Redmine - Feature #26791

Send individual notification mails per mail recipient

2017-08-30 12:46 - Holger Just

Status:	Closed	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:	Jean-Philippe Lang	**% Done:**	0%
Category:	Email notifications	**Estimated time:**	0.00 hour
Target version:	4.0.0		
Resolution:	Fixed		

Description

With this patch series, we are introducing the general concept to send multiple mails per notification event. Specifically, we are sending one mail per recipient user. Each of those mails is rendered in the context of the recipient (i.e. with User.current set to the recipient). The mails can be sent immediately or optionally delayed using ActiveJob.

This approach has a couple of advantages over the existing approach of sending a single mail to multiple recipients:

- We ensure object visibility for the rendered mails in the most generic way. Since each mail is rendered with User.current set to the recipient, all content rendered in the mail (including shown issue attributes, macros and links in rendered textile text, ...) is guaranteed to only be visible to the user without having to ensure this manually for each mail using the same visibility rules as used for the HTML views already.
- The mail will always be sent in the recipients language, even if multiple users with different languages are notified
- There is no risk of the whole notification mail being rejected because of a single invalid recipient address (see #8157, #6829, #8733 and a lot of duplicates)

In the long run, it will also allow us to introduce user-specific additional information in the rendered notification, e.g. related issues, links to assign the issue, ...

With this new approach, we do have a some challenges which we are trying to address separately:

- Since each mail is always rendered with an adapted User.current, any mail views which expected to use the original User.current need to be adapted. In Redmine core, this only affects the security notifications.
- Depending on the number of recipients for a notification, there might be many rendered mails which can take some time in the request. To ensure a snappy response, we can use ActiveJob to render and send the mails in the background. This can make the async_* delivery methods obsolete.

Reminder: How mail sending works in plain Rails (and Redmine)

This patch series changes how mails are generated and delivered. Let's first look at how a mail normally gets generated and delivered in ActionMailer.

- mail = Mailer.news_added(news) - The user calls a class-method of the Mailer class (a child-class of ActionMailer::Base). This method is usually not implemented in the Mailer class itself. Instead, there is a method_missing method on ActionMailer::Base. Here, ActionMailer creates a ActionMailer::MessageDelivery object. Which is basically a delegator to an *eventually rendered* mail object. Now, the mail is not actually rendered here. The MessageDelivery object only collects all arguments it needs to eventually render the mail. This happens implicitly when any method on the mail is accessed (e.g. when the mail is about to be delivered). The MessageDelivery delegator is returned to the user.

Now that we have the (delegated) mail, we can deliver it with mail.deliver. This method used to be the only delivery method. With newer Rails versions, there is (with some variations) deliver_now and deliver_later. Redmine usually calls deliver which in source:trunk/config/initializers/10-patches.rb#L160 calls deliver_now currently.

When calling mail.deliver, the delegator starts to render the mail:

- MessageDelivery#processed_mailer creates a new Mailer instance and calls process on it.
- Mailer#process(:news_added, news) - The process method is part of ActionMailer (and in turn part of the ActionController framework on which ActionMailer is based on). Among things, it calls the specified action, i.e. the instance method.
- Mailer#news_added(news) - This instance method is implemented on the Mailer class and is responsible to render the actual mail message. Here, we set headers, instance variables for the view and basically do the usual controller stuff. With a final call to mail, we eventually render a single mail and return a Mail::Message object.
- This returned Mail::Message object is delivered with a call to Mail::Message#deliver_now.

How we hook into this process

Now, in order to send multiple mails, we hook into this process to render (and deliver) multiple mails. We want to ensure two things:

- We want to create multiple mails, one per recipient and deliver them all in one go.
- When rendering the mail (i.e. when calling the Mailer's instance method), we want to set User.current to the recipient user and set it back to the original value after the rendering is done

These two goals are achieved with two specific changes to our Mailer:

We introduce a wrapper class called Mailer::MultiMessage. Instances of this class wrap multiple Mail::Message objects (or rather ActionMailer::MessageDelivery objects which delegate to Mail::Message objects). The class provides the same deliver_* methods as the original Mail::Message objects. When calling such a method, we are calling it on each of the wrapped messages. This allows us to use an instance of this class in place of a ActionMailer::MessageDelivery object.

