

## アップグレードの説明

### 一般的な手順

通常、Trac を新しいバージョンにアップグレードするときに、7 ステップを踏まなければなりません:

#### 1. Trac のコードを更新する

[TracInstall](#) または、あなたの OS に合った方法で新しいバージョンの Trac を取得してください。

`easy_install` を使って、バージョン 0.11 の Trac をインストールした場合は、Trac のアップグレードにも `easy_install` を使うのが最も簡単な方法でしょう:

```
# easy_install --upgrade Trac==0.12
```

手動で (OS 特有でない) アップグレードをするのであれば、インストールを実行する前に起動中の Trac サーバを停止してください。"ホット" アップグレードは問題を生じることが多いです。特に Windows では出来ないと考えてください ([本家チケット 7265](#))。

すでに存在する Trac のコードを削除するには、Python の `lib/site-packages` ディレクトリから `trac` ディレクトリか、Trac の `.egg` の古いバージョンを削除します。 `site-packages` ディレクトリの位置は OS のシステム、および Python のインストールパスにより異なりますが、一般的には以下の位置にあります:

- Linux の場合: `/usr/lib/python2.X/site-packages`
- Windows の場合: `C:\Python2.X\lib\site-packages`
- MacOSX の場合: `/Library/Python/2.X/site-packages`

また、`share/trac` (正確な位置はプラットフォームに依存しますが一般的にはこの位置です。) ディレクトリ内の `cgi-bin`, `htdocs`, `templates`, `wiki-default` といったディレクトリを削除してもかまいません。(訳注: 0.11 では、これらのディレクトリは `site-packages/trac` の配下に移動しています)

このクリーンアップは必須ではありませんが、あとでトラブルシュートを行う場合の切り分けが容易になります。すでに使われていない前のリリースのコードやテンプレートは削除してください。

#### 2. [TracEnvironment](#) をアップグレードする

Environment のアップグレードは、マイナーバージョンアップに特別な注意書きがない限り、不要です。

アップグレードした Trac

がロードされると、アップグレードする必要があるインスタンスが表示されます。アップグレードはオートメーションされたスクリプトを手で実行します。これらの [trac-admin](#) を使用します。

```
trac-admin /path/to/projenv upgrade
```

このコマンドはもし [TracEnvironment](#) がすでに最新の状態になっているときは、何もしません。

Note: データベースのバックアップはアップグレード時に自動で行われます。この機能は最近の更新で、データベースバックエンドに PostgreSQL や MySQL

を使用している場合にも対応しましたが、失敗してしまう場合は手動でバックアップしてください。その後、バックアップをスキップしてアップグレードを行うための

```
trac-admin /path/to/projenv upgrade --no-backup
```

#### 3. Trac ドキュメントを更新する

すべての [Trac Environment](#) で、インストールされたバージョンの Trac ドキュメントのコピーを含んでいます。新しくインストールした Trac のドキュメントと同期を取りたいでしょう。 [trac-admin](#) がドキュメントを更新するコマンドを提供しています:

```
trac-admin /path/to/projenv wiki upgrade
```

このプロシージャはあなたの WikiStart ページ (訳注: InterMapText も) をまったく変更せず、そのままに残しておきます。

#### 4. Trac Environment とソースコードリポジトリの再同期

Trac でソースコードをブラウズしているときに "[リポジトリにチェンジセット ??? が存在しません](#)" のようなエラーが出る場合は、それぞれの [Trac environment](#) のソースコードリポジトリと再同期する必要があります。

```
trac-admin /path/to/projenv repository resync '*'
```

## 5. 静的リソースをリフレッシュする

もし、静的リソースを直接配布できる ( URL `/chrome/` を使用してアクセス) ようにウェブサーバをセットアップしていたら、同じくコマンドを使用してそれらをリフレッシュする必要があります:

```
trac-admin /path/to/env deploy /deploy/path
```

このコマンドは、新しい Trac のバージョンとそのプラグインから `/deploy/path` に静的リソースと CGI スクリプト (`trac.wsgi`, など) を抽出します。

いくつかのウェブブラウザは、CSS や Javascript ファイルをしつこくキャッシュしているので、ユーザにはこれらのブラウザのキャッシュの中身を手動で削除するように指示したほうがいいかもしれません。

## 6. 特定の Trac バージョンでの特記事項

Trac 0.11 から Trac 0.12 にアップグレードする

Python 2.3 サポート

現在 Python 2.4 以降 をサポートします。

SQLite v3.x

SQLite v2.x のサポートは終了しました。もしいまだに Trac で SQLite v2.x のデータベースを使用しているようならば、まず最初に SQLite v3.x に変換する必要があります。詳細は <http://trac.edgewall.org/wiki/PySqlite#UpgradingSQLitefrom2.xto3.x> を参照して下さい。

[PySqlite 2](#)

[PySqlite 1.1.x](#) のサポートは終了しました。可能であれば、バージョン 2.5.5 以降のバージョンをインストールして下さい。(下記 [Trac データベースのアップグレード](#) を参照して下さい。)

複数のリポジトリのサポート

最新のバージョンでは複数リポジトリの取り扱いをサポートしています。Trac に複数のリポジトリを追加する予定であるならば、今では複数のリポジトリを扱えるようになっています。詳細については、[単一リポジトリからの移行手順](#) を参照して下さい。

もし単一のリポジトリでの運用を行っていたとしてもこの手順は興味深いものとなるかもしれません。なぜなら、この方法によってリクエスト毎に発生する潜在的

向上したリポジトリの再同期

複数のリポジトリをサポートするのに加えて、今では Trac と リポジトリの同期でより効果的な方法があります。

バージョン 0.11 と同様に `post-commit` フックを使用した同期方法を続けることもできますが、[リポジトリの同期方法](#) や [明示的な同期](#) に書かれている方法の方が、より効果的な同期を行なうことができ、多かれ少なかれ、複数のリポジトリを扱う際に必須となります。

