# Redmine - Feature #36711

## Various features for MailHandler

2022-02-26 23:12 - Vincent Danjean

| | | | | |
|---|---|---|---|---|
| **Status:** | New | | **Start date:** | |
| **Priority:** | Normal | | **Due date:** | |
| **Assignee:** | | | **% Done:** | 0% |
| **Category:** | Email receiving | | **Estimated time:** | 0.00 hour |
| **Target version:** | | | | |
| **Resolution:** | | | | |

**Description**

Hi,

For my local redmine instance (4.0.7-1~bpo10+1 from Debian package), I needed to improve the MailHandler behavior.
I'm sharing here my work in case it can be useful to others. Perhaps some modification might be added to official redmine, it is up to you.
I'm new in ruby, so do not hesitate to correct me if they are improvement that can be done.

My patch add the following behaviors:

1) keep some headers in messages
It occurs several time I've been missing some information present in the original mail. So, I added some functions to keep them (to, cc, bcc (and subject for replies))
This is the part about headers_as_text and header_as_text
It is my first try. All is hardcoded in the code (no option)

2) Allows one to specify the start of a line as a regexp when matching redmine keyword
So, invoking redmine:email:receive_imap with reprefix='Redmine:? ?' forces redmine to only look at line starting with this motif (before all keywords such as Project: ...). This avoid false positive line in the mail...
For this, there is a small modif to read the option from env and a small modif in extract_keyword! to add this prefix (and to use named match group as the prefix can add new groups '()')

3) And the biggest part: something that allows to answer to an issue without knowing the issue number.
One of my initial goal was to be able to keep track of icinga problems in Redmine.
Initially, I setup icinga to send notification to redmine. It works, but all messages creates new issues (either for new problems, but also for notification of fixed problem).
So, redmine was having lots of issues, and manual work was required if we want to close issue correctly.
I was difficult to do more things from icinga because icinga runs on a machine that cannot access to the web interface of redmine (no way for me to change this politic), so I cannot imagine icinga to look for the correct issue number (and I would not even know how to do this if the systems was on the same machine).

So, I imagine another idea. I use a specific custom field. Icinga sets up this field in all its mails, making unique by problem.
On redmine part, in the dispatch_to_default, before creating an issue, I first check if
a) the custom field is set
b) and an issue is opened with the custom field set to this value
If both conditions are present, I add a reply to the issue (instead of creating a new issue)

On redmine part, there is just an option to choose the custom field to use (ext_ref in my implementation)
On icinga, I added this kind of things in the message (it is prepared by a bash script):

```
## Redmine links
cat << EOF
Redmine external reference: icinga-service: ${SERVICENAME}@${HOSTNAME}
Redmine machine: ${HOSTNAME}
Redmine service: ${SERVICENAME}
EOF
if [ "$SERVICESTATE" = OK ]; then echo "Redmine Status: Terminé(e)"; fi
```

Machine and Service being to other custom fields in my tracker (and "External Reference" is the name of the custom field I'm using and that I give with the ext_ref='...' new parameter)

So now, new icinga problems creates new tickets. But futher notifications for the same problem (host/service) are added to the same redmine issue, until the problem is gone ("$SERVICESTATE" = OK) that will close the ticket. If another problem occurs with the same host/service, a new ticket will be created. I a separate problem occurs with different host and/or service, another ticket will be used.

A few word about my code:
1) I'm not sure I correctly use IssueQuery. I do not find lots of examples (from ruby point of view) and I'm under the impression that the way I add the custom field filter (from its name and the searched value) is a hack.
2) I'm pretty sure that IssueQuery won't work with no_permission_check=1 (but again, I did not find any doc about the restriction on issues that IssueQuery return (visible to the current user?, ...?))
3) I needed to find a keyword (the custom field value) at the start, before the place where it will be extracted and write into the issue. With the initial implementation of get_keyword/extract_keyword!, the custom field value would be erased at the start, and not seen later when the issue must be updated.
So I added a parameter to these to function that allows one to avoid to modify the message.
It is not a proper code (extract_keyword! can be a method that do **not** modify the text, ...), it would probably need to be rewritten correctly.

And, for the end, an idea that I will perhaps implement (not yet done): two different options for allow_override: one used for new issues, one used when replying to an issue.

Even if these ideas are not added to core Redmine, I think they can be useful for other people. Having looked on Internet, I was not the only one trying to make icinga and redmine work together.
And this framework can (and will in my case) be use in other contexts:
- linking issues from redmine and another ticket system (just an idea for now, but it should be doable)
- allowing an external pipeline to create and update the good issue by mail (based on an external unique pipeline run ID)
- ...

Regards,
Vincent

---

## History

**#1 - 2023-02-15 23:38 - Vincent Danjean**

*- File patch added*


Updated patch for redmine 5.0.4

**#2 - 2023-02-16 22:13 - Vincent Danjean**

*- File patch added*


Updated patch for redmine 5.0.4 (the previous was wrong due to frozen literals by default)

This patch also filter-out small images attachments, usually logos in signature

## Files

| patch | 5.59 KB | 2022-02-26 | Vincent Danjean |
|-------|---------|------------|-----------------|
| patch | 6.4 KB  | 2023-02-15 | Vincent Danjean |
| patch | 6.47 KB | 2023-02-16 | Vincent Danjean |