Trac Installation Guide for 0.12

Trac は Python で書かれており、データベースとして SQLite, PostgreSQL, MySQL のどれかが必要です。 Trac は HTML レンダリングのために Genshi テンプレートシステムを使用します。

バージョン 0.12 以降で Trac はローカライズされているため、自分が普段使っている言語に翻訳されているかもしれません。 Trac のインタフェースで別の言語を使用したい場合は、任意のパッケージである Babel を 最初に インストールする必要があります。 Babel がない場合、通常通り、デフォルトの英語バージョンのみ使用することができます。もし、 Babel を Trac よりも後にインストールした場合は、 Trac を再インストールする必要があります。

新しい言語での翻訳の作成や、すでにある言語での翻訳のエンハンスをコントリビュートした場合は TracL10N を見てみてください。 (訳注: 日本語の翻訳はすでにコントリビュートされています)

Trac のインストールとセットアップに対する一般的な手順を以下に示します。 Trac を特定のシステムにインストールする手順は Trac Project サイトの TracInstallPlatforms にありますが、 まず最初に以下の一般的な手順を読み通して タスクの関係を確実に理解してください。

SilverFrost 目次

依存関係

必須の依存関係

Trac をインストールするためには以下のソフトウェアパッケージがインストールされていなければなりません:

- Python, 2.4 以上 3.0 未満 (Pyhton 2.3 のサポートは、このリリース (訳注: 0.12) で打ち切られました)
- setuptools, 0.6 以上
- Genshi, 0.6 以上

また、データベースと、それに対応する Python のバインディングが必要です。 データベースは SQLite, PostgreSQL, MySQL のいずれかが使用できます。

SQLite の場合

Python 2.5 か 2.6 の場合は、必要なライブラリは同梱されています。

Python 2.4 の場合、 pysqlite が必要です。 pysqlite は google code から Windows インストーラやソースからのビルド用の tar.gz アーカイブがダウンロードできます:

```
$ tar xvfz <version>.tar.gz
$ cd <version>
$ python setup.py build_static install
```

上記の手順で SQLite のコードも展開されバインディングがビルドされます。

SQLite をインストールするときに、システムが開発用のヘッダを必要とするかもしれません。これらのヘッダなしでは、ビルドしたときに GCC 関連のエラーがいろいろと出るでしょう:

```
$ apt-get install libsqlite3-dev
```

SQLite 2.x も PySqlite 1.1.x ももはやサポートしていません。

既知のバグとして、 PySqlite バージョン 2.5.2-4 では、 Trac のデータベースを 0.11.x から 0.12. にアップグレードすることができません。 2.5.5 以降か 2.5.1 と それ以前のバージョンを使用して下さい。詳細については、 本家チケット 9434 を参照して下さい。

PySqlite も参照することができます。

PostgreSQL の場合

下記いずれかの Python バインディングをインストールする必要があります:

- PostgreSQL, バージョン 8.0 以降
- psycopg2

詳しくは DatabaseBackend を参照してください。

MySQL の場合

Trac は以下のガイドラインで、 MySQL でも良好に動作するようになりました。

- MySQL, 5.0 以降
- MySQLdb, 1.2.2 以降

非常に 重要なことが記載されているので、データベースを作成する前に、 MySqlDb のページを注意深く読んで下さい。

任意の依存関係

バージョン管理システム

Subversion

- Subversion, 1.5.x または 1.6.x と 対応する Python バインディング。 1.4 系の古いバージョンも、今のところ動作します。トラブルシューティングの情報が TracSubversion のページに記載されていますので、確認してみてください。1.4.0より以前のバージョンでは、動作しないかもしれません。というのも、 Trac が使用する svn の中心機能（例:svn_path_canonicalize）が（svn ライブラリそのものには存在していたとしても）svn のバージョン 1.3.x 以前では python の swig のラッパーで実装されていないからです。

主要なプラットフォーム向けに コンパイル済みの SWIG バインディング が用意されていますので、通常はこれを使ってください。 (頑張って Windows 用のコンパイル済み SWIG バインディングをリストから見つけて下さい。 TracSubversion が Algazam を指し示すでしょう。 Python 2.6 では動作します。)

Note: Trac は PySVN のような新しい `ctype` 形式のバインディングでは 動作しません。 [ctype バインディングを実装するチケットはあるかな？]

重要な Note: Subversion を使用するなら Trac を 同じマシン にインストールする必要があります。リモートリポジトリは現在サポートされていません。

その他のバージョン管理システム

Subversion 以外のバージョン管理システムのサポートはサードパーティから提供されます。 PluginList や VersioningSystemBackend を参照して下さい。

