

Redmine - Feature #1380

Deleting an issue only marks it as being deleted

2008-06-05 02:36 - Anonymous

<b>Status:</b>	New	<b>Start date:</b>	2008-06-05
<b>Priority:</b>	High	<b>Due date:</b>	
<b>Assignee:</b>		<b>% Done:</b>	0%
<b>Category:</b>	Issues	<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>			
<b>Resolution:</b>			
<b>Description</b>			
It would be fantastic if deleting issues only marked them as being deleted, and providing a "recently deleted" link under "change log" link n the right-hand side. Going to this list would give you the option to completely purge items, which would actually deleting the issue from the database.			
I've learnt the hard way - just deleted two issues when I was trying to "move" them using the context menu. I though I clicked "move", quickly clicked OK on the confirmation dialog without reading it (thinking 'yes, I want to move these items'), and watched the issues disappear. Restoring our database backup from yesterday now..... :(			
<b>Related issues:</b>			
Related to Redmine - Feature #6666: Soft delete of issues		<b>New</b>	<b>2010-10-14</b>
Related to Redmine - Feature #7900: Restore issue		<b>New</b>	<b>2011-03-17</b>
Related to Redmine - Feature #18027: Notification when deleting an issue		<b>New</b>	
Has duplicate Redmine - Defect #25595: Important usability bug - Too easy to ...		<b>Closed</b>	

History

- #1 - 2008-06-05 12:00 - Thomas Lecavelier**  
I like this idea, which is very close to enterprise requirements. Maybe we could add it an option in the admin panel to bypass this new feature in order to keep the old way to do...
- #2 - 2008-07-02 12:22 - Mark Gallop**  
My workaround was to disable deletion of issues by normal users and give them a "Trash" tracker to move issues to. I then use a rake task to delete "trashed" issues if they were moved more than 2 weeks ago.  
  
A little clunky but at least it can be accomplished without altering Redmine.  
  
Mark
- #3 - 2009-07-20 22:32 - Andrew Ness**  
I would really like this to be implemented. I just came back from vacation and a couple of issues were unexpectedly deleted. I assumed they would just be flagged as deleted, but their rows in the issues table have actually been removed. I'll try to merge them back in from database archives, but it seems that implementing this feature would make things a lot easier.
- #4 - 2009-07-21 03:02 - Aaron Soules**  
There is an excellent rails plugin called acts\_as\_paranoid that could potentially be used. I sets a deleted\_at field on the table, and modifies ActiveRecord so that it does not find things deleted in the past.  
  
[[[http://github.com/technoweenie/acts\\_as\\_paranoid/tree/master](http://github.com/technoweenie/acts_as_paranoid/tree/master)]]
- #5 - 2013-03-18 16:49 - Dipan Mehta**  
+1. Very necessary to have this feature.
- #6 - 2013-07-08 12:06 - Stéphane Champeau**  
+1 today I had this problem with one user. The idea of "trash" looks good.
- #7 - 2013-11-08 14:36 - Lázaro Hermoso**  
+1  
[#6666](#)

[#7900](#)

**#8 - 2013-11-30 05:00 - Toshi MARUYAMA**

- Related to Feature #6666: Soft delete of issues added

**#9 - 2013-11-30 05:00 - Toshi MARUYAMA**

- Related to Feature #7900: Restore issue added

**#10 - 2014-05-13 14:16 - Lubos Racansky**

+1

**#11 - 2014-06-19 17:32 - Jose Daniel Narvaez Bauste**

+1

**#12 - 2014-06-24 12:00 - claude g**

+1 Also, in case of deletion (this could be a dedicated feature) : we should have a notification with the all (and final) history of the tracker. Why not like a pdf export if easier but a full history.

**#13 - 2014-06-25 06:30 - Steven M**

Looks like maintainer does not care about his users. When is this feature going to be implemented?

