Wikiprint Book

Title: アップグレードの説明

Subject: SilverFrost - TracUpgrade

Version: 3

Date: 06/04/26 04:22:53

SilverFrost 目次

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. サーバーをオフラインにする

サーバー起動中にアップデートを行うのはやめてください。パッケージの一部をメモリにキャッシュしているかもしれませんし、コードのアップデートによって内部

2. 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

3. Trac Environment 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

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

4. 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。

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（IE，Opera）は、 CSS や Javascript
ファイルを強引にキャッシュしてしまうので、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ために強制的に更新すること（<F5>）
を十分行う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 0.12 から Trac 1.0 にアップグレードする

サブバージョンサポートの Trac コンポーネントはデフォルト状態では有効にならなくなりました。サブバージョンのサポートを有効にするためには、
tracopt.versioncontrol.svn コンポーネントを有効にしてください。例えば、 TracIni に以下のように記述してください:

```
[components]
tracopt.versioncontrol.svn.* = enabled
```

明示的にサブバージョンのコンポーネントを無効に設定していない場合は、この対応をとり、 TracIni を適切に変更してください。

今回の自動アップグレードで添付ファイルが格納される場所が変わります。心配性な人は、アップグレードの前に attachments
ディレクトリのバックアップを取りたいと思うかもしれません(本当に心配性な人は、すでに environment のフルコピーを取っているでしょうね)。
attachments ディレクトリに添付ファイル 以外
のファイルが格納されていると、新しいレイアウトへの移行の最後のステップで失敗してしまいます:何かファイルやフォルダが格納されていると、今バージョンか
attachments ディレクトリを削除することができません。このエラーは無視してもかまいませんが、environment
をクリーンアップするために、ファイル内容を確認し、別の場所へ移動させ、 attachments
ディレクトリを手動で削除したほうがよいでしょう。添付ファイルは今バージョンから environment 配下の files/attachments
ディレクトリ内に格納されます。

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 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な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

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をする必要があります:

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

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-comit 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, CHANGESET_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。レポートで計算に日付/時刻カラムを使用するような場合 (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れば 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

アップグレードの際、以下のことに留意してください。

カスタマイズされたテンプレート

Trac は Environment の <env>/templates フォルダ内、または [inherit] templates_dir
コンフィグに設定された共通のディレクトリ内にテンプレートのコピーを置くことによって、Genshi
テンプレートのカスタマイズをサポートします。もしこの方法を採用している場合、テンプレートはおそらく今後も進化していくので、新しくリリースされた Trac (以前のバージョンでも)
にアップグレードする際には、新しいテンプレートのコピーを手動で変更することが繰り返し必要になることに注意してください。diff
は取っておいてください ;-)

TracInterfaceCustomization を行うには、適切な ITemplateStreamFilter
変換を行うカスタムプラグインを書く方法が望ましいでしょう。この場合、通常 id の修正や CSS の class
の変更を行わないため、変更が生じても影響をうけません。もし必要になるのであれば、 TracDev/ApiChanges
ページにドキュメントが記載されるでしょう。

ZipImportError

zip形式で圧縮されたパッケージの内部キャッシングにより、ディスク上でパッケージの内容が変わるたびに、メモリ内のzip
インデックスは一致せず、回復不能な ZipImportError が出ます。アップグレードをする前にメンテナンスのために予めサーバーを停止してください。
詳細については、 本家チケット 7014 を参照して下さい。

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。

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

関連するトピック

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、
Alagazam 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 PostgresSQL 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からのアップグレード

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