redmineorg-copy202205 - Vote #63098

Allow email to create issue or add comment

2022/05/09 13:19 - Admin Redmine

ステータス:	Closed	**開始日:**	2008/04/25
優先度:	通常	**期日:**	
担当者:		**進捗率:**	100%
カテゴリ:	Email receiving_29	**予定工数:**	0.00時間
対象バージョン:	0.8_2	**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/1110	**status_id:**	5
category_id:	29	**tracker_id:**	3
version_id:	2	**plus1:**	0
issue_org_id:	1110	**affected_version:**	
author_id:	674	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	35		

説明

h2. Summary

This is a patch for #1083 that implements an email conduit for accepting issue creation or comments via email. It's implemented as a standalone script which can be placed in your sendmail aliases configuration to receive email messages from an alias.

h2. Configuration

The user is responsible for the email alias configuration. There is some flexibility in configuration. The script gathers the following parameters in order to create a new bug:

- issue
- project
- subject
- description
- status
- tracker
- priority
- category
- author

These parameter can either come from the command line parameters given to the bugmail.rb script, or from the contents of the email message. If the parameter is specified in the email, it takes the form 'Param: value of param' where the parameter name is capitalized, and the value is whatever is left on the line. The exception is 'Description:' which takes whatever text is left to the end of the email. MIME attachments are not supported (yet... any takers?)

There is an option in the script to allow override of command line parameters by email information. This provides some measure of control over the bug creation process. For example:

h3. Use special tracker for bugmail

If you want to use a special tracker, configure the tracker in your redmine project. Add the '--tracker TrackerName' parameter to the script in your /etc/aliases file, and don't use '--allow-overrides' This will force the tracker parameter to whatever you specify, regardless of the contents of the email.

h3. Specifying project parameter

There are three options for specifying the project. One is to use the '--use-email-header' option, which grabs the project identifier from the 'To:' field of the email by decoding the address projectident@hostname.com.

Another option is to use the '--project projectid' option and don't use '--allow-overrides'.

A third option might be to use one alias, (e.g. redmine@hostname.com) and require the 'Project:' tag to be present in the email.

h4. Specifying the author

There are two ways of specifying the author. Either via the 'Author:' tag in the email body, or by matching the 'From:' field in the email headers with the email address of a particular user (if --use-email-header option is active).

h2. Adding Comments

In order to add comments, the only real information required is the 'Issue:' id and the 'Description:' text, which will be used as the comment. Again, MIME attachments are not supported (yet...)

h2. Script Usage

```
Usage: [--help] [--debug] [--allow-override] [--use-email-header] [--file ] [-- ...]
  --help      show this help
  --debug     show extra debug information
  --allow-override      allow message contents to override cmd line args
  --use-email-header  search in email header for user, subject and project
  --file            get input from a file instead of STDIN
  --      set a parameter's value.   Parameter is one of:
                         issue, project, subject, desc, status, tracker, priority
                         category, author
```

h2. Example email

```
Received: by 10.114.159.7 with HTTP; Wed, 23 Apr 2008 19:40:37 -0700 (PDT)
Message-ID: <5faca86f0804231940r7b170d3cp7b22cdb0b43586a7@mail.gmail.com>
Date: Wed, 23 Apr 2008 22:40:37 -0400
From: "Carl Nygard"
To: "blech"
Subject: This is an issue created via email


Category: Ticket
Priority: Normal
Tracker: Feature
Description: This is a multiline
description about
the problem

h2. with wiki *formatting*
```

journals

Great job, I've been wanting to create some sort of email input for Redmine but haven't found the time. So this is a command line script that is feed each email message as text (STDIN)?

h3. Suggestions

If you use Rails ActionMailer, you can save a lot of the manual parsing and even support attachments.
http://wiki.rubyonrails.org/rails/pages/HowToReceiveEmailsWithActionMailer

Requiring the entire Rails environment for each email can get really costly, especially under heavy loads. I was planning on having Redmine check an IMAP account regularly and batch process emails from it. It will save the startup time and would also make setup easier (i.e. just setup an IMAP account).

I'd love to see this implemented!

This plugin seems pretty close to doing as described, with ActionMailer and a rake task for running in a cron job.

http://github.com/mully/redmine_ticket_emailer/tree/master

It seems there might be some elements of it that are holdovers from the plugin it was maybe adapted from (Simple CI, maybe)? And it doesn't have the "add Journal" function shown in the script attached.

