Redmine - Defect #2071

Reordering priority-enumerations breaks alternate-theme's issue-colouring

2008-10-22 23:10 - Mischa The Evil

Status:	Closed	**Start date:**	2008-10-22
Priority:	Low	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	80%
Category:	UI	**Estimated time:**	0.00 hour
Target version:	2.1.0		
Resolution:	Fixed	**Affected version:**	

Description

The issue-colouring by priority which is done by the CSS of the alternate theme breaks if the order of the enumerations is changed.

This is caused by the fact that the theme uses the class-selector on the priority's position which is generated by Redmine. So let's say the priority low (which should be coloured blue by the theme's CSS) is reordered to position 5, than the CSS-styles "think" it is the highest configured priority and colours it deep red.

Credits goes to AlexC for bringing-up this issue through IRC...

I personally think this is a limitation which can't be fixed easily since we are trying to "link" dynamic priorities to static CSS-styles. Possibly this is the reason why this colouring is only done in the alternate-theme.

Related issues:

Related to Redmine - Defect #6939: Low and Normal priority displaying with sa...	**Closed**	**2010-11-18**
Related to Redmine - Defect #11880: Alternate theme colors by issue priority ...	**Closed**	
Related to Redmine - Feature #12216: Textual CSS class for priorities	**Closed**	

Associated revisions

Revision 10079 - 2012-07-26 18:55 - Jean-Philippe Lang

Changed issues css class from priority-{position} to status-{id} (#2071).

History

#1 - 2008-10-22 23:24 - Mischa The Evil

- File change_issue_color_mechanism_for_alternate_theme.diff added

A possible solution would be to change the issue-view to use the enumeration.id instead of enumeration.position for the class-definition.
That way the alternate-theme can be modified to colour the issues based on the enumeration.id. This at least ensures that the default created (5) priority-enumerations can be re-ordered without loosing the correct colouring.

This workaround doesn't solve the issue when the **amount of enumerations** is changed since than the id's aren't fixated any longer.

I've attached a patch which implements this workaround against r1941.

#2 - 2008-11-08 17:05 - Jean-Philippe Lang

> So let's say the priority low (which should be coloured blue by the theme's CSS) is reordered to position 5, than the CSS-styles "think" it is the highest configured priority and colours it deep red.

If you move a priority to 5, it is actually a high priority and should be colored in red, whatever its name is.

I think that using the priority position to color the issue list makes really more sense than using ids:
If you have 5 priorities, their position will be 1 to 5 even if you deleted the default one and recreated new ones. We can't assume that priorities have ids from 1 to 5.

#3 - 2008-12-08 15:03 - Mischa The Evil

Jean-Philippe Lang wrote:

> > So let's say the priority low (which should be coloured blue by the theme's CSS) is reordered to position 5, than the CSS-styles "think" it is

> > the highest configured priority and colours it deep red.
>
> If you move a priority to 5, it is actually a high priority and should be colored in red, whatever its name is.

So Redmine accepts only 5 priorities which are ordered like by default? Hypothetically, if I were using 10 priorities, priority 5 is expected to be something like medium. How can we make the required info available to the CSS to make this colouring of issue possible without these side-effects?

Jean-Philippe Lang wrote:

> I think that using the priority position to color the issue list makes really more sense than using ids:
> ...
> We can't assume that priorities have ids from 1 to 5.

Agreed! That's also why I added this warning (This workaround doesn't solve the issue when the amount of enumerations is changed since than the id's aren't fixated any longer.) to the patch.

Jean-Philippe Lang wrote:

> If you have 5 priorities, their position will be 1 to 5 even if you deleted the default one and recreated new ones.

True, but if a user decides to modifiy the order of the priorities (so instead of 1 low & 5 high he now re-ordered them to match 1 high & 5 low) or modifies the amount of them, the colouring added by the current CSS-styles isn't sufficient to colour the issues correctly after the changes...

#4 - 2010-12-17 06:14 - Mischa The Evil

The behaviour of Redmine within the scope of this issue hasn't been changed. Any other updates regarding this issue?
It can make it pretty difficult to code themes including issue-prio colouring without having a need to tweak the CSS to adapt the theme manually according to the specific instances' configuration (read: prio-order)...

