Upgrade Instructions

Error: Macro TracGuideToc(None) failed

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

Instructions

Typically, there are seven steps involved in upgrading to a newer version of Trac:

1. Update the Trac Code

Get the new version as described in TracInstall, or your operating system specific procedure.

If you already have a 0.11 version of Trac installed via `easy_install`, it might be the easiest to use it also to upgrade you Trac installation.

```
# easy_install --upgrade Trac==0.12b1
```

(once beta1 is released, that is)

If you do a manual (not operating system specific) upgrade, you should also stop any running Trac server before the installation. Doing "hot" upgrades is not advised, especially not on Windows (#7265).

You may also want to remove the pre-existing Trac code by deleting the `trac` directory from the Python `lib/site-packages` directory, or remove Trac .eggs from former versions. The location of the site-packages directory depends on the operating system, and the location in which Python was installed. However, the following locations are common:

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

You may also want to remove the Trac `cgi-bin`, `htdocs`, `templates` and `wiki-default` directories that are commonly found in a directory called `share/trac` (the exact location depends on your platform).

This cleanup is not mandatory, but it makes it easier to troubleshoot issues later on, as you won't waste your time looking at code or templates from a previous release that are not being used anymore... As usual, make a backup before actually deleting things.

2. Upgrade the Trac Environment

Environment upgrades are not necessary for minor version releases unless otherwise noted.

After restarting, Trac should show the instances which need a manual upgrade via the automated upgrade scripts to ease the pain. These scripts are run via trac-admin:

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

This command will do nothing if the environment is already up-to-date.

Note that a backup of your database will be performed automatically prior to the upgrade. This feature is relatively new for the PostgreSQL or MySQL database backends, so if it fails, you will have to backup the database manually. Then, to perform the actual upgrade, run:

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

3. Update the Trac Documentation

Every Trac environment includes a copy of the Trac documentation for the installed version. As you probably want to keep the included documentation in sync with the installed version of Trac, trac-admin provides a command to upgrade the documentation:

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

Note that this procedure will of course leave your `WikiStart` page intact.

4. Resynchronize the Trac Environment Against the Source Code Repository

Each Trac environment must be resynchronized against the source code repository in order to avoid errors such as "No changeset ??? in the repository" while browsing the source through the Trac interface:

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

5. Refresh static resources

If you've setup web server to give out static resources directly (accessed using /chrome/ URL) then you need to refresh them using the same command:

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

this will extract static resources and CGI scripts (trac.wsgi, etc) from new Trac version and its plugins into `/deploy/path`.

Some web browsers cache the CSS and Javascript file in a strong way, so you'll perhaps need to refresh the cache of the browsers by a force reload (and tell your users to do so!).

6. Steps specific to a given Trac version

Upgrading from Trac 0.11 to Trac 0.12

Python 2.3 no longer supported

The minimum supported version of python is now 2.4

SQLite v3.x required

SQLite v2.x is no longer supported, if you happen to still use a Trac database using this format, you'll need to convert it to SQLite v3.x first. See PySqlite#UpgradingSQLitefrom2.xto3.x for details.

Multiple Repository Support

If you plan to add more repositories to your Trac instance, as this is now possible with the newly introduced multiple repository support, please refer to TracRepositoryAdmin#Migration.

This can be of interest even if you only have one repository, as there's now a way to avoid the potentially costly resync check at every request.

Improved repository synchronization

In addition to supporting multiple repositories, a new more efficient method for synchronizing Trac and your repositories was implemented.

While you can keep the same synchronization as in 0.11 adding the post-commit hook as outlined in TracRepositoryAdmin#Synchronization and TracRepositoryAdmin#ExplicitSync will allow more efficient synchronization and is more or less required for multiple repositories.

Note that if you were using the `trac-post-commit-hook`, *you're strongly advised to upgrade it* to the new hook documented in the above references, as the old hook will not work with anything else than the default repository and even for this case, it won't trigger the appropriate notifications.

Upgrading from Trac 0.10 to Trac 0.11

Site Templates and Styles

The templating engine has changed in 0.11 to Genshi, please look at TracInterfaceCustomization for more information.

If you are using custom CSS styles or modified templates in the `templates` directory of the TracEnvironment, you will need to convert them to the Genshi way of doing things. To continue to use your style sheet, follow the instructions at TracInterfaceCustomization#SiteAppearance.

Trac Macros, Plugins

The Trac macros will need to be adapted, as the old-style wiki-macros are not supported anymore (due to the drop of ClearSilver and the HDF); they need to be converted to the new-style macros, see WikiMacros. When they are converted to the new style, they need to be placed into the plugins directory instead and not wiki-macros, which is no longer scanned for macros or plugins.

For FCGI/WSGI/CGI users

For those who run Trac under the CGI environment, run this command in order to obtain the trac.*gi file:

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

This will create a deploy directory with the following two subdirectories: `cgi-bin` and `htdocs`. Then update your Apache configuration file `httpd.conf` with this new `trac.cgi` location and `htdocs` location.

Web Admin plugin integrated

If you had the webadmin plugin installed, you can uninstall it as it is part of the Trac code base since 0.11.

7. Restart the Web Server

If you are not running CGI, reload the new Trac code by restarting your web server.

Known Issues

parent dir

If you use a trac parent env configuration and one of the plugins in one child does not work, none of the children work.

Wiki Upgrade

`trac-admin` will not delete or remove default wiki pages that were present in a previous version but are no longer in the new version.

Changing Database Backend

SQLite to PostgreSQL

The sqlite2pg script on trac-hacks.org has been written to assist in migrating a SQLite database to a PostgreSQL database

Older Versions

For upgrades from versions older than Trac 0.10, refer first to wiki:0.10/TracUpgrade#SpecificVersions.

See also: TracGuide, TracInstall