

Redmine - Feature #1675

Add 'affected version' as a standard field

2008-07-23 12:17 - Anonymous

Status: Closed	Start date: 2008-07-23
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category: Custom fields	Estimated time: 0.00 hour
Target version:	
Resolution: Fixed	
Description	
<p>As described in this forum post , the way the Redmine project uses affected versions isn't usable by any redmine installation that has multiple projects.</p> <p>Because custom fields are not on a per-project basis, we'd have to have separate 'project X affected version' and 'project Y affected version' fields as each project has different version numbers.</p> <p>Affected version could be a standard field on all projects that have a 'target version' field. That way you can easily do a query on all issues that exist in a particular release.</p>	
Related issues:	
Related to Redmine - Feature #685: New Custom Field "Found in Version"	Closed 2008-02-18
Related to Redmine - Feature #1266: Feature: Allow setting multiple target-mi...	New 2008-05-20
Related to Redmine - Patch #7443: Found-in-version patch	Closed 2011-01-25
Related to Redmine - Feature #2096: Custom fields referencing system tables (...)	Closed 2008-10-27
Related to Redmine - Feature #284: Issues fixed in multiple versions	Closed
Related to Redmine - Feature #8572: Configuration of which versions (by versi...	Closed 2011-06-09
Has duplicate Redmine - Feature #3000: New issue - Affected version droplist	Closed 2009-03-18
Has duplicate Redmine - Feature #5879: Affected version field in issue types	Closed 2010-07-14
Has duplicate Redmine - Feature #7571: "Affected version" as a Ticket field l...	Closed 2011-02-07
Has duplicate Redmine - Feature #5996: Affected version feature	Closed 2010-07-29

History

#1 - 2008-07-23 12:19 - Anonymous

Related this to issue 1245 as the issue came up in the comments for that issue too.

#2 - 2008-08-04 03:11 - Mischa The Evil

+1 Though this seems to be also becoming difficult due to the fact that you can have multiple branches (versions) out in the field which are indeed affected by one issue. Also those versions can be "closed" already.

So do we allow this "affected version" field to contain multiple versions for one issue? And do we allow issues with the affected version set to a "closed" version?

Also important to mention in this issue is that this "affected version" field should ideally populate it's contents from the project's versions...

#3 - 2008-08-04 23:01 - Sven Schuchmann

I think it is enough to allow only one version to be affected.

I already mentioned a "found in version" field which seems to be the same as "affected version" over here [#685](#)

#4 - 2008-08-05 01:22 - Anonymous

Sven Schuchmann wrote:

I think it is enough to allow only one version to be affected.

Strongly disagree. Working on a large project that has **many** version in the field, it is common that an issue is reported by different customers and affect multiple released versions. Add to this any unreleased versions that our internal QA team finds.

This same logic can (and should) be applied for multiple **fixed** versions (for different branches), as can be seen in my feature request in [#284](#) and comment in [#1266](#).

#5 - 2008-08-05 05:10 - Mischa The Evil

Nick Read wrote:

Sven Schuchmann wrote:

I think it is enough to allow only one version to be affected.

Strongly disagree. Working on a large project that has **many** version in the field, it is common that an issue is reported by different customers and affect multiple released versions. Add to this any unreleased versions that our internal QA team finds.

This same logic can (and should) be applied for multiple **fixed** versions (for different branches), as can be seen in my feature request in [#284](#) and comment in [#1266](#).

I agree with Nick.

#6 - 2008-08-05 09:45 - Anonymous

Nick Read wrote:

Strongly disagree. Working on a large project that has **many** version in the field, it is common that an issue is reported by different customers and affect multiple released versions. Add to this any unreleased versions that our internal QA team finds.

This same logic can (and should) be applied for multiple **fixed** versions (for different branches), as can be seen in my feature request in [#284](#) and comment in [#1266](#).

While I agree that issues will be found in multiple versions and also therefore fixed in multiple versions, that can already be achieved by relating tickets and therefore creating multiple tickets, one for each version it was fixed in. That way, you have a specific ticket to close when you merge the fix to the appropriate stable branch.

I'd be happy with it implemented either way, but if a single affected version and target version would make it easier (therefore quicker) to implement, then I'd be happy with that.

