

Wikiprint Book

Title: チケット更新情報のメール通知

Subject: SilverFrost - TracNotification

Version: 3

Date: 06/04/26 06:06:24

SilverFrost 目次

| | |
|--------------------------------|---|
| チケット更新情報のメール通知 | 3 |
| 通知メールを受け取る | 3 |
| 通知メールで受信するメールアドレスを設定する方法 | 3 |
| SMTP 通知のコンフィグレーション | 3 |
| コンフィグレーション オプション | 3 |
| コンフィグレーションの例 (SMTP) | 4 |
| コンフィグレーションの例 (sendmail) | 4 |
| メールの件名をカスタマイズする | 4 |
| メールの内容をカスタマイズする | 5 |
| Email サンプル | 5 |
| SMTP リレーホストとして GMail を使用する | 6 |
| 自分が変更した通知をフィルタする | 6 |
| トラブルシューティング | 6 |
| Permission denied エラー | 6 |
| Suspected spam エラー | 7 |
| 501, 5.5.4 Invalid Address エラー | 7 |

チケット更新情報のメール通知

Trac はチケット更新情報をメールで通知出来ます。(英語版では Notification)

メールによる更新通知は、ユーザに関係するチケットについての最新情報を随時に提供したり、専用 ML にチケットの更新情報を配信したりするのに便利な機能です。例えば、[Trac-tickets](#) を見るとどのようにセットアップされているか見ることができます。

通知機能はデフォルトでは無効になっています。有効化とコンフィグレーションは [trac.ini](#) で行うことができます。

通知メールを受け取る

新しいチケットを登録したりコメントを追加する際、報告者(英語版では reporter)、担当者(英語版では assigned to/owner) もしくは 関係者(英語版では cc) フィールドに有効なメールアドレスやユーザ名を入力してください。チケットに更新があると、Trac が自動的にメールを送信します。(どのように通知メール機能を設定しているかによります)

これは、興味のあるチケットや機能追加に関する最新情報を随時受け取りたい場合に便利な機能です。

通知メールで受信するメールアドレスを設定する方法

完全なメールアドレスあるいはユーザー名を入力することで、通知メールを受け取ることができます。単純なユーザ名やログイン名で通知を受け取るには、ユーザ設定 (Preferences) ページで、正しいメールアドレスを入力する必要があります。

[TracIni](#) ファイルにデフォルトドメイン名 (`smtp_default_domain`) を指定することで、単純なユーザ名で通知を受け取ることができるようになります。(後述する [コンフィグレーション オプション](#) を参照してください)。この場合、ユーザ名にデフォルトドメインが追加されますので、"イントラネット" 環境で仕様している場合は便利です。

SMTP 通知のコンフィグレーション

重要: [TracNotification](#) を正しく動作させるには、[trac.ini](#) に `[trac] base_url` を設定する必要があります。

コンフィグレーション オプション

`trac.ini` の `[notification]` セクションで設定できるオプションです。
(訳注: 0.10 以降では `TracIni` マクロを使用することで正確なオプション一覧を取得できます。以下のリストは必要に応じて `[[TracIni(notification)]]` に置き換えて使用してください。)

- `smtp_enabled`: メール通知を有効にします。
- `smtp_from`: 通知メールの Sender-ヘッダに使用するメールアドレス。
- `smtp_from_name`: 通知メールの Sender-ヘッダに使用する送信者名。
- `smtp_replyto`: 通知メールの Reply-To-ヘッダに使用するメールアドレス
- `smtp_default_domain`: (0.10 以降)
ドメイン名を含んでいないアドレスに特定のドメイン名を追加します。完全修飾されたアドレスは修正されません。ユーザの設定からメールアドレスが分からず / ログイン名に追加します。
- `smtp_always_cc`: 常に通知メールを送信するメールアドレスのリスト。主に専用MLにチケット更新を通知するのに使用されます。
- `smtp_always_bcc`: (0.10 以降)
常に通知メールを送信するメールアドレスの一覧。ただし、通知メールの他の受信者からメールアドレスが見えません。
- `smtp_subject_prefix`: (0.10.1 以降) e-mail の subject の前に挿入されるテキスト。デフォルトは `"_default_"`
- `always_notify_reporter`: 報告者フィールドのすべてのメールアドレスに常に通知メールを送信します。(default: false)
- `always_notify_owner`: (0.9 以降) 担当者フィールドのメールアドレスに常に通知メールを送信します。(default: false)
- `always_notify_updater`: (0.10 以降) チケットの更新者に常に通知メールを送信します。(default: false)
- `use_public_cc`: (0.10 以降) To: (担当者, 報告者) と CC: のアドレスのリストがすべての受信者に見えるかどうかを設定します (デフォルトは Bcc: で見えません)
- `use_short_addr`: (0.10 以降) 通知メールの配信でメールアドレスがドメイン名を含むようにするかどうかを設定します (例 `@<domain.com>` で終わらないようにする)。このオプションは SMTP サーバがローカルアドレスをハンドリングでき、ローカルのメールボックスとユーザ名/ログイン名をマップできるような環境、イントラネット内で使用するときに
- `mime_encoding`: (0.10 以降) MIME のエンコードのスキームを選択します。サポートする値:
 - `none`: デフォルト値。テキストがプレーンな ASCII 文字コード、もしくはその他 8 ビットの文字コードであるときに、7 ビットエンコードを使用します。