We use this class by implementing *class methods* for each mailer action. These class methods implement the same external interface (i.e. accept the same arguments) as the old instance methods. This ensures that we keep the existing public interface. The class methods fetch the recipient users for the respective object (i.e. the issue, news, comment, ...). We use Mailer::MultiMessage#for to add an ActionMailer::MessageDelivery object for each recipient which eventually renders the mail.

When creating the delegator, we also pass the recipient user to the arguments for the mail rendering. We use this user in the overwritten Mailer#process method to set User.current and their language before actually rendering the message. This convention allows us to keep the interfaces clear: the user is only passed between the class method and Mailer#process but doesn't need to be specified on the instance action. Here, it is already set as User.current for each rendered mail.

We also extend Mailer.method_missing (on the class) to create a single-mail Mailer::MultiMessage object with the current user in case the class method for an action is not explicitly overwritten. This ensures that our convention of always passing a user to Mailer#process is kept up even if a plugin only adds an instance method (which was valid before).

What we tried and did not work

I first attempted to set the current user directly in the action's class method. and reset it directly afterwards. That would have made the code much more localized. However, due to the delayed rendering of the mail with the ActionMailer::MessageDelivery object, the mail will be rendered only after we have already reset the current user.

One way to work around this would be to introduce deliver_* methods (like the now obsolete Mailer.deliver_issue_add method). However doing so would have changed the intended external interface of our Mailer class and would likely lead to integration pain with all existing Redmine plugins sending mail.

What we ship

We ship three patches here:

- 0001-Send-individual-emails-for-each-mail-recipient.patch - The main implementation including tests.
- 0002-Cleanup-Remove-Issue-each_notification-and-Journal-e.patch - Removes unneccessary methods for grouping recipients based on custom field visibility
- 0003-Optional-Ensure-that-ActiveRecord-Base-objects-are-f.patch - An optional patch which changes the serialization behavior when sending mails with ActiveJob (e.g. with Mailer.news_added(news).deliver_later). By default, Rails serialized the arguments as a globalid (which is basically a reference to the object in the database identified by its ID). If there are concurrent changes to e.g. a newly created issue, this might result in notifications using the new state instead of the original state. With this patch, we serialize the exact attributes of the given objects into the job. Since this also causes nested attributes to be serialized along, care must be taken to avoid reference errors. In general, we should probably try to use mostly immutable state in the database where it is an important consideration that the notification state is exactly shown in the DB.

Future work

ActiveJob sending

If the general direction of this patch is accepted, it is probably desireable to move the mail sending to ActiveJob completely. This could replace the async_* delivery methods currently configurable in configuration.yml. This change can either be hardcoded in source:trunk/config/initializers/10-patches.rb#L160 or made configurable with a setting. I'll be glad to provide a patch for this. It would only be a few lines of code.

By default, we could use the in-process ActiveJob runner to send mails. More advanced users with a dedicated job worker (e.g. delayed job) can easily switch to that without changes to the Redmine code itself.

BCC recipients

Since each user gets their own mail, we can probably remove the setting to send mails with BCC completely. If plugins want to send mails to non-user recipients, we would have to make sure to either set the recipients manually on BCC or group them appropriately. It is not a required setting in Redmine core anymore though.

Related issues:

Related to Redmine - Defect #8157: Redmine do not send notification emails if...	**Closed**	
Related to Redmine - Feature #5990: Email notification should reflect user la...	**Closed**	**2010-07-29**
Related to Redmine - Defect #17096: Issue emails cannot be threaded by some m...	**Closed**	
Related to Redmine - Defect #27242: issue creation failed if email notificati...	**Closed**	
Related to Redmine - Defect #11106: Email notification processing is slow - R...	**Closed**	
Related to Redmine - Defect #8137: HTML email doesn't obey directionality of ...	**Closed**	**2011-04-13**
Related to Redmine - Defect #5703: On SMTP failure, an internal error occurs ...	**Closed**	**2010-06-17**
Related to Redmine - Feature #28338: Make Redmine Email Delivery compatible w...	**Closed**	
Related to Redmine - Defect #16784: Notifications not sent when receiving ema...	**Closed**	
Related to Redmine - Feature #30068: Remove :async_smtp and :async_sendmail d...	**Closed**	
Related to Redmine - Defect #26010: can't create Thread	**Closed**	
Related to Redmine - Defect #30787: Other recipients are not listed in To fie...	**Closed**	
Related to Redmine - Feature #30820: Drop setting "Blind carbon copy recipien...	**Closed**	
Related to Redmine - Feature #32781: [Mail Notifications] (feature back) allo...	**Closed**	
Related to Redmine - Patch #36005: Adopt 2FA emails to new Mailer interface	**Closed**	
Has duplicate Redmine - Feature #4231: Send email notifications one by one	**Closed**	**2009-11-17**
Has duplicate Redmine - Defect #11981: Redmine send mail fail when project ha...	**Closed**	
Has duplicate Redmine - Feature #32291: Need some patch to enable emails queue	**Closed**	
Blocks Redmine - Feature #2496: Per user email format settings (HTML/plain text)	**New**	**2009-01-13**

Associated revisions

Revision 17583 - 2018-10-06 15:08 - Jean-Philippe Lang

Send individual emails for each mail recipient (#26791).

We are creating multipe mails per class notification event, one per recipient, wrapped in a Mailer::MultiMessage object to send them all at once.

We keep the existing interface of all class methods intended to be used by external code the same as they were before, with one exception:

We provide additional recipient addresses in options[:recipients] for Mailer.security_notification. Since the first-class recipients have to be users to render individual mails for them, additional recipient addresses have to be provided with some other channel.

By providing additional recipients in options[:recipients], we can solve the use-case for address change notifications for users, which probably is the only real use-case for having to use a plain email address instead of a User as a notification recipient.

Patch by Holger Just and Marius BALTEANU.

Revision 17584 - 2018-10-06 15:09 - Jean-Philippe Lang

Cleanup: Remove Issue#each_notification and Journal#each_notification (#26791).

Patch by Holger Just.

Revision 17585 - 2018-10-06 15:10 - Jean-Philippe Lang

Ensure that ActiveRecord::Base objects are fully serialized for mail sending (#26791).

Patch by Holger Just.

Revision 17587 - 2018-10-06 18:07 - Jean-Philippe Lang

Fixed that test_email raises an error with #deliver_later (#26791).

Revision 17588 - 2018-10-10 19:13 - Jean-Philippe Lang

Send emails asynchronously (#26791).

Custom async_* delivery methods are removed in favor of ActiveJob (Async by default).

Revision 17590 - 2018-10-10 20:45 - Jean-Philippe Lang

Replaces remaining #deliver with #deliver_later (#26791).

Revision 17591 - 2018-10-10 21:04 - Jean-Philippe Lang

Removes method_missing override (#26791).

Revision 17668 - 2018-12-01 07:45 - Go MAEDA

Use "abort" instead of "exit" in order to make error log more informative (# 26791).

History

#2 - 2017-08-30 16:58 - Jan from Planio www.plan.io

- Related to Defect #8157: Redmine do not send notification emails if a recipients email address is not valid added

#3 - 2017-08-31 04:41 - Go MAEDA

- Related to Defect #11981: Redmine send mail fail when project has too much members added

#4 - 2017-09-03 08:30 - Go MAEDA

- Related to Feature #5990: Email notification should reflect user language setting added

#5 - 2017-09-04 14:04 - Go MAEDA

- Blocks Feature #2496: Per user email format settings (HTML/plain text) added

#6 - 2017-09-08 09:54 - Go MAEDA

- Related to Defect #17096: Issue emails cannot be threaded by some mailers due to inconsistent Message-ID and References field added

#7 - 2017-09-29 06:44 - Toshi MARUYAMA

- Description updated

#8 - 2017-10-19 21:32 - Juan Diego Iannelli

+1

#9 - 2017-10-21 09:26 - Go MAEDA

- Related to Defect #27242: issue creation failed if email notification fail added

#10 - 2017-11-04 06:49 - Go MAEDA

- Target version set to 4.1.0

This patch can fix many annoying behaviors related to email notification (see "Related issues"). Setting target version to 4.1.0.