Note: もし、 `trac-post-commit-hook` を使用していたならば、上記を参照して、新しいフックに アップグレードすることを強くお勧めします。なぜなら、古いフックはデフォルトのリポジトリ以外では動きません。そしてこの場合、適切な通知のトリガーとなりません。

Authz のパーミッションチェック

`authz` のパーミッションチェックが粒度の細かいパーミッションポリシーとしてマイグレートされました。もし `authz` パーミッションを使用しているならば、( `[trac] authz_file` や `authz_module_name` を参照)、 `[trac] permission_policies` で定義するパーミッションポリシーの先頭に、 `AuthzSourcePolicy` を追加しなければなりません。また、グローバルのパーミッション設定から `BROWSER_VIEW`, `CHANGESSET_VIEW`, `FILE_VIEW`, `LOG_VIEW` を削除しなければなりません (`trac-admin $ENV permission remove` コマンドまたは、管理パネルの "権限" から削除されます。)

マイクロ秒のタイムスタンプ

データベースのテーブルにある全てのタイムスタンプ (セッションテーブルは除く) は "エポックからの秒数" から "エポックからのマイクロ秒数" へ値を変更しました。この変更はカスタムレポートを除いて、ほとんどのユーザは意識せずにすむはずですが、レポートで計算に日付/時刻カラムを使用するような場 (例えば、`datetime()` に渡す場合)、データベースから取り出した値を 1,000,000 で割る必要があります。同様に、もしレポートで日付/時刻と表示される計算された値 (カラム名が "time", "datetime", "changetime", "date", "created" や "modified") をマイクロ秒のタイムスタンプで用いる必要がある場合には、先の計算を1,000,000倍して下さい。

Trac 0.10 から Trac 0.11 へのアップグレード

テンプレート と スタイルシート

テンプレートエンジンが Trac 0.11 から Genshi に変わりました。詳細については、[TracInterfaceCustomization](#) を参照して下さい。

もし、[TracEnvironment](#) の `templates` ディレクトリの中のカスタマイズされた CSS や 修正したテンプレートを使用しているならば、Genshi のスタイルに変換する必要があります。カスタマイズしたスタイルシートを使い続けるためには、[サイトの外観](#) の手順に従ってください。

Trac マクロプラグイン

[ClearSilver](#) と HDF が使用されなくなったことで、古いスタイルの Wiki マクロ は使用できなくなります。そのため Trac マクロを適応させる必要があるでしょう; 新しいスタイルのマクロに変更する必要がある場合は [WikiMacros](#) を参照してください。新しいスタイルにコンバートした後、配置するディレクトリは `wiki-macros` ではなく、`plugins` を使用してください。`wiki-macros` ディレクトリからマクロやプラグインを探すことはもうありません。

FCGI/WSGI/CGI を使用する場合

CGI で Trac を起動している場合、以下のコマンドを実行して `trac.*gi` ファイルを取得してください:

```
trac-admin /path/to/env deploy /deploy/directory/path
```

このコマンドでは、デプロイ用ディレクトリを作成します。デプロイ用ディレクトリには `cgi-bin` と `htdocs` の二つのサブディレクトリが含まれています。Apache の `httpd.conf` を新しい `trac.cgi` と `htdocs` の場所に更新してください。

Web アドミンプラグインのインテグレーション

もし、Web アドミンプラグインをインストールしていたら、Trac 0.11 以降より、Trac のコードベースの一部となっているのでアンインストールできます。

7. Web サーバを再起動する

[CGI](#) 以外で起動している場合は、Web サーバを再起動して、新しい Trac コードをリロードしてください。

既知の問題

複数プロジェクトのホストに関して

複数のプロジェクトをホストした場合に、配下のプロジェクトのうち一つのプロジェクトで、プラグインの一つが動作していないとき、配下のすべてのプロジェクト

Wiki のアップグレード

`trac-admin` は新しいバージョンでは存在しない以前のバージョンの Wiki ページを削除や移動しません。

Trac データベースのアップグレード

既知の問題として、PySqlite のいくつかのバージョン (2.5.2-2.5.4) では、`trac-admin upgrade` スクリプトを使用してデータベースを更新することができません。このエラーを避けるために、`sqlite` の `python` バインディングのバージョンをより新しいものかより古いバージョンを使用することを推奨します。詳細については、[本家チケット 9434](#) を参照して下さい。

Python のアップグレード

Python を新しいバージョンにアップグレードすると Python パッケージの再インストールが必要となります: Trac も例外ではないですし、[easy\\_install](#) も然りです (もし使用しているならばですが)。Subversion を使用しているならば、Subversion の Python のバインディングもアップグレードする必要があります。

## Windows と Python 2.6

もしあなたが、CollabNet の Subversion のパッケージを使用しているならばアンインストールする必要があるかもしれません。というのも、[Algazam](#) に気軽に使える Python バインディングがあるからです。 ([TracSubversion](#) 参照) いいニュースとして、調整なしに動作します。

## データベースの変更

SQLite から PostgreSQL へ

[trac-hacks.org](#) の [sqlite2pg](#) は SQLite のデータベースを PostgreSQL に移行するためのサポートをするスクリプトです。

## より古いバージョン

さらに前のバージョンからのアップグレードについては [wiki:0.10/TracUpgrade#SpecificVersions](#) を最初に参照してください。

---

See also: [TracGuide](#), [TracInstall](#)