- `base64`: どのような内容が含まれていても動作します。敏感なアンチスパム / アンチウイルスエンジンではいくつかの問題があるかもしれません。
- `qp` または `quoted-printable`: 8 ビットエンコーディングが使用できない場合、ヨーロッパの言語に対して最適です (`base64` よりもコンパクトです)。
- `ticket_subject_template`: (0.11 以降) 通知メールの件名 (Subject) に使用する [Genshi テキストテンプレート](#) 。
- `email_sender`: (0.12 以降) `IEmailSender` インタフェースを実装しているコンポーネント名。通知システムは、ここで設定されたコンポーネントを使用してメールを送信します。現在のところ Trac では以下のコンポーネントが提供されています:
 - `SmtplibEmailSender`: SMTP サーバに接続してメールを送信する (デフォルト)。
 - `SendmailEmailSender`: `sendmail` 互換の実行ファイルを実行してメールを送信する。

`smtp_from` か `smtp_replyto` の一方 (もしくは両方) が 必ず 設定されていなければなりません。設定されていない場合、Trac は通知メールの送信を拒否します。

以下のオプションは SMTP でのメッセージ伝送方法を指定するために使用します。

- `smtp_server`: 通知メッセージに使用されるSMTPサーバ。
- `smtp_port`: (0.9 以降) SMTP サーバが使用するポート番号。
- `smtp_user`: (0.9 以降) SMTP 認証アカウントのユーザID。
- `smtp_password`: (0.9 以降) SMTP 認証アカウントのパスワード。
- `use_tls`: (0.10 以降) SMTP サーバ経由で通知メールを送信するときに、GMail などの [TLS](#) を使用するかどうかを切り替えます。

以下のオプションは `sendmail` 互換の実行ファイルを使用する場合のメッセージ伝送方法を指定するために使用します。

- `sendmail_path`: (0.12 以降) `sendmail` の実行ファイルへのパスを設定します。ここで指定する `sendmail` プログラムは `-i` および `-f` オプションを解釈できなければなりません。

コンフィグレーションの例 (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
```

コンフィグレーションの例 (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
```

メールの件名をカスタマイズする

`ticket_subject_template` オプションを使用することでメールの件名をカスタマイズできます。このオプションには [Genshi テキストテンプレート](#) を含めることができます。デフォルト値は以下の通りです:

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

テンプレートでは以下の変数が使用可能です:

- `env`: プロジェクトの Environment ([env.py](#) 参照)。
- `prefix`: `smtp_subject_prefix` で定義したプレフィックス。
- `summary`: チケットの概要 (Summary), 概要が変更されている場合は古い値。

- ticket: ticket model オブジェクト ([model.py](#) 参照)。個別のチケット属性は `$ticket.milestone` のように、ドット (.) で区切って (訳注: 英語での) 属性名を加えることで、参照可能。

メールの内容をカスタマイズする

通知メールの内容は `trac/ticket/templates` の `ticket_notify_email.txt` を元に生成されます。編集した `ticket_notify_email.txt` を Environment の `templates` ディレクトリに配置することで、カスタマイズすることができます。デフォルトはこのようになっています:

```
$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
```

Email サンプル

```
#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/>
```

SMTP リレーホストとして GMail を使用する

以下のような設定を使用します

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

user と password の箇所は、GMail のアカウントで置き換えてください。たとえば <http://gmail.com> にログインするときに使用しているものです。