Cheers

Russell

#7 - 2008-08-05 10:41 - Simone Carletti

I just want to cast my vote for this feature.

#8 - 2008-08-05 16:15 - Mischa The Evil

Russell Hind wrote:

Nick Read wrote:

Strongly disagree. Working on a large project that has **many** version in the field, it is common that an issue is reported by different customers and affect multiple released versions. Add to this any unreleased versions that our internal QA team finds.

This same logic can (and should) be applied for multiple **fixed** versions (for different branches), as can be seen in my feature request in [#284](#) and comment in [#1266](#).

While I agree that issues will be found in multiple versions and also therefore fixed in multiple versions, that can already be achieved by relating tickets and therefore creating multiple tickets, one for each version it was fixed in. That way, you have a specific ticket to close when you merge the fix to the appropriate stable branch.

I'd be happy with it implemented either way, but if a single affected version and target version would make it easier (therefore quicker) to implement, then I'd be happy with that.

If you look at it that way i'm going to doubt since that approach could be workable too imo. Though I'd then recommend to add a specific issue-relation for this kind of ticket-branch_relationship.

Either way it will be a change to redmine with a higher severity imo. At the moment I'd vote therefor to further discuss the pros and cons of either implementation before changing anything already :-)

#9 - 2008-08-06 01:26 - Anonymous

Russell Hind wrote:

Nick Read wrote:

....

While I agree that issues will be found in multiple versions and also therefore fixed in multiple versions, that can already be achieved by relating tickets and therefore creating multiple tickets, one for each version it was fixed in. That way, you have a specific ticket to close when you merge the fix to the appropriate stable branch.

I'd be happy with it implemented either way, but if a single affected version and target version would make it easier (therefore quicker) to implement, then I'd be happy with that.

See comment no.3 in [#1266](#) for my reasons against this data duplication against tickets. Tickets are not a static thing; they get added to, edited, moved around. Unless the relation type is special enough to perform automatic data updates across the related tickets, then it's not suitable. I think I'm going to have to rewrite my multi-fixed versions patch for the latest trunk - the same logic could be used for a multi-affected versions patch.

#10 - 2008-08-08 06:32 - Patrick Oppenlander

+1

#11 - 2008-10-10 10:55 - Jędrzej Sieracki

+1

#12 - 2008-10-13 10:27 - Anonymous

+1

#13 - 2008-10-22 15:43 - Guilherme Schneider

+1. And I agree with Russell.

#14 - 2008-10-26 18:30 - Andrea Campi

+1, it's absolutely needed at my company.

#15 - 2008-10-27 01:53 - Kihyun Yun

+1

#16 - 2008-11-05 15:57 - Luiz Carlos Junior

Ticket [#2081](#) is also related to this feature.

#17 - 2008-11-05 16:03 - Luiz Carlos Junior

I don't know if this was already proposed, but what if the custom field "List" was extended in order to allow multiple selection instead of just one?

In this case, the combo box would appear with checkboxes on the side of each option, allowing multiple selection.

I think this feature would allow all requests above and the "Assign a task to multiple users" (ticket [#408](#))

#18 - 2009-11-23 16:24 - Dan MacMillan

+1

This is an important feature.

#19 - 2010-04-01 11:08 - Milos Zikic

Do we have any update on this request?

This is something that is very very needed. Redmine.org has it :)

It would be nice to have a custom field type that is called 'Version' so it will pick up values from project itself. It can be the same single/multiple choice option as in filters.

#20 - 2010-05-17 05:00 - Nathan Rijksen

+many

I am surprised this isn't implemented yet, nor has anyone bothered making a plugin for it.

#21 - 2010-07-20 10:20 - Roberto Marcellino

As many other guys stated, this feature would be very helpful, it would allow the testers job to be integrated into the development iterations. Please...implement it...

#22 - 2010-07-21 21:29 - Pieter Smith

+1

#23 - 2010-07-22 14:12 - Curtis Stewart

I do believe that this would help with any of my software projects. However it should not be mandatory. I have a large number of projects that are not related to software. They just want to use this application to manage their day to day work. In fact some would like to remove a couple of the required fields that they are not using, but that is another issue. The Redmine application has proven to be an effective and simple way to manage many projects, not just software.

