ChiliProject - Bug # 736: Adding users with a dash "-" in email address is broken sometimes

Status:	Closed	**Priority:**	Normal
Author:	Chris Serio	**Category:**	Mail Sending
Created:	2011-11-27	**Assignee:**	Felix Schäfer
Updated:	2011-11-30	**Due date:**	

Remote issue URL:

Affected version: 2.4.0

Description: I have ChiliProject 2.4.0 installed on an Ubuntu server with Apache2. I'm using Ruby 1.9.2. When i create a user with an email address with a dash such as test@my-domain.com I get a 500 internal error with this stack:

Notice the "y-d".

```
<pre>
ArgumentError (invalid range "y-d" in string transliteration):
app/models/mailer.rb:399:in `delete'
app/models/mailer.rb:399:in `create_mail'
app/controllers/users_controller.rb:106:in `create'
<internal:prelude>:10:in `synchronize'
passenger (3.0.9) lib/phusion_passenger/rack/request_handler.rb:96:in `process_request'
passenger (3.0.9) lib/phusion_passenger/abstract_request_handler.rb:513:in
`accept_and_process_next_request'
passenger (3.0.9) lib/phusion_passenger/abstract_request_handler.rb:274:in `main_loop'
passenger (3.0.9) lib/phusion_passenger/classic_rails/application_spawner.rb:321:in `start_request_handler'
passenger (3.0.9) lib/phusion_passenger/classic_rails/application_spawner.rb:275:in `block in
handle_spawn_application'
passenger (3.0.9) lib/phusion_passenger/utils.rb:479:in `safe_fork'
passenger (3.0.9) lib/phusion_passenger/classic_rails/application_spawner.rb:270:in
`handle_spawn_application'
passenger (3.0.9) lib/phusion_passenger/abstract_server.rb:357:in `server_main_loop'
passenger (3.0.9) lib/phusion_passenger/abstract_server.rb:206:in `start_synchronously'
passenger (3.0.9) lib/phusion_passenger/abstract_server.rb:180:in `start'
passenger (3.0.9) lib/phusion_passenger/classic_rails/application_spawner.rb:149:in `start'
passenger (3.0.9) lib/phusion_passenger/spawn_manager.rb:219:in `block (2 levels) in
spawn_rails_application'
passenger (3.0.9) lib/phusion_passenger/abstract_server_collection.rb:132:in `lookup_or_add'
passenger (3.0.9) lib/phusion_passenger/spawn_manager.rb:214:in `block in spawn_rails_application'
passenger (3.0.9) lib/phusion_passenger/abstract_server_collection.rb:82:in `block in synchronize'
<internal:prelude>:10:in `synchronize'
passenger (3.0.9) lib/phusion_passenger/abstract_server_collection.rb:79:in `synchronize'
passenger (3.0.9) lib/phusion_passenger/spawn_manager.rb:213:in `spawn_rails_application'
passenger (3.0.9) lib/phusion_passenger/spawn_manager.rb:132:in `spawn_application'
passenger (3.0.9) lib/phusion_passenger/spawn_manager.rb:275:in `handle_spawn_application'
passenger (3.0.9) lib/phusion_passenger/abstract_server.rb:357:in `server_main_loop'
passenger (3.0.9) lib/phusion_passenger/abstract_server.rb:206:in `start_synchronously'
passenger (3.0.9) helper-scripts/passenger-spawn-server:99:in `<main>'
</pre>
```

Associated revisions

2011-11-30 06:04 pm - Felix SchÃ¤fer

[#736] force recipients/cc to arrays in the Mailer

String#delete might break on 1.9 with dashes in the author's email address. Furthermore, String#delete doesn't do what the original author thought it does.

History

2011-11-27 06:54 pm - Chris Serio

I fixed it locally but my ruby knowledge is terrible so someone else should definitely look into this..

Go to app/models/mailer.rb around line 399. You'll see the code below.

Look at the variable "recipients". Sometimes recipients is an array and then calling "delete" on the array removes any elements that contain the email address that is marked as "don't self notify". This works fine...HOWEVER, sometimes recipients is NOT an array and just contains a single string. In this case, the "delete" method being called is NOT on an array object but on a STRING object which interprets a dash as a range of values and that's why it crashes.

```
<pre>
  # Overrides the create_mail method
  def create_mail
    # Removes the current user from the recipients and cc
    # if he doesn't want to receive notifications about what he does
    @author ||= User.current
    if @author.pref[:no_self_notified]
      recipients.delete(@author.mail) if recipients
      cc.delete(@author.mail) if cc
    end

    notified_users = [recipients, cc].flatten.compact.uniq
    # Rails would log recipients only, not cc and bcc
    mylogger.info "Sending email notification to: #{notified_users.join(', ')}" if mylogger

    # Blind carbon copy recipients
    if Setting.bcc_recipients?
      bcc(notified_users)
      recipients []
      cc []
    end
    super
  end
</pre>
```

2011-11-27 06:55 pm - Felix Schäfer

Thanks, looking at the best way to handle this, if you're lucky it might make it into 2.5.0 :-)