#11 - 2017-11-04 06:59 - Go MAEDA

- Related to Defect #11106: Email notification processing is slow - Redmine 1.2.2 added

#12 - 2017-11-26 05:28 - Go MAEDA

- Related to Defect #8137: HTML email doesn't obey directionality of locale added

#13 - 2018-03-31 10:07 - Go MAEDA

- Related to Defect #5703: On SMTP failure, an internal error occurs and all changes to an issue are lost added

#14 - 2018-04-03 02:39 - Go MAEDA

- Related to Feature #28338: Make Redmine Email Delivery compatible with ActiveJob added

#15 - 2018-04-24 09:59 - Go MAEDA

- Target version changed from 4.1.0 to 4.0.0

#16 - 2018-05-22 14:21 - Enziin System

After applying the patch and edit: /config/initializers/10-patches.rb

```
module ActionMailer
  class MessageDelivery < Delegator
    def deliver
      deliver_later
    end
  end
end
```

I get an error when updating an issue.

```
Started PATCH "/issues/2" for 127.0.0.1 at 2018-05-23 04:38:23 +0000
Processing by IssuesController#update as HTML
  Parameters: {"utf8"=>"□",...}
...

[ActiveJob] [ActionMailer::DeliveryJob] [a5d3c247-2e7a-4879-aaad-d0590de9026b] Performing ActionMailer::Delive
ryJob (Job ID: a5d3c247-2e7a-4879-aaad-d0590de9026b) from Async(mailers) with arguments: "Mailer", "issue_edit
", "deliver_now", #<GlobalID:0x00007fa2d053b888 @uri=#<URI::GID gid://redmine-app/User/5>>, #<GlobalID:0x00007
fa2d053af28 @uri=#<URI::GID gid://redmine-app/Journal/8>>

[ActiveJob] [ActionMailer::DeliveryJob] [a5d3c247-2e7a-4879-aaad-d0590de9026b] Error performing ActionMailer::
DeliveryJob (Job ID: a5d3c247-2e7a-4879-aaad-d0590de9026b) from Async(mailers) in 0.15ms: ArgumentError (wrong
 number of arguments (given 2, expected 1)):

.../app/models/mailer.rb:246:in `issue_edit'
```

#17 - 2018-09-10 23:12 - Ken Zalewski

Is there a plan to get this patch merged into the core code base?

#18 - 2018-09-11 08:18 - Bernhard Rohloff

Ken Zalewski wrote:

> Is there a plan to get this patch merged into the core code base?

It's on the roadmap for 4.0.0 so I think there is a plan. ;-)

#19 - 2018-09-15 09:49 - Go MAEDA

- Has duplicate Feature #5821: Respond to unprocessed emails added

#20 - 2018-09-15 09:50 - Go MAEDA

- Has duplicate deleted (Feature #5821: Respond to unprocessed emails)

#21 - 2018-09-15 09:50 - Go MAEDA

- Has duplicate Feature #4231: Send email notifications one by one added

#22 - 2018-09-29 09:55 - Jean-Philippe Lang

Because of r17269, the patch does not apply cleanly. Could you update it so we can commit this change? Thanks.

#23 - 2018-09-29 23:46 - Marius BĂLTEANU

- File 0001-Send-individual-emails-for-each-mail-recipient.patch added

I took the liberty to update the first patch to apply cleanly on the current trunk just in case of Holger Just or some else from Plan.io team do not have the time to do it until the new release. All tests pass and the rest of the patches apply cleanly.

#24 - 2018-09-30 11:13 - Jean-Philippe Lang

Thank you Marius, but still I get many conflicts when trying to apply your patch on current trunk.

#25 - 2018-09-30 11:14 - Marius BĂLTEANU

Jean-Philippe Lang wrote:

> Thank you Marius, but still I get many conflicts when trying to apply your patch on current trunk.

Let me take a look, maybe I missed something.