代わりに smtp_port = 25 を使用することもできます。(訳注: おそらく gmail.com 宛のメールしか届きません)

しかし smtp_port = 465 は使用できません。これは動作しないだけでなく、通知メール送信がデッドロックします。ポート番号 465 は SMTPS プロトコル (訳注: SSMTP とも言います) に使用されますが、Trac はこれをサポートしていません。詳しくは [#7107](#) を参照してください。

自分を変更した通知をフィルタする

Gmail では、以下のフィルタを使用できます:

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

Trac 0.10 の場合は、下記のフィルタを使用してください:

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

通知メールを削除する場合などにも使用できます。

Thunderbird で IMAP を使用している場合は、この方法は使えません ([http://kb.mozillazine.org/Filters_\(Thunderbird\)#Filtering_the_message_body](http://kb.mozillazine.org/Filters_(Thunderbird)#Filtering_the_message_body) 参照)。

最良の方法は conf/trac.ini の "always_notify_updater" を false に設定することです。しかし、この方法では自分が担当者 (owner) になっていたり、報告 (reporter) したチケットへのコメントが取得できなくなります。

プラグインを追加することで解決するかもしれません: <http://trac-hacks.org/wiki/NeverNotifyUpdaterPlugin>

トラブルシューティング

通知メールの設定がうまくいかないとき、最初にログが出力されるようになっているか確かめて下さい。そしてどういうエラーメッセージが出力されているかを見て [TracLogging](#) にログについて書いてあるので見て下さい。

通知メールのエラーは Web

インタフェースで報告されないので、誰かがチケットを更新したり、新規のチケットを登録しても通知メールが届かないことにまったく気づかないでしょう。Trac の管理者はエラートレースを追い、ログをみる必要があります。

Permission denied エラー

たいていのエラーメッセージ:

```
...
File ".../smtplib.py", line 303, in connect
```

```
raise socket.error, msg
error: (13, 'Permission denied')
```

このエラーはたいていサーバのセキュリティの設定に起因します: 多くの Linux のディストリビューションでは Web サーバ (Apache など ...) からローカルの SMTP サーバにメールを送信させません。

多くのユーザはマニュアルに SMTP サーバに接続できるか試すよう書いてあり、成功するので混乱するでしょう:

```
telnet localhost 25
```

トラブルの原因は、一般ユーザは SMTP サーバに接続できるけれども、Web サーバは接続できないということです:

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

このような場合、Web サーバが SMTP サーバにメールを送信するときに認証を行うように設定する必要があります。実際の設定は使用している Linux のディストリビューションと現在のセキュリティのポリシーによります。Trac の [MailingList](#) のアーカイブを参照して下さい。

関係のあるメーリングリストのスレッド:

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

Fedora 10 の SELinux では下記のコマンドで対処できます:

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

Suspected spam エラー

SMTP サーバの中には Trac によって送信される通知メールを拒否するものもあるでしょう。

Trac はデフォルトで通知メールを Base64 エンコーディングして受信者に送信します。メールの本文 (Body) 全体がエンコードされる (訳注: チケットのタイトルに日本語が含まれる場合、通知メールの Subject ヘッダも utf-8 base64 エンコーディングされます) ので、繊細なメールサーバ上の false positive な SPAM 発見プログラムのトリガになってしまいます。このような状況に遭遇した場合、mime_encoding オプションでデフォルトのエンコーディングを "quoted-printable" に変更することを推奨します。

"quoted-printable" エンコーディングは ラテン系の文字セットで使用すると効果があります。アジア系の文字セットでは、Base64 エンコーディングに固定することを推奨します。(訳注: 日本語のメールで主に使用されるエンコードは ISO-2022-JP ですが、Trac が送信する通知メールは UTF-8 でエンコードされ、さらに BASE64 が quoted-printable で 7bit-safe な形式にエンコードされます。ISO-2022-JP エンコーディングでない日本語のメールは Spam 検出エンジンに誤判定されやすくなる傾向があるようです。また、MUA がこれらのエンコードに対応していない場合、たとえ受信できたとしても文字化けしたり表示できない可能性があります。)

501, 5.5.4 Invalid Address エラー

IIS 6.0 で

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

上記のエラーが trac のログにでる場合、[ここ](#)を参考に解決してください。

See also: [TracTickets](#), [TracIni](#), [TracGuide](#)