

Redmine - Defect #5324

Git not working if color.ui is enabled

2010-04-15 14:41 - Rodrigo Toledo

Status:	Closed	Start date:	2010-04-15
Priority:	Normal	Due date:	
Assignee:	Jean-Baptiste Barth	% Done:	100%
Category:	SCM	Estimated time:	0.00 hour
Target version:	1.0.4	Affected version:	
Resolution:	Fixed		
Description			
<p>I'm using redmine trunk and git 1.6.0.4. I started a new project, and it didn't work with a working bare repo. Then I found that webrick was outputting this error on its log:</p> <pre>fatal: Not a valid object name master:</pre> <p>with the word "*master" highlighted in green.</p> <p>So I went to the bare repo, and running "*git branch" I saw that the green color was on the branch list too. So I disabled the ui.color locally in that bare repo using:</p> <pre>git config color.ui false</pre> <p>and now redmine sees my repo correctly.</p> <p>So, I managed to fix it on the git side, but it should be good if redmine could be able to see this by itself, and fix it or warn the user.</p>			
Related issues:			
Related to Redmine - Defect #26645: git 2.14 compatibility		Closed	

Associated revisions

Revision 4310 - 2010-10-30 01:21 - Jean-Baptiste Barth

Force color to be disabled on git shell-outs. #5324

Contributed by Felix Schäfer

Revision 4386 - 2010-11-07 16:19 - Jean-Baptiste Barth

Added missing --no-color option in some git shell-outs. #5324

History

#1 - 2010-04-17 14:16 - Jean-Philippe Lang

How Redmine could fix it?

#2 - 2010-04-17 14:17 - Jean-Philippe Lang

- Category changed from Administration to SCM

#3 - 2010-04-17 14:44 - Rodrigo Toledo

My guess is that the issue has to do with some kind of special characters used to put color in the terminal, and that redmine is reading from that output in order to do something with the branch name. If this guess is right, the solution should be stripping that output of any special character before using it for something.

Anyway it's just a guess, I don't know anything about the internal workings of redmine.

#4 - 2010-04-17 19:37 - Rodrigo Toledo

Maybe something like this <http://www.commandlinefu.com/commands/view/3584/remove-color-codes-special-characters-with-sed> but in the context of capturing any git output

#5 - 2010-04-17 23:01 - Bernhard Furtmueller

I had a brief look to:

```
lib/redmine/scm/adapters/git_adapter.rb
```

It looks these commands could need additionally the "--no-color" option:

```
git branch --no-color
```

```
git log --no-color
```

```
git diff --no-color
```

```
git show --no-color
```

```
git blame --no-color <- I'm not sure if it's really necessary here, but it doesn't hurt though.
```

#6 - 2010-04-17 23:23 - Bernhard Furtmueller

- *File git--no-color.patch added*

Rodrigo, could you test the attached patch?

#7 - 2010-04-17 23:23 - Rodrigo Toledo

That should probably work. I tried to write a test to prove it, but I don't know how should I get the git_repository that's used in the unit tests without making it from scratch.

#8 - 2010-04-17 23:49 - Rodrigo Toledo

Somehow I cannot reproduce the issue at home, so I cannot confirm that the patch works. On monday I'll test it at work where it originally happened and I'll give confirmation.

#9 - 2010-04-18 11:37 - Jean-Philippe Lang

I can not reproduce either. I have set:

```
git config --global color.ui auto
```

and everything still works fine (git 1.5.4.3).

#10 - 2010-04-18 13:50 - Bernhard Furtmueller

given that it's not for every one reproducibile, I'd consider git isn't doing the color.ui=auto thing correctly depending on gits version.

For me "git show" is colored, but when piping it to less it's uncolored.

(git version 1.6.3.3)

#11 - 2010-07-08 18:11 - Felix Schäfer

Can anyone confirm this is still an issue?

#12 - 2010-08-22 14:54 - Gustavo Delfino

I am experiencing this exact same problem. git 1.6.2.3 & redmine 0.9.4

#13 - 2010-08-24 03:28 - Rodrigo Toledo

It shouldn't hurt redmine to use --no-color on each command. Just in case.

#14 - 2010-08-30 13:40 - Rodrigo Toledo

I found that in the machine that I had this problem, color.ui was set to "always", maybe this helps anyone trying to reproduce this issue.

#15 - 2010-09-30 17:43 - Felix Schäfer

- *Target version set to 1.0.3*

Eric, the fix for that is here: <http://github.com/thecat/redmine/commit/f9583578af6760377b4d46a5b5907f92c8d47e29>

I don't think it's possible to selectively activate color other than shelling out to make the change, the change no activates color to always, so if anything ever changes in the color stuff in future git versions, we should notice through the tests. If you don't want that in, just ignore the archive of the git repo.

#16 - 2010-10-30 01:16 - Jean-Baptiste Barth

- *Status changed from New to Resolved*

- *Assignee set to Jean-Baptiste Barth*

- % Done changed from 0 to 100

- Resolution set to Fixed

I was able to reproduce too with a recent git version (1.7.0.4) on current trunk, with color.ui set to always and no other color.* behavior defined (see git config -llgrep color ; it seems git options precedence doesn't always work as you guess).

Applied Felix's patch in [r4310](#).

#17 - 2010-11-01 01:22 - Eric Davis

- Status changed from Resolved to Closed

Merged into 1.0-stable for release in 1.0.3

#18 - 2010-11-07 16:14 - Jean-Baptiste Barth

- Status changed from Closed to Reopened

- Target version changed from 1.0.3 to 1.0.4

--no-color still missing in some places. <http://ci.finn.de/> is all green, but I suspect it's because tmp/test/git_repository is still the old one. Fixed in [r4386](#).

I tested it with GIT 1.5.6.5 and 1.7.1, if anyone finds a problem with a different version let us know before it's merged in stable branch.

#19 - 2010-11-07 16:14 - Jean-Baptiste Barth

- Status changed from Reopened to Resolved

#20 - 2010-11-07 16:19 - Felix Schäfer

Jean-Baptiste Barth wrote:

--no-color still missing in some places.

Not all git commands support --no-color (or at least have an explicit colored output...). Did you encounter any test errors? They did all pass when I tested it here.

#21 - 2010-11-07 16:42 - Jean-Baptiste Barth

- File failing_test_5324.txt added

git show supports it at least from 1.5.6.5.

I don't remember having found a failing test when committed it, but I found this one yesterday when investigating something else:

```
% RAILS_ENV=test ruby test/functional/repositories_git_controller_test.rb -n test_diff > failing_test_5324.txt
2>&1
```

See result attached. No output from app/views/common/_diff partial, and when I put the debugger in the view, I saw the diff output was full of color codes.

#22 - 2010-11-07 16:45 - Felix Schäfer

Odd. Anyway, the sure-fire way would be to add it to GIT_BIN, but I didn't want to add it there lest it breaks something else...

#23 - 2010-11-27 16:49 - Jean-Philippe Lang

- Status changed from Resolved to Closed

[r4386](#) merged in 1.0-stable for 1.0.4 release.

#24 - 2017-08-08 05:54 - Toshi MARUYAMA

- Related to Defect #26645: git 2.14 compatibility added

Files

git--no-color.patch	1.8 KB	2010-04-17	Bernhard Furtmueller
failing_test_5324.txt	5.08 KB	2010-11-07	Jean-Baptiste Barth