#26 - 2018-09-30 11:22 - Marius BĂLTEANU

- File deleted (0001-Send-individual-emails-for-each-mail-recipient.patch)

#27 - 2018-09-30 11:24 - Marius BĂLTEANU

- File 0001-Send-individual-emails-for-each-mail-recipient_v2.patch added

My last shot.

```
vagrant@jessie:/vagrant/project/redmine$ git log -1
commit 6e4deb6c10668b5432fc6990fa7e08a22208c18f
Author: Jean-Philippe Lang <jp_lang@yahoo.fr>
Date:   Sun Sep 30 09:10:10 2018 +0000

    Reverts unwanted change.

    git-svn-id: http://svn.redmine.org/redmine/trunk@17537 e93f8b46-1217-0410-a6f0-8f06a7374b81

vagrant@jessie:/vagrant/project/redmine$ patch -p1 < 0001-Send-individual-emails-for-each-mail-recipient_v2.pa
tch
patching file app/models/email_address.rb
patching file app/models/mailer.rb
patching file app/views/mailer/security_notification.html.erb
patching file app/views/mailer/security_notification.text.erb
patching file app/views/mailer/settings_updated.html.erb
patching file app/views/mailer/settings_updated.text.erb
patching file test/functional/documents_controller_test.rb
patching file test/functional/issues_controller_test.rb
patching file test/functional/issues_custom_fields_visibility_test.rb
patching file test/functional/messages_controller_test.rb
patching file test/functional/news_controller_test.rb
patching file test/unit/changeset_test.rb
patching file test/unit/comment_test.rb
patching file test/unit/document_test.rb
patching file test/unit/issue_test.rb
patching file test/unit/journal_observer_test.rb
patching file test/unit/journal_test.rb
patching file test/unit/mail_handler_test.rb
patching file test/unit/mailer_test.rb
patching file test/unit/news_test.rb
patching file test/unit/repository_test.rb
patching file test/unit/wiki_content_test.rb
vagrant@jessie:/vagrant/project/redmine$
```

#28 - 2018-10-01 17:25 - Marius BĂLTEANU

- Assignee set to Holger Just

Holger, can you take a look to my updated patch or update your first patch to apply cleanly?

#29 - 2018-10-01 21:35 - Jean-Philippe Lang

Thanks Marius, your patch applies cleanly but I get a few test failures: IP address is blank in security notifications.

#30 - 2018-10-01 22:54 - Marius BĂLTEANU

Jean-Philippe Lang wrote:

> Thanks Marius, your patch applies cleanly but I get a few test failures: IP address is blank in security notifications.

Sorry Jean-Philippe, I'm not sure how to fix the failing test and I need more time to investigate the problem by myself. I propose to postpone this ticket to 4.1.0 if Holger can't update the patch in a proper time. IMHO, I consider the release of 4.0.0 top priority.

#31 - 2018-10-02 00:34 - Marius BĂLTEANU

I looked again in the code and a fix that makes sense for me is the following one:

```
diff --git a/app/models/mailer.rb b/app/models/mailer.rb
index 717b000..fad1939 100644
--- a/app/models/mailer.rb
+++ b/app/models/mailer.rb
@@ -604,7 +604,7 @@ class Mailer < ActionMailer::Base
       value: options[:value]
     )
     @title = options[:title] && l(options[:title])
-    @originator = options[:originator] || User.current
+    @originator = options[:originator] || sender
     @remote_ip = options[:remote_ip] || @originator.remote_ip
     @url = options[:url] && (options[:url].is_a?(Hash) ? url_for(options[:url]) : options[:url])
     redmine_headers 'Sender' => @originator.login
```

because sender is received from method self.security_notification where is already defined as User.current.
Because of this, I think is safe also to remove the line 602 @user = User.current from method security_notification.

```
diff --git a/app/models/mailer.rb b/app/models/mailer.rb
index fad1939ce..0e3f01483 100644
--- a/app/models/mailer.rb
+++ b/app/models/mailer.rb
@@ -598,7 +598,6 @@ class Mailer < ActionMailer::Base
   def security_notification(sender, options={})
     @sender = sender
     redmine_headers 'Sender' => sender.login
-    @user = User.current
     @message = l(options[:message],
       field: (options[:field] && l(options[:field])),
       value: options[:value]
```

The test MailerTest#test_security_notification (IP address is blank in security notifications.) passes now and the results for the entire tests suite look good on my environment (with the updated patch and the above two proposed fixes applied):