**#14 - 2014-10-03 16:18 - Toshi MARUYAMA**

- Related to Feature #18027: Notification when deleting an issue added

**#15 - 2015-02-07 02:44 - Sam XIAO**

+1

Dipan Mehta wrote:

+1. Very necessary to have this feature.

admin made a mistake delete a issue. and then want to bring it back.

and no email notification configured, so i have no any history information about the deleted issue.

**#16 - 2015-02-07 02:47 - Sam XIAO**

i want to add delete trigger on the table issue, can i?

**#17 - 2015-02-25 11:22 - Kevin Palm**

+1

**#18 - 2015-03-16 15:28 - Daniel de Wildt**

+1

**#19 - 2015-03-24 17:03 - Rudolf Osman**

+1

**#20 - 2015-04-06 09:52 - Đức Long Nguyễn**

Good idea! I like it

Nick Read wrote:

It would be fantastic if deleting issues only marked them as being deleted, and providing a "recently deleted" link under "change log" link n the right-hand side. Going to this list would give you the option to completely purge items, which would actually deleting the issue from the database.

I've learnt the hard way - just deleted two issues when I was trying to "move" them using the context menu. I though I clicked "move", quickly clicked OK on the confirmation dialog without reading it (thinking 'yes, I want to move these items'), and watched the issues disappear. Restoring our database backup from yesterday now..... :(

**#21 - 2015-04-16 08:27 - Ami Desai**

+1

**#22 - 2015-05-26 09:03 - Paco Alcaide**

+1

I would like too.

**#23 - 2015-07-21 11:10 - Matthew Palencia**

+1

**#24 - 2016-01-18 03:55 - Juntao liu**

+1

**#25 - 2016-02-18 03:46 - Anh Le Giang**

+1 this is a "must have" feature nowadays

**#26 - 2016-06-09 16:15 - Fred B**

This feature is mandatory. +1000!

Any development? Plugin that can do something similar? Main goal is to have a way to rollback. Or at least to prevent deleting by replacing the Delete button for a Archive function.

**#27 - 2016-08-09 11:46 - Yevgen Vor**

+1 . Guys, please, I don't want to come back to Jira hell.

**#28 - 2016-08-10 12:52 - Laurent Barbareau**

+1

Unbelievable that there's at least no confirmation message !

Managing a deletion flag instead of a pure deletion is certainly tricky to implement now but backing up the records prior to delete them, into dedicated tables or even a kind of log file, should be quite easy to implement, no ?!

**#29 - 2016-08-29 09:28 - Alexander Dyachkov**

+1

I was really amazed when didn't find this feature in new & stable version of Redmine (after occasionally deleting an issue, of course). According to this issue's info, it's 8 years old - please devs, do something about it.

**#30 - 2016-10-26 10:11 - Steve Hanselman**

I think this is an essential feature, however, after 8 years it's not looking likely!

The ideal would be to mark as deleted then have a job that hard deletes after a set period of time (configured from settings).

**#31 - 2016-11-22 09:48 - Inese Ez**

+1

**#32 - 2017-01-19 18:13 - Fred B**

Essential feature!

Anyone working on this?

**#33 - 2017-05-07 15:14 - Randa Bustami**

Fred B wrote:

Essential feature!

Anyone working on this?

its very important to know who delete and when, if you can generate report from redmine log file would be great

**#34 - 2017-08-27 08:26 - Randa Bustami**

please please

its very important to know who delete and when, if you can generate report from redmine log file would be great

#### #35 - 2017-10-05 16:32 - Jaime Irazo

+1

Please, essential feature!

#### #36 - 2017-11-12 04:54 - Karl Wilbur

+1

Not being able to track deletion of, and potentially restore, deleted issues is significant shortcoming.

This is a seemingly simple thing to implement. Why has it not been implemented in **9 years**?

#### #37 - 2017-12-11 13:00 - Aleksandar Pavic

There are some plugins helping with this but +1...