Web サーバ

Trac にはサーバ機能が組み込まれているので、 Web サーバは必須ではありません。このページに下にある スタンドアロンサーバの起動 セクションを参照してください。

Trac は下記の要件を満たす Web サーバで動作します。

Apache との組み合わせで
  - mod_wsgi, TracModWSGI および http://code.google.com/p/modwsgi/wiki/IntegrationWithTrac を参照
  - mod_python 3.3.1, 廃止予定: TracModPython 参照)
- FastCGI が使用可能な Web サーバ (TracFastCgi 参照)
- AJP が使用可能な Web サーバ (TracOnWindowsIisAjp 参照)
- CGI が使用可能な Web サーバ (TracCgi 参照), しかし Trac を CGI スクリプトとして使用することは 全く推奨されていませんので、上に挙げた方法を選択するようにして下さい。

その他の Python パッケージ

- Babel, 0.9.5 以上 ローカライズの機能を使用する場合は必要。 Note: 他の言語で Trac のインタフェースを使用したい場合、必ず最初に Babel をインストールして下さい。 Babel をインストールしていない場合は、通常通りデフォルトの英語バージョンの Trac を使用することができます。 Babel を後からインストールした場合、 Trac を再インストールする必要があります。
- docutils, 0.3.9 以上 WikiRestructuredText の機能を使用する場合は必要
- Pygments シンタックスハイライティング の機能を使用する場合は必要。 SilverCity や Enscript も、今のところ使用できますが、サポートを打ち切る予定なので、 Pygments を使用して下さい。
- pytz, タイムゾーンの完全なリストを取得する場合に必要。 pytz がない場合、 Trac は内部で定義している短いタイムゾーンの実装にフォールバックします。

Attention: これらの依存関係は様々なバージョンで必ずしも置き換えできるとは限らないので、上記のバージョン番号に注意してください。 Trac を動かす上で問題が発生した場合は メーリングリスト や IRC チャネル で質問をする前にすべての依存関係を再度確認してください。

これらのパッケージのドキュメンテーションを参照して、それらが最も上手にインストールできる方法を探してください。また プラットフォーム特有の説明 の多くに、これらの依存関係のインストール方法が記述されています。しかしながら プラットフォーム特有の説明 の情報はあなたがインストールしている Trac より古い バージョンについての説明があることを覚えておいてください (なんと Trac 0.8 に関する説明をしているページもあります)。

Trac のインストール

`easy_install` を使用したインストール

Trac をインストールする方法のひとつに setuptools の利用があります。 setuptools を使用すると、 Trac を Subversion リポジトリからインストールすることもできます;

例:

- 始めに最新の安定バージョン Trac 0.12.1 とローカライズサポートをインストールする:

```
easy_install Babel==0.9.5 Genshi==0.6
easy_install Trac
```

easy_install コマンドを二回、別々に実行することはとても重要です。そうしなければ、メッセージカタログが生成されません。

- 最新バージョンの Trac にアップグレードする:

```
easy_install -U Trac
```

- 最新の開発中のバージョン (0.13dev) にアップグレードする:

```
easy_install -U Trac==dev
```

アップグレードの場合には、必ず TracUpgrade を読むようにして下さい。

ソースからのインストール

よりインストールを管理したい場合には、アーカイブにあるソースをダウンロードできます。また、本家 ソースコードリポジトリ からチェックアウトすることができます。

必ず事前にインストール条件を整えておいて下さい。 Genshi と Babel のソースパッケージは http://www.edgewall.org で手に入れることができ、同様の手順でインストールすることができます。また、これらのパッケージは単に easy_install でインストールすることもできます。上記 参照。

Trac アーカイブの解凍またはチェックアウトを実行した後、トップレベルのディレクトリに移動し、以下を実行します:

```
$ python ./setup.py install
```

このステップを実行するためには root 権限 (または root 権限と同等の権限) が必要です。

この操作で Python のソースコードがバイトコンパイルされ、 .egg ファイルかディレクトリが Python インストールの `site-packages` ディレクトリにインストールされます。 .egg には htdocs や templates のような、ソースファイル以外に標準インストールの Trac が必要とするすべてのリソースが含まれています。

このスクリプトは tracd スタンドアロンサーバと一緒に、 プロジェクト Environment を作成し維持するための trac-admin コマンドラインツールをインストールします。