View test resultsView test results

```
Run options: --seed 34100

# Running:

..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
..........................................................................................................
.....................................................................F

Failure:
CalendarsControllerTest#test_show [/work/test/functional/calendars_controller_test.rb:56]:
Expected at least 1 element matching "img[class="gravatar"]", found 0..
Expected 0 to be >= 1.

bin/rails test test/functional/calendars_controller_test.rb:35

Error:
GanttsControllerTest#test_gantt_should_export_to_png:
Magick::ImageMagickError: non-conforming drawing primitive definition `text' @ error/draw.c/DrawImage/3265
    lib/redmine/helpers/gantt.rb:439:in `draw'
    lib/redmine/helpers/gantt.rb:439:in `to_image'
    app/controllers/gantts_controller.rb:42:in `block (2 levels) in show'
    app/controllers/gantts_controller.rb:40:in `show'
    lib/redmine/sudo_mode.rb:63:in `sudo_mode'
    test/functional/gantts_controller_test.rb:153:in `test_gantt_should_export_to_png'

bin/rails test test/functional/gantts_controller_test.rb:152

Finished in 451.992636s, 9.5909 runs/s, 63.0984 assertions/s.
```

```
4335 runs, 28520 assertions, 1 failures, 1 errors, 2 skips

You have skipped tests. Run with --verbose for details.
```

where:

1. CalendarsControllerTest#test_show [/work/test/functional/calendars_controller_test.rb:56] fails also on the current trunk (https://redmine.org/builds/logs/build_trunk_mysql_ruby-2.5_384.html)
2. "GanttsControllerTest#test_gantt_should_export_to_png" is caused by an issue on my local environment.

#32 - 2018-10-02 17:31 - Marius BĂLTEANU

Marius BALTEANU wrote:

> [...]
> where:
>
> 1. CalendarsControllerTest#test_show [/work/test/functional/calendars_controller_test.rb:56] fails also on the current trunk (https://redmine.org/builds/logs/build_trunk_mysql_ruby-2.5_384.html)

I think that was an external problem because now the test passes on redmine.org and on my environment as well.

#33 - 2018-10-02 21:34 - Holger Just

Thank you all for working on the patch and considering its inclusion!

I'll try to get to this tomorrow to fix any remaining test failures. Unfortunately, it's been a crazy week so far and I hadn't had much time so far...

#34 - 2018-10-03 05:14 - Go MAEDA

- Related to Feature #12239: Translateable Notifications added

#35 - 2018-10-06 18:14 - Jean-Philippe Lang

Holger Just wrote:

> If the general direction of this patch is accepted, it is probably desireable to move the mail sending to ActiveJob completely. This could replace the async_* delivery methods currently configurable in configuration.yml. This change can either be hardcoded in source:trunk/config/initializers/10-patches.rb#L160 or made configurable with a setting. I'll be glad to provide a patch for this. It would only be a few lines of code.

Yes, we should now use to ActiveJob completely. I had to revert patch 0003 because it was raising an error when trying to send the test_email with #deliver_later (see the test added in r17587), something wrong with the serialization:

```
Error:
MailerTest#test_test_later:
ArgumentError: First argument has to be a user, was "--- !ruby/object:User\nconcise_attributes:\n-
!ruby/object:ActiveModel::Attribute::FromDatabase\n  name: id\n  value_before_type_cast: 1\n-
!ruby/object:ActiveModel::Attribute::FromDatabase\n  name: login\n  value_before_type_cast: admin\n-
!ruby/object:ActiveModel::Attribute::FromDatabase\n  name: hashed_password\n  value_before_type_cast:
b5b6ff9543bf1387374cdfa27a54c96d236a7150\n- !ruby/object:ActiveModel::Attribute::FromDatabase\n  name: firstna
me\n
value_before_type_cast: Redmine\n- !ruby/object:ActiveModel::Attribute::FromDatabase\n
...
```

#36 - 2018-10-07 20:23 - Jean-Philippe Lang

holger mareck, have you an implementation of MultiMessage that is compatible with Async delivery?

Current trunk seems to be far from being compatible. I've spent a few hours on it and it just seems to be broken. For example, I tried to send settings_updated notification asynchronously (see patch below), and only the current user is getting an email. Mailer.settings_updated is called the first time with the array of admins, and then when the "real" notification is processed in the background, it's called with the current user as the first argument.