However there might be trouble giving proper statistics with tickets...

Statistics would have to include "if not some\_field=deleted" ...

#### #38 - 2018-05-15 15:22 - Anton Sergeev

Thanks to **Guy Baconniere** for him [#1380#note-24](#). I restore deleted *Issue* (issue\_id=3381) in **PostgreSQL** using this:

```
createdb restore_redmine_db
zcat my_redmine_DB_dump.sql.gz | psql restore_redmine_db

tables='
issues;id
journals;journalized_id
journal_details;journal_id
custom_values;customized_id
attachments;container_id
time_entries;issue_id
issue_relations;issue_from_id
'
issue_id=3381
for i in $tables; do
  _t=${i%;*}; _id=${i#*};; echo "t=$_t; _id=$_id";
  psql --command="COPY (SELECT * FROM $_t where $_id = ${issue_id}) TO STDOUT;" restore_redmine_db | psql
--command="COPY $_t FROM STDIN;" original_redmine_db
done

dropdb restore_redmine_db
```

In addition, you need to restore attachment files. Their locations can be found with this:

```
RAILS_ENV=production bin/rails r "Attachment.where(container_id: ${issue_id}).sort.each do |_a| puts _a.diskfi
le; end"
```

#### #39 - 2019-02-17 07:49 - Yuuki NARA

+1

As a workaround, there is also a way to simply change the project\_id of the issue.  
(Project\_id -> 0, -% Project\_id%,% Project\_trashcan%)

The following requirements can be satisfied.

- 1.Even if you specify the ticket number, it is not displayed on the screen
- 2.The ticket is not displayed on the Project issue list screen
- 3.Data on the issue can not be changed.
- 4.The administrator can easily return to the original state when necessary.

#### #40 - 2019-09-17 17:29 - Peter Lustig

Rename buttons on-the-fly via reverse proxy Nginx (nginx-extras on Ubuntu 18.04). Of course you need proxy\_pass with include proxy\_params to do this.

```

# Umlaute are OK.
# g = Replace all the match strings.
# i = Perform a case-insensitive match.
# o = Just replace the first one.
# r = The pattern is treated as a regular expression, default is fixed string.
subs_filter '(<a class="projects(?: selected)?" href="/projects">Projects)' '<a class="projects" href=
"/issues">Issues</a></li><li>$1' or;
subs_filter '(<a class="projects(?: selected)?" href="/projects">Projekte)' '<a class="projects" href=
"/issues">Tickets</a></li><li>$1' or;
subs_filter '(<a[^>]+href="/issues[^>"]+>")Löschen</a>' '$1Das gesamte Ticket löschen</a>' gr;
subs_filter '(<a[^>]+href="/issues[^>"]+>")Delete</a>' '$1Delete entire ticket</a>' gr;

```

It has happened to us many times that people wanted to delete their comment quickly and deleted the whole ticket because the button is just Delete. With this solution, the probability of this happening is minimized.

#### #41 - 2020-05-01 10:35 - Miodrag Milic

During the last 10 years people deleted accidentally countless tickets.

There is no easy way to recover them.

I found that setting up RSS client to record each and every ticket change is probably the easiest way (files are gone tho but this can be downloaded by some clients too).

#### #42 - 2023-05-06 00:02 - Go MAEDA

- Has duplicate Defect #25595: Important usability bug - Too easy to delete issues added

#### #43 - 2023-06-01 00:19 - Martin von Wittich

Anton Sergeev wrote in [#note-38](#):

Thanks to **Guy Baconniere** for him [#1380#note-24](#). I restore deleted *Issue* (issue\_id=3381) in **PostgreSQL** using this:

[...]

In addition, you need to restore attachment files. Their locations can be found with this:

[...]

I'm pretty sure this does not correctly restore journal\_details. The code attempts to match journal\_details.journal\_id with the ticket ID, but as far as I can tell, journal\_details.journal\_id references journals.id.