#24 - 2010-08-19 15:05 - Rob D

+1

#25 - 2010-08-23 08:45 - Be Fio

Definitely a must have. Should've been a standard feature already. It'd be pretty easy for the affected version field to pull values from the target version anyways.. +1

#26 - 2010-09-18 09:00 - Anthony Meugui

+1

#27 - 2010-11-12 23:14 - James Moore

+1

#28 - 2011-01-06 01:16 - Brian Lindahl

- *File affected-version.patch added*

Here's a patch you can apply to redmine-1.0.4 to add the 'Affected Version' feature. You can only select 'locked' versions (in my world, these are versions that have made it past integration and internal release). It took me about 3 hours to write, since I don't know my way around the source code and this is my first work in the Ruby language.

/rant

For something **SO EASY** to implement, there's a terrible amount of resistance to implementing this feature. Why? If it's only because people want a multi-selection ability, and that's difficult, who cares! Since it's so **EASY** to implement, so **INCREDIBLY** useful, and so **DESIRED**, implement the single-selection feature now, so that people can start taking advantage of it now! When multi-selection ability is added later to 'Target Version', just add it for 'Affected Version' at the same time.

The resistance to add this feature is just a silly dragging of feet (for 2 years!!!) that I wouldn't expect to see in an open-source project. Heck, I'd be surprised to see this kind of resistance in anywhere but a government contractor waterfall-development world! Are you guys that clueless about the virtues of agile development?

/end rant

Now that I'm done ranting, I love your product... except for the lack of this incredibly important feature - it was my biggest resistance to bringing it into my company. Bring my patch into the mainline and finally resolve this issue. Thanks!

#29 - 2011-01-06 01:37 - Brian Lindahl

- *File found-version.patch added*

Actually, on second thought, it should be named 'Found in version'. Since 'Found in version' is quite different from 'Affected versions'. 'Found in version' is the version it was found in, and is filled in by the submitter. It can and should only be a single-selection for reproduction's sake. 'Affected versions' is filled in by the software engineers when doing version analysis, and can be (and likely is) more than one version. Here is the new patch with the different name.

#30 - 2011-01-07 15:27 - Attila Antal

These are very important features for us! We are a governmental software development company, we have a bit difficult product and release management, so I'd like to promote these features in the base installation. Thanks!

#31 - 2011-01-13 06:33 - Ryan Thrash

+100

I have to concur that this is a sorely missing and critical piece of managing projects. Manually maintained custom enumerations per-project is weak workaround that we painfully have to endure.

#32 - 2011-01-19 16:23 - Vincent Frédérick

+1 Currently working manually.

#33 - 2011-01-19 17:01 - Patrick Dubois

+1, please incorporate the "Found version" patch in the trunk.

Also adding an "Affected versionS" field which could contain multiple versions would be great.

#34 - 2011-01-25 19:28 - Brian Lindahl

Created a patch issue to follow the progression of the 'found-version.patch' patch file I created.

#35 - 2011-02-15 18:26 - Axel Müller

+1, this is a basic feature of every TTS and the pathc is already there - please incorporate it in the release. Not everybody wants to build from sources ...

#36 - 2011-03-25 17:01 - Carlo Alberto Degli Atti

+1 it would be great!

#37 - 2011-05-18 18:27 - Gilles Cornu

+1!

- We are using Redmine for more than 2 years with many projects and subprojects. We tried so far to make it without any "found in" field, but we miss it more and more...
- We are also going to merge/import some Trac projects into Redmine (and Trac provides by default the 'found in' field) Too bad to lose this information during the migration...
- Could at least the Data-Model/Architecture of the patch be reviewed and validated by a Redmine core developer ? (goal: a patched system could thus smoothly be upgraded when the feature will be native in Redmine release 1.x). With no such warranty, I'm not ready to apply any patch... (maybe I'm too careful, please correct me if I'm wrong...)

#38 - 2011-05-18 18:32 - Etienne Massip

What about using a Version custom field as it will be available with [1.2.0](#) release (see [#2096](#)) ?

#39 - 2011-05-18 20:40 - Gilles Cornu

Etienne Massip wrote:

What about using a Version custom field as it will be available with [1.2.0](#) release (see [#2096](#)) ?