Please let me know if you have worked on it before I go on.

```
Index: app/models/mailer.rb
===================================================================
--- app/models/mailer.rb    (revision 17587)
+++ app/models/mailer.rb    (working copy)
@@ -657,7 +657,7 @@
     return unless changes.present?
```

```
     users = User.active.where(admin: true).to_a
-    settings_updated(users, changes).deliver
+    settings_updated(users, changes).deliver_later
    end

    # Build a Mail::Message object with a test email for the current user
Index: app/models/setting.rb
===================================================================
--- app/models/setting.rb    (revision 17581)
+++ app/models/setting.rb    (working copy)
@@ -131,7 +131,7 @@
       previous_value = Setting[name]
       set_from_params name, value
       if available_settings[name.to_s]['security_notifications'] && Setting[name] != previous_value
-        changes << name
+        changes << name.to_s
       end
     end
     if changes.any?
```

#37 - 2018-10-09 22:00 - Jean-Philippe Lang

> One way to work around this would be to introduce deliver_* methods (like the now obsolete Mailer.deliver_issue_add method). However doing so would have changed the intended external interface of our Mailer class and would likely lead to integration pain with all existing Redmine plugins sending mail.

I understand that it would change the external interface of the Mailer but I think it's a better way to go than the trickier proposed solution. Maybe we should revert all the changes done here and postpone this feature in order not to delay 4.0 release.

#38 - 2018-10-10 09:31 - Marius BĂLTEANU

Jean-Philippe Lang wrote:

> > One way to work around this would be to introduce deliver_* methods (like the now obsolete Mailer.deliver_issue_add method). However doing so would have changed the intended external interface of our Mailer class and would likely lead to integration pain with all existing Redmine plugins sending mail.
>
> I understand that it would change the external interface of the Mailer but I think it's a better way to go than the trickier proposed solution. Maybe we should revert all the changes done here and postpone this feature in order not to delay 4.0 release.

In my opinion, the safest option now (from many reasons) is to revert all the changes done and postpone this feature for 4.1.0.

#39 - 2018-10-10 20:01 - Holger Just

Sorry for the long silence from my part. It was rather busy at work recently and I had no further energy to work on this issue during the evenings. With that being said, thank you for working on this patch Marius and Jean-Philippe. I sincerely appreciate it.

In general, I tried to keep the external interface of the mailer (i.e. the interface used by all the rest of Redmine's code) as stable as possible. This resulted in some complications which I tried to solve with the MultiMailer. With my initial tests, I checked that it the MultiMailer compatible with redmine's existing async-* delivery methods. I had not initially checked whether this works with ActiveJob. I do appreciate your work on moving over to it since it will help reduce special homegrown code in favor of upstream-maintained mechanisms.

Personally, I'm very fine with adding the deliver_* methods to the mailer. Since the methods signatures of the previously used Mailer methods change (because if the added user parameter), it should be apparent when some plugin is not yet upgraded since it will result in a load ArgumentError instead of silently swallowing notifications. Using the deliver_* class methods makes the whole implementation of the mailer much simpler.

Some comments to your last patch, Jean-Philippe:

- Do we still need Mailer#method_missing? Since we are reverting on the deprecation, we should probably get rid of the fallback to avoid confusion.
- On some places in the Mailer class, you still directly call deliver. This seems to still work as an alias to deliver_now but I have no idea why since you removed the patch in config/initializers/10-patches.rb. In any case, shouldn't the calls also be to deliver_later?

#40 - 2018-10-10 20:24 - Holger Just

As for my third patch to adapt the serialization of the references objects for ActiveJob, this was more or less a proof-of concept to ensure that the full state of the notified object is encoded into the job. This would have caused side-effects if you are not very careful since it could also serialize nested ActiveRecord objects which might cause strange consistency issues if they differ from the state in the database.

I have attempted to test this when I wrote the patch and found no further issues, but I did not primarily test it to use ActiveJob's deliver_later function yet. This was something I intended to further investigate once the general idea of the patch was accepted.

In general, I think it is desirable to not mess with the serialization at all and not use the third patch (which you have now reverted in r17587 already). Instead, we might try to make the objects used for rendering the mails (mostly) immutable over time. For issues an option could be to use "representer" objects instead of actual issues. These would be Structs (or similar) which could be serialized into the job and contain all necessary information to render the mail without having to rely in the current state of the issue object in the database.

