

Redmine - Defect #10414

Duplicated Message-Id in headers of mails send by Redmine

2012-03-09 17:27 - Jérôme BATAILLE

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Email notifications	Estimated time:	0.00 hour
Target version:		Affected version:	
Resolution:	Invalid		
Description			
If a modification is made on the same issue the message Id can be the same :			
<pre>def self.message_id_for(object) # id + timestamp should reduce the odds of a collision # as far as we don't send multiple emails for the same object timestamp = object.send(object.respond_to?(:created_on) ? :created_on : :updated_on) hash = "redmine.#{object.class.name.demodulize.underscore}-#{object.id}.#{timestamp.strftime("%Y%m%d%H%M%S")}" host = Setting.mail_from.to_s.gsub(%r{^.*@}, '') host = "[:Socket.gethostname].redmine" if host.empty? "<#{hash}@#{host}>" end</pre>			
the message id is made with the created_on date and not with the updated_on date. Usually it's more likely that there is a created_on field when there is no updated_on, and not the opposite.			
Here is a fix :			
<pre>timestamp = object.send(object.respond_to?(:updated_on) ? :updated_on : :created_on)</pre>			

History

#1 - 2012-03-09 17:46 - Jérôme BATAILLE

The only case that seems odd is the table wiki_contents that as an updated_on field but no created_on one.

Anyway the patch as the original code always fallback on the existing field. The patch just finds the more accurate information, updated_on.

#2 - 2012-03-09 17:52 - Etienne Massip

Isn't the message id the conversation id? If yes, it does not seem too weird to have the same id for every update of a specific issue?

#3 - 2012-03-09 18:01 - Etienne Massip

- Target version set to Candidate for next minor release

It is not.

#4 - 2012-03-09 18:09 - Jérôme BATAILLE

That's it, the conversation id is called **references**.

Thanks to have integrated the patch in the next minor release.

#5 - 2012-03-09 19:16 - Etienne Massip

It's candidate =)

From what I've seen, it seems that Original-Message-ID header is more usable then References which should list all the ids of all the messages of the conversation.

#6 - 2012-03-09 22:14 - Jean-Philippe Lang

If a modification is made on the same issue the message Id can be the same

How to reproduce? Because when an update is done on an issue, this is the timestamp of the journal that is used, not the issue timestamp.

#7 - 2012-03-12 16:29 - Jérôme BATAILLE

In the main actual cases the created_on is reflecting the update time.

So there is no case for the moment were you can reproduce the bug because in the only table having the two fields (issues) the date is get through the journal table.

But I suppose it's sort of misleading (and we were mis leaded, lost a few days to understand why messages were not received) when adding new mailer methods, because we were sending mails reflecting changes in the issues, but not using the journal informations.

I suppose again that if other mailer methods are added, they can fall into this case, it's the reason why I proposed the patch.

#8 - 2013-08-20 19:36 - Jérôme BATAILLE

- Status changed from New to Resolved

After digging into the code, I realized the patch I proposed leads to an inconsistency : **references** and **message_id** use the method that I proposed to patch. It implies that the patch breaks the references for an issue update.

I now use message_id on a journal object, thus the message_id is always unique.

Sorry for ma bad understanding of the problem.
I close this issue

#9 - 2013-09-30 15:13 - Jérôme BATAILLE

You can remove it from **Candidate for next minor release**

#10 - 2013-10-01 12:53 - Toshi MARUYAMA

- Status changed from Resolved to Closed

- Target version deleted (Candidate for next minor release)

Thanks.

#11 - 2017-03-31 14:23 - Jérôme BATAILLE

Hi, you can indicate in the Resolution that's it's **Invalid**, if you have the time to, thanks a lot.

#12 - 2017-03-31 14:35 - Toshi MARUYAMA

- Resolution set to Invalid

Done.

#13 - 2017-03-31 14:36 - Jérôme BATAILLE

Thanks !