2011-11-27 07:03 pm - Felix Schäfer

Mmh, I've tried the following:

```
<pre><code class="diff">diff --git a/app/models/mailer.rb b/app/models/mailer.rb
index 2be9de7..7a004c4 100644
--- a/app/models/mailer.rb
+++ b/app/models/mailer.rb
@@ -396,8 +396,8 @@ class Mailer < ActionMailer::Base
     # if he doesn't want to receive notifications about what he does
     @author ||= User.current
     if @author.pref[:no_self_notified]
-      recipients.delete(@author.mail) if recipients
-      cc.delete(@author.mail) if cc
+      recipients(Array.new(recipients).delete(@author.mail)) if recipients
+      cc(Array.new(cc).delete(@author.mail)) if cc
     end

     notified_users = [recipients, cc].flatten.compact.uniq</code></pre>
```

But there's some failing tests with that, and I don't have time to look today anymore. Care to post your solution so I can have a look? Thanks.

2011-11-27 07:10 pm - Chris Serio

I'm a C++ dev so I know nothing about Ruby but here's what I did...There are two parts, first, I test if recipients/cc are arrays. If they are, I call delete, otherwise I call gsub. The second part of the fix is in the BCC block of code where I just test "if recipients" and "if cc" before nuking the arrays (if that's what the empty [] brackets even do).

```
<pre>
  # Overrides the create_mail method
  def create_mail
    # Removes the current user from the recipients and cc
    # if he doesn't want to receive notifications about what he does
    @author ||= User.current
    if @author.pref[:no_self_notified]
        if recipients.kind_of?(Array)
            recipients.delete(@author.mail) if recipients
        else
            recipients = recipients.gsub(@author.mail, '') if recipients
        end
        if cc.kind_of?(Array)
            cc.delete(@author.mail) if cc
        else
            cc = cc.gsub(@author.mail, '') if cc
        end
    end

    notified_users = [recipients, cc].flatten.compact.uniq
    # Rails would log recipients only, not cc and bcc
    mylogger.info "Sending email notification to: #{notified_users.join(', ')}" if mylogger

    # Blind carbon copy recipients
    if Setting.bcc_recipients?
      bcc(notified_users)
      recipients [] if recipients
      cc [] if cc
    end
    super
  end
</pre>
```

2011-11-28 06:55 am - Felix Schäfer

- Target version set to 2.5.0

- (deleted custom field) set to 2.4.0

- Status changed from Open to Ready for review

Chris Serio wrote:
> I'm a C++ dev so I know nothing about Ruby but here's what I did...There are two parts, first, I test if recipients/cc are arrays. If they are, I call delete, otherwise I call gsub.

That would probably work, the problem with your solution is that you call @recipients =@ and @cc =@, in that context, the @recipients@ and @cc@ you want to call are methods, i.e. something like @cc(cc.gsub(@author.mail, '')) if cc@ would have worked. (and yes, you can have a method and a variable with the same name in the same scopeâ€¦)

> The second part of the fix is in the BCC block of code where I just test "if recipients" and "if cc" before nuking the arrays (if that's what the empty [] brackets even do).

@recipients []@ resets the recipients to an empty array indeed, but no need to test if there are recipients or not.

Anyway, my proposed solution is to force the recipients and the cc to an array before removing the author email. The tests pass on 1.8.7 and 1.9.2 with the following, could you try it on your system?

```
<pre><code class="diff">
diff --git a/app/models/mailer.rb b/app/models/mailer.rb
index 2be9de7..cfeab4e 100644
--- a/app/models/mailer.rb
+++ b/app/models/mailer.rb
@@ -396,8 +396,8 @@ class Mailer < ActionMailer::Base
     # if he doesn't want to receive notifications about what he does
     @author ||= User.current
     if @author.pref[:no_self_notified]
-      recipients.delete(@author.mail) if recipients
-      cc.delete(@author.mail) if cc
+      recipients((recipients.is_a?(Array) ? recipients : [recipients]) - [@author.mail]) if recipients
+      cc((cc.is_a?(Array) ? cc : [cc]) - [@author.mail]) if cc
     end

     notified_users = [recipients, cc].flatten.compact.uniq
</code></pre>
```

2011-11-29 10:45 am - Chris Serio

Felix SchÃ¤fer wrote:
> Chris Serio wrote:
> > The second part of the fix is in the BCC block of code where I just test "if recipients" and "if cc" before nuking the arrays (if that's what the empty [] brackets even do).
>
> @recipients []@ resets the recipients to an empty array indeed, but no need to test if there are recipients or not.
>
> Anyway, my proposed solution is to force the recipients and the cc to an array before removing the author email. The tests pass on 1.8.7 and 1.9.2 with the following, could you try it on your system?
>
> [...]

Hi Felix, I have not had time to test this on my system yet but in my fix, I definitely needed to test whether there were recipients/cc or not because it crashed when I didn't. I kept getting "NoMethodError (undefined method `[]' for nil:NilClass)". Perhaps this is caused by my fix doing something wrong that yours does not?

2011-11-30 05:04 pm - Felix Schäfer

- Status changed from Ready for review to Closed

Pushed in commit:f333f43, thanks for the report and your help :-)