What do you think?

#41 - 2018-10-10 21:03 - Jean-Philippe Lang

Marius BALTEANU wrote:

> In my opinion, the safest option now (from many reasons) is to revert all the changes done and postpone this feature for 4.1.0.

That was the idea but I had some time to work on this and I finally decided to commit the changes.

Holger Just wrote:

- Do we still need Mailer#method_missing? Since we are reverting on the deprecation, we should probably get rid of the fallback to avoid confusion.
- On some places in the Mailer class, you still directly call deliver. This seems to still work as an alias to deliver_now but I have no idea why since you removed the patch in config/initializers/10-patches.rb. In any case, shouldn't the calls also be to deliver_later?

Thanks, this is fixed.

> In general, I think it is desirable to not mess with the serialization at all and not use the third patch (which you have now reverted in r17587 already). Instead, we might try to make the objects used for rendering the mails (mostly) immutable over time. For issues an option could be to use "representer" objects instead of actual issues. These would be Structs (or similar) which could be serialized into the job and contain all necessary information to render the mail without having to rely in the current state of the issue object in the database.

That would be nice but we don't just need the issue and its attributes. Rendering the emails relies on many associations like attachments, custom fields. I'm afraid this solution would lead to different kind of problems. I'd happy to get some feedback anyway if you want to give it a try. Thanks.

#42 - 2018-10-11 03:50 - Go MAEDA

- Related to Defect #16784: Notifications not sent when receiving emails with rake task and async deliveries added

#43 - 2018-11-29 17:40 - Jean-Philippe Lang

- Tracker changed from Patch to Feature

- Status changed from New to Closed

- Assignee changed from Holger Just to Jean-Philippe Lang

- Resolution set to Fixed

#44 - 2018-11-30 11:21 - Go MAEDA

- Related to deleted (Feature #12239: Translateable Notifications)

#45 - 2018-11-30 14:05 - Marius BĂLTEANU

- Related to deleted (Defect #11981: Redmine send mail fail when project has too much members)

#46 - 2018-11-30 14:05 - Marius BĂLTEANU

- Has duplicate Defect #11981: Redmine send mail fail when project has too much members added

#47 - 2018-12-01 08:17 - Go MAEDA

- Related to Feature #30068: Remove :async_smtp and :async_sendmail delivery methods added

#48 - 2019-02-03 19:30 - Marius BĂLTEANU

- Related to Defect #26010: can't create Thread added

#49 - 2019-02-11 16:05 - Go MAEDA

- Related to Defect #30787: Other recipients are not listed in To field even if Setting.bcc_recipients is false added

#50 - 2019-02-17 21:00 - Marius BĂLTEANU

- Related to Feature #30820: Drop setting "Blind carbon copy recipients (bcc)" added

#51 - 2019-02-17 21:05 - Marius BĂLTEANU

Holger Just wrote:

> ### BCC recipients
>
> Since each user gets their own mail, we can probably remove the setting to send mails with BCC completely. If plugins want to send mails to non-user recipients, we would have to make sure to either set the recipients manually on BCC or group them appropriately. It is not a required setting in Redmine core anymore though.

I've attached a patch to #30820 to drop the bcc setting.

#52 - 2019-10-17 12:33 - Marius BĂLTEANU

- Has duplicate Feature #32291: Need some patch to enable emails queue added

#53 - 2020-01-12 19:56 - Marius BĂLTEANU

- Related to Feature #32781: [Mail Notifications] (feature back) allow mails to be sent to multiple recipients at once added

#54 - 2021-11-13 10:22 - Go MAEDA

- Related to Patch #36005: Adopt 2FA emails to new Mailer interface added

Files

0002-Cleanup-Remove-Issue-each_notification-and-Journal-e.patch	1.87 KB	2017-08-30	Holger Just
0001-Send-individual-emails-for-each-mail-recipient.patch	53.1 KB	2017-08-30	Holger Just
0003-Optional-Ensure-that-ActiveRecord-Base-objects-are-f.patch	1.88 KB	2017-08-30	Holger Just
0001-Send-individual-emails-for-each-mail-recipient_v2.patch	76.2 KB	2018-09-30	Marius BĂLTEANU