ソースからインストールする場合や、新しい言語で Trac を国際化するためには Babel をインストールしておく必要があります。この場合も `install` を実行するだけです (Babel がインストールされていない状態で、すでに Trac をインストールしてしまった場合でも、 `install` をやり直すことで Babel サポートを有効化できます):

```
$ python ./setup.py install
```

また、 `bdist_egg` を実行すると dist ディレクトリに作成される .egg ファイルをインストール先にコピーしたり、 (`bdist_wininst` の実行によって) Windows インストーラを作成しても構いません。

高度なオプション

Trac のインストール場所を変えるなどの高度なインストールオプションを知りたければ以下を実行してください:

```
easy_install --help
```

詳細な情報は Python モジュールをインストールする を参照してください。

特にあなたは以下に興味を持つかもしれません:

```
easy_install --prefix=/path/to/installdir
```

Mac OS X に Trac をインストールする場合:

```
easy_install --prefix=/usr/local --install-dir=/Library/Python/2.5/site-packages
```

Note: Mac OS X 10.6 上で easy_install http://svn.edgewall.org/repos/trac/trunk を使用する場合は、オプションを指定しなくても /usr/local および /Library/Python/2.6/site-packages にインストールされます。

上記の例は、 tracd と trac-admin コマンドを /usr/local/bin に、 Trac のライブラリと依存ファイルを /Library/Python/2.5/site-packages にインストールします。これらのパスは Apple での Pyhton サードパーティアプリケーションの標準ロケーションです。(訳注: つまり、上記と違うパスにインストールしたい場合のみ、オプションの指定が必要になります)

pip を使用したインストール

'pip' は easy_install のリプレースであり、とても簡単に素早く Python パッケージをインストールすることができます。 Trac を 5 分程度でインストール、起動できます:

pip によってインストールされる場所を /opt/user/trac とした場合の例です:

```
pip -E /opt/user/trac install trac psycopg2
```

または

```
pip -E /opt/user/trac install trac mysql-python
```

PostgreSQL (libpg-dev) や MySQL (libmysqlclient-dev) へのバインディングも自動でビルドできるように、 pip が OS 固有のヘッダファイルを確実に利用できるようにして下さい。

また pip は (Genshi, Pygments などの) 依存関係を解決し、 pypi.python.org から最新のパッケージをダウンロードして、 /opt/user/trac の配下にインストールするところまで自動化されています。

すべてのコマンド (tracd, trac-admin) は /opt/user/trac/bin の配下にインストールされます。 mod_python (PythonHandler ディレクティブを使用する場合) や mod_wsgi (!WSGIDaemonProcess ディレクティブを使用する場合) においても活用することができます。

加えて、 Trac プラグインのうちのいくつか (ここ で一覧を見ることができます) も pip からインストールすることが可能です。

プロジェクト Environment の作成

Trac Environment は Trac が Wiki ページ、チケット、レポート、設定などの情報を保存するバックエンドストレージです。基本的に人間が読み込み可能な 構成ファイル と他の様々なファイルやディレクトリで構成されます。

新しい Environment は trac-admin を使用して作成します:

```
$ trac-admin /path/to/myproject initenv
```

trac-admin は、プロジェクトの名前や データベース接続文字列 など、 Environment を新規作成するために必要な情報を入力するためのプロンプトを表示します。これらの設定項目について特に変更が必要ない場合は、単に <Enter> を押下すると、デフォルト値が使用されます。

データベース接続文字列のデフォルトは SQLite が使用されます。 SQLite がインストールされている場合は、他の設定は不要です。 他の データベースバックエンド を使用する場合は、あらかじめデータベースが使用可能な状態にしておかねばなりません。

0.12 以降で Trac は、新しい Environment の作成時に ソースコードリポジトリ を尋ねないようになりました。リポジトリを後で 追加する までの間、バージョン管理のサポートは無効化されます。

また、 ここで指定した値は conf/trac.ini 設定ファイルを直接編集することで後から変更できます。

最後に、 Web のフロントエンドを実行しているユーザアカウントは、 Environment のディレクトリと、その中のすべてのファイルに対する書き込み権限が必要です。 trac-admin ... initenv

の実行を該当するユーザで実行した場合は、この作業は不要ですが、そうでない場合、ただしユーザに権限を付与する作業が必要になります。たとえば Linux で apache ユーザ apache グループで Web サーバを起動する場合は:

```
# chown -R apache.apache /path/to/myproject
```

警告 アカウント名とプロジェクトパスには ASCII 文字のみを使用して下さい。 unicode 文字はサポートしていません。

スタンドアロンサーバの起動

Trac 環境を作成した後に、スタンドアロンサーバ tracd を実行することで簡単に Web インタフェースを試すことができます。

```
$ tracd --port 8000 /path/to/myproject
```

