Table of Contents

The Trac Environment 3
Creating an Environment 3
Database Connection Strings 3
Source Code Repository 4
Directory Structure 4

The Trac Environment

Trac uses a directory structure and a database for storing project data. The directory is referred to as the “environment”.

Creating an Environment

A new Trac environment is created using trac-admin:

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

trac-admin will ask you for the name of the project, the database connection string (explained below), and the type and path to your source code repository.

Note: The web server user will require file system write permission to the environment directory and all the files inside. Please remember to set the appropriate permissions. The same applies to the Subversion repository Trac is eventually using, although Trac will only require read access as long as you're not using the BDB file system.

Database Connection Strings

Since version 0.9, Trac supports both SQLite, PostgreSQL and MySQL as database backends. The default is to use SQLite, which is probably sufficient for most projects. The database file is then stored in the environment directory, and can easily be backed up together with the rest of the environment.

The connection string for an embedded SQLite database is:

```
sqlite:db/trac.db
```

If you want to use PostgreSQL or MySQL instead, you'll have to use a different connection string. For example, to connect to a PostgreSQL database on the same machine called `trac`, that allows access to the user `johndoe` with the password `letmein`, use:

```
postgres://johndoe:letmein@localhost/trac
```

If PostgreSQL is running on a non-standard port (for example 9342), use:

```
postgres://johndoe:letmein@localhost:9342/trac
```

Note that with PostgreSQL you will have to create the database before running `trac-admin initenv`.

And make sure PostgreSQl DB name is “trac”. What worked for me: And didn't work uppercase trac-user-name

```
sudo su - postgres -c createdb trac
sudo su - postgres -c psql trac
CREATE USER trac-user-name WITH PASSWORD 'trac-pass-name';
```

(Just to remind you, if you don't have a sudo/su setup, you just need to do the createdb and psql statements. That threw me the first couple of times I read this.)

Source Code Repository

You'll first have to provide the *type* of your repository (e.g. `svn` for Subversion, which is the default), then the *path* where the repository is located.

If you don't want to use Trac with a source code repository, simply leave the *path* empty (the *type* information doesn't matter, then).

For some systems, it is possible to specify not only the path to the repository, but also a *scope* within the repository. Trac will then only show information related to the files and changesets below that scope. The Subversion backend for Trac supports this; for other types, check the corresponding plugin's documentation.

Example of a configuration for a Subversion repository:

```
[[:trac]]
repository_type = svn
repository_dir = /path/to/your/repository
```

The configuration for a scoped Subversion repository would be:

```
[[:trac]]
repository_type = svn
repository_dir = /path/to/your/repository/scope/within/repos
```

Directory Structure

An environment directory will usually consist of the following files and directories:

- `README` - Brief description of the environment.
- `VERSION` - Contains the environment version identifier.
- `attachments` - Attachments to wiki pages and tickets are stored here.
- `conf`
    - `trac.ini` - Main configuration file. See TracIni.

- `db`
  - `trac.db` - The SQLite database (if you're using SQLite).
- `plugins` - Environment-specific plugins (Python eggs)
- `templates` - Custom environment-specific templates.
  - `site_css.cs` - Custom CSS rules.
  - `site_footer.cs` - Custom page footer.
  - `site_header.cs` - Custom page header.
- `wiki-macros` - Environment-specific Wiki macros.

```
'''Note: don't confuse a Trac environment directory with the source code repository directory.
```

It happens that the above structure is loosely modelled after the Subversion repository directory structure, but they are not and *must not* be located at the same place.'''

See also: TracAdmin, TracBackup, TracIni, TracGuide

From:
https://wiki.linux-ottawa.org/ - **Linux-Ottawa (OCLUG) Wiki**

Permanent link:
https://wiki.linux-ottawa.org/doku.php?id=tracenvironment

Last update: **2015/06/09 19:23**