  
  }  
}
```

It is essentially `start = 0;`. The proposed fix seems reasonable to me. [@ko1 \(Koichi Sasada\)](#) What do you think?

#7 - 01/16/2024 06:14 PM - dylants (Dylan Thacker-Smith)

The original patch has a merge conflict. However, I have opened a pull request with the fix for this issue (<https://github.com/ruby/ruby/pull/2713>) that has been rebased to resolve the merge conflict.

The github PR has been merged, so this issue can be closed now. It doesn't look like I have permission to change its status though.

#8 - 01/16/2024 06:21 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Assigned to Closed