ブラウザを起動して、 http://localhost:8000/ にアクセスしてください。 tracd が認識しているすべての Environment の簡単な一覧が表示されます。作成した Environment へのリンクにアクセスすることで Trac が動作中であることを確認できます。 Trac でプロジェクトをひとつだけ管理したい場合、以下のように起動することで、スタンドアロンサーバは Environment 一覧の表示をスキップして、直接 Environment を表示します:

```
$ tracd -s --port 8000 /path/to/myproject
```

Web サーバ上での Trac の起動

Trac に "真の" Web サーバから接続するには、いくつかの方法があります: CGI, FastCGI, mod_wsgi, mod_python です。まともな性能を出すには FastCGI か mod_wsgi のどちらかを使用することが推奨されます。

Trac では AJP も使用できます。これを使うと IIS とも接続することができます。

Trac の cgi-bin ディレクトリを生成する

Trac を FastCGI などで正しく機能させるには、 FastCGI であれば trac.fcgi ファイル、 mod_wsgi であれば trac.wsgi ファイルが必要となります。これらのファイルは適切な Python コードをロードする Python スクリプトです。 trac-admin コマンドの deploy オプションを使用することで生成できます。

若干の「卵が先か鶏が先か」問題があります。 trac-admin コマンドが機能するためには Environment が必要なのですが、 deploy には既に存在するディレクトリは使用できません。これに起因して、 Environment は depoly するディレクトリのサブディレクトリを使用することができません。この制限を回避するには次のようにします:

```
mkdir -p /usr/share/trac/projects/my-project
trac-admin /usr/share/trac/projects/my-project initenv
trac-admin /usr/share/trac/projects/my-project deploy /tmp/deploy
mv /tmp/deploy/* /usr/share/trac
```

プラグインキャッシュの設定

Python
プラグインの中にはキャッシュディレクトリを必要とするものがあります。デフォルトではキャッシュディレクトリは、現在のユーザのホームディレクトリに置かれ
Trac を Web サーバで動作させている場合、ホームディレクトリを持たない専用ユーザであることが多く
(強く推奨します)、プラグインの起動が妨げられることがあります。キャッシュディレクトリの場所を変更するには、環境変数 PYTHON_EGG_CACHE を設定してください。環境変数を設定する方法の詳細は使用しているサーバのドキュメントから参照してください。

認証の構成

認証のためのユーザアカウントを追加、削除、構成する方法は Trac を起動する方法により異なります。基本的な手順は TracCgi ページの 認証を追加する セクションで説明されていますが、各フロントエンドのための認証をセットアップする方法は、以下のいずれかを参照してください:

- スタンドアロンサーバ tracd を使用する場合は TracStandalone
- CGI か FastCGI を使用する場合は TracCgi
- Apache の mod_wsgi を使用する場合は TracModWSGI
- Apache の mod_python を使用する場合は TracModPython

SVN のチェンジセットを Trac のチケットに自動リンクする方法

変更をリポジトリにコミットした時に、チェンジセットへのリンクをチケットのコメントに自動で追加するように SVN を設定することができます。コミットメッセージには以下に示すいずれかの書式が含まれていなければなりません:

- Refs #123 - このチェンジセットへのリンクをチケット #123 に追加します
- Fixes #123 - このチェンジセットへのリンクをチケット #123 に追加し、チケットを fixed でクローズします。

この機能を使用するためには post-commit フックを TracRepositoryAdmin に記載したリポジトリにインストールし、 commit updater コンポーネントを有効にせねばなりません。コンポーネントの有効化は、 trac.ini ファイルの [components] セクションに下記記述を追加するか、"プラグイン" (英語版では "Plugins") 管理パネルから設定します。

```
tracopt.ticket.commit_updater.* = enabled
```

詳細な情報は "プラグイン" 管理パネルの CommitTicketUpdater コンポーネントにあるドキュメントを参照してください。

Tracを使用する

一度 Trac サイトを稼働させれば、チケットを作成したり、タイムラインを見たり、 (設定されていれば) バージョン管理のリポジトリを閲覧したりできるはずです。

anonymous (ログインしていない) でアクセスするユーザは、デフォルトでほとんどの機能を使用することができますが、すべての機能を使用できるわけではないことに留意してください。すべての機
アクセス許可 を与える必要があるでしょう。

Enjoy!

The Trac Team

See also: TracInstallPlatforms, TracGuide, TracCgi, TracFastCgi, TracModPython, TracModWSGI, TracUpgrade, TracPermissions