In short, it might only be "almost" right, but it seems like a darned good start. Has anyone seen it before? Tried it?

Randy Harmon wrote:

> In short, it might only be "almost" right, but it seems like a darned good start. Has anyone seen it before? Tried it?

Yea I've taken a look at it and it is great start. This week I'm going to try to merge that plugin with the parser from @bugmail.rb@ and see what I can get.

I've just forked mully's plugin and added in Carl Nygard's parser from @bugmail.rb@. So far it's working really good for adding new issues and only needs a little bit more to support updating issues. You can follow my progress at:

http://github.com/edavis10/redmine_ticket_emailer/tree/master

Eric Davis wrote:

> I've just forked mully's plugin and [...]

Sweet. There might be some good stuff in this fork too. I don't agree with all of it (user creation from email? yuck), but some might be good.

http://github.com/rollick/redmine_ticket_emailer/commit/ac2b1050d67527aa12106949821ba639e2d89dd0

Randy Harmon wrote:

> I don't agree with all of it (user creation from email? yuck)

This is my fork. User creation was ok for me as I was filtering the incoming emails. Yes, it is still yuck.

How would you propose dealing with users/authors not in the system? I can't ignore them, nor can I set the author to an "anonymous" user as I need the email (and name) of the issue author.

Mark Gallop wrote:

> How would you propose dealing with users/authors not in the system? I can't ignore them, nor can I set the author to an "anonymous" user as I need the email (and name) of the issue author.

For my case, it's safe to ignore them because there are only a few people who will be emailing the system.

If you wanted to create a user (say for automated bug reports) then the idea that you have is fine. I got some more time allocated to work on it next week, and will be adding a configuration panel which will allow us to get rid of the yml file. I think it would be great to have some options in there for creating new accounts:

- Allow new users to be added to the project
- Role to add new users at

Eric

Clearly some people would want to capture user-info. Just because I don't want it for my environment doesn't mean it's bad. I do wonder if it's a vector for spammers to exploit, but some careful thought (probably already done by you, Mark) can probably address that.

Another method would be to augment the model with a field to capture this information; assign tickets to the "incoming email" user; and allow people (with sufficient privileges) to promote that person to a User object (fixing up Reported-By in the bargain).

Eric, that could be an additional option in the configuration panel, for example:

When the From: address is not recognized:

- Ignore the incoming email
- Bounce the email back to the sender
- Accept the email, and: ** Add the user to the User table *** With user role ** Assign to 'anonymous' [could username] *** Capture From: address and allow moderated adding of user accounts

Come to think of it, this last checkbox may as well be full-time behavior, if that moderated-adding feature does get implemented.

Issue #1142 is a request for CC: field for additional email addresses. I know 'watch'ing an issue is essentially the same, but there's currently no way of adding someone else that I know would be interested. If you implemented the CC: feature, it could be used when the bug reporter is not a registered user. This might be a simpler solution than monkeying around with user tables.

And yes, this could be a vector for spam, so mail moderation would be important if the incoming email address is publicly available.

Carl Nygard wrote:

> Issue #1142 is a request for CC: field for additional email addresses. I know 'watch'ing an issue is essentially the same, but there's currently no way of adding someone else that I know would be interested. If you implemented the CC: feature, it could be used when the bug reporter is not a registered user. This might be a simpler solution than monkeying around with user tables.

Oh, interesting. "Add to CC:" could be another reasonable option.

> And yes, this could be a vector for spam, so mail moderation would be important if the incoming email address is publicly available.

To be more explicit, I was thinking less about the risk of spam harvesters, and more about a spammer actively somehow using it as a vector for "broadcasting their pirate signal" ;)

Randy Harmon wrote:

> To be more explicit, I was thinking less about the risk of spam harvesters, and more about a spammer actively somehow using it as a vector for "broadcasting their pirate signal" ;)

As was I. This is similar to the problems the bloggers face with link farms and spam comments. Given an email address to a redmine site, I could spam all the issues with comments trying to convince JP he needs pharmaceutical "help", or whatever. Hence the moderation requirements for all changes made via incoming email.

Or at least an option to moderate, since I don't necessarily want to be bothered with moderation if the email methods are internal only, or restricted to small set of clients or something similar.

