Email Notification of Ticket Changes

Error: Macro TracGuideToc(None) failed

```
'NoneType' object has no attribute 'find'
```

Trac supports notification of ticket changes via email.

Email notification is useful to keep users up-to-date on tickets/issues of interest, and also provides a convenient way to post all ticket changes to a dedicated mailing list. For example, this is how the ?Trac-tickets mailing list is set up.

Disabled by default, notification can be activated and configured in trac.ini.

Receiving Notification Mails

When reporting a new ticket or adding a comment, enter a valid email address or your username in the *reporter*, *assigned to/owner* or *cc* field. Trac will automatically send you an email when changes are made to the ticket (depending on how notification is configured).

This is useful to keep up-to-date on an issue or enhancement request that interests you.

How to use your username to receive notification mails

To receive notification mails, you can either enter a full email address or your username. To get notified with a simple username or login, you need to specify a valid email address in the *Preferences* page.

Alternatively, a default domain name (**`smtp_default_domain`**) can be set in the TracIni file (see Configuration Options below). In this case, the default domain will be appended to the username, which can be useful for an "Intranet" kind of installation.

Configuring SMTP Notification

Important: For TracNotification to work correctly, the `[trac] base_url` option must be set in trac.ini.

Configuration Options

These are the available options for the `[notification]` section in trac.ini.

- **`smtp_enabled`**: Enable email notification.
- **`smtp_from`**: Email address to use for *Sender*-headers in notification emails.
- **`smtp_from_name`**: Sender name to use for *Sender*-headers in notification emails.
- **`smtp_replyto`**: Email address to use for *Reply-To*-headers in notification emails.
- **`smtp_default_domain`**: (*since 0.10*) Append the specified domain to addresses that do not contain one. Fully qualified addresses are not modified. The default domain is appended to all username/login for which an email address cannot be found from the user settings.
- **`smtp_always_cc`**: List of email addresses to always send notifications to. *Typically used to post ticket changes to a dedicated mailing list.*
- **`smtp_always_bcc`**: (*since 0.10*) List of email addresses to always send notifications to, but keeps addresses not visible from other recipients of the notification email

- **`smtp_subject_prefix`**: (*since 0.10.1*) Text that is inserted before the subject of the email. Set to "__default__" by default.
- **`always_notify_reporter`**: Always send notifications to any address in the reporter field (default: false).
- **`always_notify_owner`**: (*since 0.9*) Always send notifications to the address in the owner field (default: false).
- **`always_notify_updater`**: (*since 0.10*) Always send a notification to the updater of a ticket (default: true).
- **`use_public_cc`**: (*since 0.10*) Addresses in To: (owner, reporter) and Cc: lists are visible by all recipients (default is *Bcc:* - hidden copy).
- **`use_short_addr`**: (*since 0.10*) Enable delivery of notifications to addresses that do not contain a domain (i.e. do not end with @<*domain.com*>).This option is useful for intranets, where the SMTP server can handle local addresses and map the username/login to a local mailbox. See also `smtp_default_domain`. Do not use this option with a public SMTP server.
- **`mime_encoding`**: (*since 0.10*) This option allows selecting the MIME encoding scheme. Supported values:
  - `none`: default value, uses 7bit encoding if the text is plain ASCII, or 8bit otherwise.
  - `base64`: works with any kind of content. May cause some issues with touchy anti-spam/anti-virus engines.
  - `qp` or `quoted-printable`: best for european languages (more compact than base64) if 8bit encoding cannot be used.
- **`ticket_subject_template`**: (*since 0.11*) A ?Genshi text template snippet used to get the notification subject.
- **`email_sender`**: (*since 0.12*) Name of the component implementing `IEmailSender`. This component is used by the notification system to send emails. Trac currently provides the following components:
  - `SmtpEmailSender`: connects to an SMTP server (default).
  - `SendmailEmailSender`: runs a `sendmail`-compatible executable.

Either **`smtp_from`** or **`smtp_replyto`** (or both) *must* be set, otherwise Trac refuses to send notification mails.

The following options are specific to email delivery through SMTP.

- **`smtp_server`**: SMTP server used for notification messages.
- **`smtp_port`**: (*since 0.9*) Port used to contact the SMTP server.
- **`smtp_user`**: (*since 0.9*) User name for authentication SMTP account.
- **`smtp_password`**: (*since 0.9*) Password for authentication SMTP account.
- **`use_tls`**: (*since 0.10*) Toggle to send notifications via a SMTP server using ?TLS, such as GMail.

The following option is specific to email delivery through a `sendmail`-compatible executable.

- **`sendmail_path`**: (*since 0.12*) Path to the sendmail executable. The sendmail program must accept the `-i` and `-f` options.

Example Configuration (SMTP)

```
[notification]
smtp_enabled = true
smtp_server = mail.example.com
smtp_from = notifier@example.com
smtp_replyto = myproj@projects.example.com
smtp_always_cc = ticketmaster@example.com, theboss+myproj@example.com
```

Example Configuration (`sendmail`)

```
[notification]
smtp_enabled = true
email_sender = SendmailEmailSender
sendmail_path = /usr/sbin/sendmail
smtp_from = notifier@example.com
smtp_replyto = myproj@projects.example.com
smtp_always_cc = ticketmaster@example.com, theboss+myproj@example.com
```

Customizing the e-mail subject

The e-mail subject can be customized with the `ticket_subject_template` option, which contains a ?Genshi text template snippet. The default value is:

```
$prefix #$ticket.id: $summary
```

The following variables are available in the template:

- `env`: The project environment (see ?env.py).
- `prefix`: The prefix defined in `smtp_subject_prefix`.
- `summary`: The ticket summary, with the old value if the summary was edited.
- `ticket`: The ticket model object (see ?model.py). Individual ticket fields can be addressed by appending the field name separated by a dot, e.g. `$ticket.milestone`.