#5 - 2011-02-28 12:38 - Jan from Planio www.plan.io

We have had this issue over at Planio today, so I'm attaching our patch. It will actually color code the three top most priorities in different shades of red and the lowest priority in blue just like before. However, should you have more than 5 priorities, it will behave nicely and just leave the color for priorities 2 through x, where x = number of prios - 3, alone. It will also make more sense (according to our opinion) for cases where you have 2 priorities, i.e. it will use gray and red and not blue and gray.

#6 - 2011-02-28 12:39 - Jan from Planio www.plan.io

- File patch_for_2071_against_r4961.diff added

Patch is here, apologies for the double post.

#7 - 2011-09-19 19:26 - Robert Hailey

I don't think that patch will do... I think the only options are:

- reference by priority position (what is currently done, breaks re-ordering & addition)
- reference by priority id (will survive reordering & addition, but not portable)
- reference by "nicified" priority name (will not survive a re-name)

#8 - 2012-07-26 18:56 - Jean-Philippe Lang

- Status changed from New to Closed

- Assignee set to Jean-Philippe Lang

- Target version set to 2.1.0

- Resolution set to Fixed

CSS classes are now priority-{id} instead of priority-{position} so that they don't change when inserting/reordering priorities.

#9 - 2012-08-02 21:34 - Philippe Lafoucrière

Hi Jean-Philippe,

I just updated our instance and all colors are messed up.
I reverted the priority-{position} change, and think it's maybe not a good idea: How can we create a theme now? It's barely impossible to guess IDs from redmine instances.
Any chance this could be reverted?

Thanks
Philippe

#10 - 2012-08-03 10:21 - Etienne Massip

Philippe Lafoucrière wrote:

> It's barely impossible to guess IDs from redmine instances.

Yes it is, the id is part of the url used to update or delete the priority element in the administration screen.

> Any chance this could be reverted?

The use of ids is more consistent than the use of the position since the id can't be updated, so the answer is no.

#11 - 2012-09-19 23:52 - Julien Petit

Etienne Massip wrote:

> The use of ids is more consistent than the use of the position since the id can't be updated, so the answer is no.

Shouldn't the alternate theme be updated then ?

#12 - 2012-10-22 21:23 - Ken Zalewski

- % Done changed from 0 to 80

I disagree with the current solution to this issue. Attaching CSS display effects to database IDs (which can vary across installations) is not a good idea.

Robert Hailey's idea above is the best solution: Use textual priorities for CSS coloring, such as "priority-low", "priority-normal", "priority-high", and so on.

In our current Redmine2 installation, the five priorities have database IDs from 3 through 7. Most themes are looking for priorities from 1 to 5. This botches almost every available theme.

Attaching a color to a textual priority will be resilient against reordering and new priority creation, because the color represents the priority itself, not its position. Attaching colors to database IDs is just not a robust solution to the problem.

For now, the New York State Senate is being forced to change its selected theme (modula-gitlab) to match the database IDs (3 through 7). However, I suggest a patch to issue.rb such that the css_class() method utilizes the priority_name, instead of the priority_id.

#13 - 2012-10-23 22:14 - Etienne Massip

Ken Zalewski wrote:

> Robert Hailey's idea above is the best solution: Use textual priorities for CSS coloring, such as "priority-low", "priority-normal", "priority-high", and so on.
>
> In our current Redmine2 installation, the five priorities have database IDs from 3 through 7. Most themes are looking for priorities from 1 to 5. This botches almost every available theme.

I would agree also if the name was not locale-specific.

#14 - 2012-10-26 03:35 - Jean-Philippe Lang

The current css class based on ids will be preserved (for future use) but r10720 adds a textual class that do not rely on ids or the number of priorities. See #12216 for more details.

#15 - 2013-03-31 03:16 - Mikołaj Milej

how about adding additional "css" field to issue priority?
Then you can reorder whatever you like, add any amount of issues and you only have to adjust css name for given issue.
eg:

priority	css
high	i_hate_that
normal	my_css
whatever	whatever_priority

low	my_favorite_priority

In my opinion current solution where default priority act as base priority is very strange. We don't care about which priority is default we care only on priority. What if I'd like to change my default priority? Everything would mess up!.
The other thing is that the priorities numbers raises in strange way:
instead of

- high2
- high1
- default
- low1
- low2

it's:

- high1
- high2
- default
- low2
- low1

In my opinion it's totally weird now...

Files

change_issue_color_mechanism_for_alternate_theme.diff	5.13 KB	2008-10-22	Mischa The Evil
patch_for_2071_against_r4961.diff	5 KB	2011-02-28	Jan from Planio www.plan.io