Okay, maybe I went a little wild, but maybe not. The attached mock shows a possible incarnation of the configuration form for this plugin, incorporating the various concerns and options that we've discussed so far.

I expect that this UI should make it pretty easy to configure, and should also serve as a little snippet of built-in documentation as to the functionality. Depending on how much functionality gets implemented, parts of this mock would be deferred until (if/when) those features are done.

One other point: I didn't include a username/password field for the IMAP; it occurred to me that this would likely want to be done on a project-by-project basis, and there's an interaction with the "Assign to Project" option at the bottom. I'm not certain whether a plugin can provide per-project configuration panels; so rather than assume, I figured I'd just put these in as drafts for your comments.

I've checked in *basic* support for creating tickets via email in r1568.

IMHO, fetching emails from an IMAP server and parsing them to create/reply a ticket should be dissociated, as there can be other means to retreive the emails but the parsing logic is the same.

r1568 adresses the second part only (email parsing). For now, a simple rake task lets you read an email from STDIN. Another rake task will be added to fetch emails from an IMAP server and submit them to @MailHandler.receive@.

Looked at your patch. There's no way to set the status, priority, or category. Or the tracker, but there's a TODO comment about that. It might be easy to refactor the code that grabs the value from the options[] or from the text via regex (especially if the option name matches the tag name). I'll work on a patch if you like...

Nice how you can grab the issue info from a reply. But wouldn't that get broke if someone translates the email notifications differently (i.e. removes the []'s that you're looking for in the regex)?

Carl Nygard wrote:

> I'll work on a patch if you like...

Your patch is welcome :-)

> Nice how you can grab the issue info from a reply. But wouldn't that get broke if someone translates the email notifications differently (i.e. removes the []'s that you're looking for in the regex)?

[]'s are hard-coded in source:trunk/app/models/mailer.rb@1483#L45, not in translation files.

But if you have a better solution, feel free to submit it.

Jean-Philippe Lang wrote:

> IMHO, fetching emails from an IMAP server and parsing them to create/reply a ticket should be dissociated, as there can be other means to retreive the emails but the parsing logic is the same.

I agree, but in the interest of time I kept all the logic in the same class because that's how mully had it in the plugin. Ideally it would be great to see the emailer support IMAP, POP, and standard input. Perhaps by subclassing @MailHandler@ or adding modules for the specific server behavior.

I agree with JP that things need to be separated. Even moreso, I really don't understand why this is implemented as a plugin? How is the plugin supposed to poll for pop/imap input? I don't understand why you aren't writing a script (or rake task, as JP had it) that makes it easy for the admin to integrate with their mail server (sendmail, postfix, etc.)? Emails appearing to be processed seems to dictate something that would handle that even, so what am I not getting?

I'm going to guess an answer to my own question... the plugin is for configuration, the rake task is for the cron. If that's true... then you're putting a restriction on how this can be implemented. For example, I'd like to put it as a mail handler in my sendmail configuration, on a different machine. How does it get the config info from the yml file? Whatever you do, I think it should be flexible to allow:

- sendmail/postfix script handler on separate server
- crontab with imap/postfix

PS: Eric, thanks for the credit on github, but why are you pointing to redmine.shaneandpeter.com instead of this issue on redmine.org? And why does it have a certificate for plesk/SWsoft that mismatches? That just looks fishy.

Carl Nygard wrote:

> I agree with JP that things need to be separated. Even moreso, I really don't understand why this is implemented as a plugin?

The *only* reasons I worked on it as a plugin are:

- There was some existing work by mully on it
- I don't have commit rights to the svn repository

> How is the plugin supposed to poll for pop/imap input? I don't understand why you aren't writing a script (or rake task, as JP had it) that makes it easy for the admin to integrate with their mail server (sendmail, postfix, etc.)? Emails appearing to be processed seems to dictate something that would handle that even, so what am I not getting?

The plugin comes with a "Rake task":http://github.com/edavis10/redmine_ticket_emailer/tree/master/lib/tasks/emailer.rake that will check the IMAP server and process all new mail. I outlined why a shell script or piping from STDIN in my "first comment":http://www.redmine.org/issues/show/1110#note-1

> I'm going to guess an answer to my own question... the plugin is for configuration, the rake task is for the cron.