Customizing the e-mail content

The notification e-mail content is generated based on `ticket_notify_email.txt` in `trac/ticket/templates`. You can add your own version of this template by adding a `ticket_notify_email.txt` to the templates directory of your environment. The default looks like this:

```
$ticket_body_hdr
$ticket_props
#choose ticket.new
  #when True
$ticket.description
  #end
  #otherwise
    #if changes_body
Changes (by $change.author):

$changes_body
    #end
    #if changes_descr
      #if not changes_body and not change.comment and change.author
Description changed by $change.author:
      #end
$changes_descr
--
    #end
    #if change.comment

Comment${not changes_body and '(by %s)' % change.author or ''}:

$change.comment
    #end
  #end
#end

--
```

```
Ticket URL: <$ticket.link>
$project.name <${project.url or abs_href()}>
$project.descr
```

Sample Email

```
#42: testing
---------------------------+-----------------------------------------------
       Id:  42             |      Status:  assigned
Component:  report system  |    Modified:  Fri Apr  9 00:04:31 2004
 Severity:  major          |   Milestone:  0.9
 Priority:  lowest         |     Version:  0.6
    Owner:  anonymous      |    Reporter:  jonas@example.com
---------------------------+-----------------------------------------------
Changes:
  * component:  changset view => search system
  * priority:  low => highest
  * owner:  jonas => anonymous
  * cc:  daniel@example.com =>
         daniel@example.com, jonas@example.com
  * status:  new => assigned

Comment:
I'm interested too!

--
Ticket URL: <http://example.com/trac/ticket/42>
My Project <http://myproj.example.com/>
```

Using GMail as the SMTP relay host

Use the following configuration snippet

```
[notification]
smtp_enabled = true
use_tls = true
mime_encoding = base64
smtp_server = smtp.gmail.com
smtp_port = 587
smtp_user = user
smtp_password = password
```

where *user* and *password* match an existing GMail account, *i.e.* the ones you use to log in on ?http://gmail.com

Alternatively, you can use `smtp_port = 25`.
You should not use `smtp_port = 465`. It will not work and your ticket submission may deadlock. Port 465 is reserved for the SMTPS protocol, which is not supported by Trac. See #7107 for details.

Filtering notifications for one's own changes

In Gmail, use the filter:

```
from:(<smtp_from>) (("Reporter: <username>" -Changes) OR "Changes (by <username>)")
```

For Trac .10, use the filter:

```
from:(<smtp_from>) (("Reporter: <username>" -Changes -Comment) OR "Changes (by <username>)" OR "Comment
```

to delete these notifications.

In Thunderbird, there is no such solution if you use IMAP (see ?http://kb.mozillazine.org/Filters_(Thunderbird)#Filtering_the_message_body).

The best you can do is to set "always_notify_updater" in conf/trac.ini to false. You will however still get an email if you comment a ticket that you own or have reported.

You can also add this plugin: ?http://trac-hacks.org/wiki/NeverNotifyUpdaterPlugin

Troubleshooting

If you cannot get the notification working, first make sure the log is activated and have a look at the log to find if an error message has been logged. See TracLogging for help about the log feature.

Notification errors are not reported through the web interface, so the user who submit a change or a new ticket never gets notified about a notification failure. The Trac administrator needs to look at the log to find the error trace.

Permission denied error

Typical error message:

```
...
File ".../smtplib.py", line 303, in connect
  raise socket.error, msg
error: (13, 'Permission denied')
```

This error usually comes from a security settings on the server: many Linux distributions do not let the web server (Apache, ...) to post email message to the local SMTP server.

Many users get confused when their manual attempts to contact the SMTP server succeed:

```
telnet localhost 25
```

The trouble is that a regular user may connect to the SMTP server, but the web server cannot:

```
sudo -u www-data telnet localhost 25
```

In such a case, you need to configure your server so that the web server is authorized to post to the SMTP server. The actual settings depend on your Linux distribution and current security policy. You may find help browsing the Trac ?MailingList archive.

Relevant ML threads:

- SELinux: ?http://article.gmane.org/gmane.comp.version-control.subversion.trac.general/7518

For SELinux in Fedora 10:

```
$ setsebool -P httpd_can_sendmail 1
```

Suspected spam error

Some SMTP servers may reject the notification email sent by Trac.

The default Trac configuration uses Base64 encoding to send emails to the recipients. The whole body of the email is encoded, which sometimes trigger *false positive* SPAM detection on sensitive email servers. In such an event, it is recommended to change the default encoding to "quoted-printable" using the `mime_encoding` option.

Quoted printable encoding works better with languages that use one of the Latin charsets. For Asian charsets, it is recommended to stick with the Base64 encoding.

501, 5.5.4 Invalid Address error

On IIS 6.0 you could get a

```
Failure sending notification on change to ticket #1: SMTPHeloError: (501, '5.5.4 Invalid Address')
```

in the trac log. Have a look ?here for instructions on resolving it.

See also: TracTickets, TracIni, TracGuide