Thanx a lot for the hint! (linking [#2096](#) to [#1675](#) as 'related' would help to know about this nice feature)

Maybe the only drawback will be that *unreleased* versions are also in the dropdown list (but it's not a big problem, at least for my team)

I'll study ASAP how I can manage with Trac migration...

What about [1.2.0](#) release date ? I noticed in [RedmineInstall](#) that Rails and Rack requirements will upgrade with 1.2.x compared to 1.1,1.0,0.9. I hope it won't be a problem to upgrade my server to be compatible with this new stack.

#40 - 2011-07-03 02:08 - Gilles Cornu

Gilles Cornu wrote:

- We are also going to merge/import some Trac projects into Redmine (and Trac provides by default the 'found in' field) Too bad to lose this information during the migration...

Etienne Massip wrote:

What about using a Version custom field as it will be available with [1.2.0](#) release (see [#2096](#)) ?

I upgraded to Redmine [1.2.0](#) and successfully imported Trac, with following add-ons in [lib/tasks/migrate_from_trac.rake](#)

IMHO, this ticket could now be closed since **single-affected-version** is possible to *configure* with Redmine ≥ [1.2.0](#)

Thanx and Congratulations!

#41 - 2011-07-06 12:31 - kurumi kurogi

Hello,

- I would like to install the proposed patches, but I don't know how to do? Could you help me please ?

- Is it possible to install them with the last version of Redmine?

Thanks.

#42 - 2013-03-18 20:01 - Dipan Mehta

As such, now Redmine support Version type custom field - which sufficiently apply for this stated usecase. We also use this in the same manner as well. However, in my opinion, Affected_version or origin_version MUST be standard property of the Redmine core.

Given that it would be a standard field, it makes it possible to create more functionality around it.

1. Specifically, I would like (may be a plugin or so) about the statistic about which specific issues and versions have resulted into many bugs, (which references those affected versions).
2. One can also automatically create/link Relevant tests based on all test cases that have been applied on Affected versions.
3. One can make developer/tester assessment based on number of bugs filed against version where particular user was developer or tester.
4. One can develop software reliability matrices by combining maximum bugs affected on versions - and modules that got modified on during those versions.

Now may be many such features are far beyond the scope of the ticket describing above, and also quite a few plugins as well, but the limited point here is that "Affected version" should really be a standard attribute of bugs by all means.

#43 - 2013-03-28 15:56 - Sebastian Denz

After reading this and other issues i was happy to find the new type "Version" for custom fields, but there is one big problem if you want to use this type for "found in version".

In my case the custom field of type version only show the upcoming versions and not the already closed from the past...

But for reporting bugs for already released versions, it would make sense to let the reporter choose from the old/closed versions instead of the future versions.

i think, if we would have another type like "closed versions", that should do the trick! :)

edit: while looking at redmine.org's "new issue", i was wondering if the dropdown menu is configured as a list (manually maintained?!) or if there is a trick to let the type Version even show the old/closed versions?!

#44 - 2014-08-15 02:29 - Toshi MARUYAMA

- Related to deleted (Feature #284: Issues fixed in multiple versions)

#45 - 2014-08-15 02:29 - Toshi MARUYAMA

- Related to Feature #284: Issues fixed in multiple versions added

#46 - 2016-12-26 16:14 - Go MAEDA

- Related to Feature #8572: Configuration of which versions (by version-status) are shown in version-format custom fields added

#47 - 2016-12-26 16:14 - Go MAEDA

- Status changed from New to Closed

- Resolution set to Fixed

Sebastian Denz wrote:

In my case the custom field of type version only show the upcoming versions and not the already closed from the past...

But for reporting bugs for already released versions, it would make sense to let the reporter choose from the old/closed versions instead of the future versions.

In Redmine 2.5.0 and later, we can configure which status of version are shown in a version type custom field ([#8572](#)). So we can realize 'affected version' described in this issue by using version type custom field.

version-type-custom-field.png

Files

affected-version.patch	18.6 KB	2011-01-06	Brian Lindahl
found-version.patch	18.6 KB	2011-01-06	Brian Lindahl