No, the plugin would do both the configuration and provide a rake task to run. Ideally, I would want the plugin to support other input methods like POP or STDIN but they haven't been implemented yet.

> PS: Eric, thanks for the credit on github, but why are you pointing to redmine.shaneandpeter.com instead of this issue on redmine.org?

I"m sorry, it was a typo and has been corrected. Shane and Peter are the people who were sponsoring the work on it.

> And why does it have a certificate for plesk/SWsoft that mismatches? That just looks fishy.

Thats the default certificate for a Plesk server. They haven't had the need to upgrade it yet since it's only used for internal projects right now.

I've started a doc: [[RedmineReceivingEmails]].

We could also add a simple action on some controller so that a separate email server could post raw incoming emails to the application via a HTTP request. Unlike the rake task that read emails from STDIN, this would avoid to reload the environment each time an email is received.

Jean-Philippe Lang wrote:

> We could also add a simple action on some controller so that a separate email server could post raw incoming emails to the application via a HTTP request. Unlike the rake task that read emails from STDIN, this would avoid to reload the environment each time an email is received.

That would be great. Sounds more like a REST API than a mailer.

Jean-Philippe Lang wrote:

> We could also add a simple action on some controller so that a separate email server could post raw incoming emails to the application via a HTTP request. Unlike the rake task that read emails from STDIN, this would avoid to reload the environment each time an email is received.

I like that idea. Configuration-wise, it's not so cumbersome as setting up cron tasks, and the email changes get processed as soon as the email appears.

Carl, this idea was taken from Request Tracker.

It was committed in r1584. Doc is available in [[RedmineReceivingEmails]].

I'm looking at using this new receive email functionality for supporting messages to forums. I'm most interested in the IMAP option and have been looking at the rake task. After looking at the rake task I'm having a hard time understanding how the checks for email create or update issues. For example, the last line of the IMAP rake task is: Redmine::IMAP.check(imap_options, options) This call appears to do the check on the IMAP server but nothing more. I was thinking there would be a call to MailHandler.receive. Am I missing something?

Joshua Hoover wrote:

> Am I missing something?

Yes, you are.
See what @Redmine::IMAP.check@ does: source:/trunk/lib/redmine/imap.rb@1572#L23

Here you'll find the call to @MailHandler.receive@.

Jean-Philippe Lang wrote:

> Joshua Hoover wrote:
>
> > Am I missing something?
>
> Yes, you are.
> See what @Redmine::IMAP.check@ does: source:/trunk/lib/redmine/imap.rb@1572#L23
> Here you'll find the call to @MailHandler.receive@.

To help out anyone who doesn't know how Rails and ActionMailer work the process is like this:

Redmine::IMAP.check connects to IMAP server

Each new message is sent to MailHandler.receive

MailHandler.receive will parse the email and attachments then

MailHandler.receive will use Redmine's MailHandler.dispatch to update the issue

---

I'm marking this as Resolved, I've set it up and it works quite good. New keyword support is pretty easy to add (E.g. #1669) and can be requested in separate issues.

Hello, If redmine would recognize otrs forwards mails it would be beautiful.

Could you add a option config / add to code , so #20081110xxxx is not a redmine ticket ?

The problem is when you forward a ticket from otrs system to redmine.

Thank you

-

Hello, I've done creating issue via email in Readmine. Now I want to update existing issue

via email by using issue number or something else. Do I've to change script in redm-mailhandler.rb file? or any other suggestions regarding this?

Ahammad Fekri wrote:

> Hello, I've done creating issue via email in Readmine. Now I want to update existing issue via email by using issue number or something else. Do I've to change script in redm-mailhandler.rb file? or any other suggestions regarding this?

You can update Notes through mail. You just have to add Redmine ID in subject e.g. [#123]. Regarding updating other fields I am still testing this feature.

I am using this feature, really amazing and saves lot of time.

I have read doc from [[RedmineReceivingEmails.]].

related_issues

relates,Closed,1083,Accept new tickets via email
relates,Closed,444,submit issue via email
relates,Closed,1388,Email integration: track email answers
relates,Closed,1669,Add Status field to the Mail Handler
relates,Closed,13826,application.js: randomKey function optimization

履歴

#1 - 2022/05/10 17:29 - Admin Redmine

- カテゴリ を Email receiving_29 にセット

- 対象バージョン を 0.8_2 にセット