

Redmine - Defect #10954

Git/Mercurial revision graph is not clear

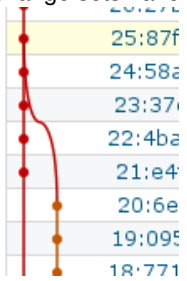
2012-05-17 17:51 - Siebe Joris Jochems

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	SCM	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:			


Description

The revision graph used to show changes in a Mercurial (and possibly Git) repository is not clear. When multiple commits are done in parallel and merged point of merge is displayed weird.

In the screenshot below two change sets have been merged in revision 25. 25 has 24 and 20 as parents. It seems like the two change sets have been merged in revision 24.



In the screenshot below you can see that revision 25 indeed has 24 and 20 as parents.



Revision 25:87f

ID 87f94517b740
Parent 24:58aaec15659b, 20:6eff
Child 26:27b85671e26c

History

#1 - 2012-05-17 18:08 - Siebe Joris Jochems

Forgot to attach info:

Database	MySQL 5.1.61
Ruby version	1.8.7 (i686-linux)
RubyGems version	1.3.7
Rack version	1.1.3
Rails version	2.3.14
Active Record version	2.3.14
Active Resource version	2.3.14
Action Mailer version	2.3.14
Active Support version	2.3.14
Redmine	1.4.1.stable
SCM	Mercurial 2.0
RMagick	2.13.0

#2 - 2012-05-18 00:31 - Toshi MARUYAMA

- Subject changed from Mercurial graph is not clear to Git/Mercurial revision graph is not clear

- Category set to SCM

#3 - 2012-05-19 16:17 - Etienne Massip

Here's a patch, but dunno if it's pretty enough?:

Index: public/javascripts/revision_graph.js

```
-----  
--- public/javascripts/revision_graph.js      (revision 9700)  
+++ public/javascripts/revision_graph.js      (working copy)  
@@ -65,10 +65,12 @@  
         'V', parent_y]);  
     } else {  
         // path to a commit in a different branch (Bezier curve)  
+         sign = parent_x > x ? 1 : -1;  
         path = revisionGraph.path([  
             'M', x, y,  
-             'C', x, y, x, y + (parent_y - y) / 2, x + (parent_x - x) / 2, y + (parent_y - y) / 2,  
-             'C', x + (parent_x - x) / 2, y + (parent_y - y) / 2, parent_x, parent_y - (parent_y - y) /  
2, parent_x, parent_y]);  
+             'C', x, y + 4, x, y + 4, x + sign * 4, y + 4,  
+             'S', parent_x - sign * 8, parent_y - 4, parent_x - sign * 4, parent_y - 4,  
+             'S', parent_x, parent_y - 4, parent_x, parent_y]);  
         }  
     } else {  
         // vertical path ending at the bottom of the revisionGraph
```

#4 - 2012-06-27 13:14 - Michael Sieberer

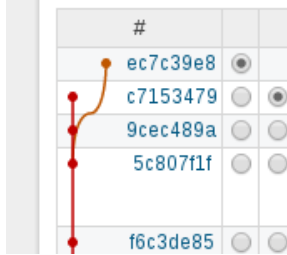
I can confirm the issue with the Git-Backend. In my case, there are about 30 commits between the parent commit and the new branch, which makes it impossible to guess where the two lines meet.

#5 - 2013-01-09 16:54 - C Tibirna

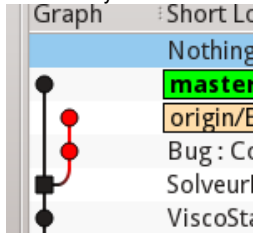
- File *redmine_commit_graph_misleading.png* added

- File *qgit_commit_graph.png* added

And here is an example (with GIT) that shows in what measure this could become debilitating. Where does the branch detach? At 9cec489a? No. At 5x807f1f!



An example of the same commit graph rendered in a "readable" manner, i.e. not only the "branching" should be made clearer, but the latest commit should stay on the "direct" branch, not on the "detached" one:



Another (perhaps separate) issue is that the "Revisions" view (obtained when clicking on "View all revisions" in the "Repository" view) does not identify branches in any way.

Files

File Name	Size	Date	Author
unclear_revision_graph.png	5.67 KB	2012-05-17	Siebe Joris Jochems
revision_parents.png	8.27 KB	2012-05-17	Siebe Joris Jochems
redmine_commit_graph_misleading.png	6.27 KB	2013-01-09	C Tibirna
qgit_commit_graph.png	8.17 KB	2013-01-09	C Tibirna