- Wikiprint Book
- Title: The Trac Environment

Subject: eDokumenty - elektroniczny system obiegu dokumentów, workflow i CRM - TracEnvironment

Version: 1

Date: 06/06/25 20:21:40

# **Table of Contents**

The Trac Environment	3
Creating an Environment	3
Database Connection Strings	3
Embedded SQLite Connection String	3
PostgreSQL Connection String	3
MySQL Connection String	4
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/myproject 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. Also, it seems that project names with spaces can be problematic for authentication (see <u>1#7163</u>).

#### **Database Connection Strings**

Since version 0.9, Trac supports both <u>SQLite</u> and <u>PostgreSQL</u> database backends. Preliminary support for <u>MySQL</u> was added in 0.10. 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 <u>backed up</u> together with the rest of the environment.

### **Embedded SQLite Connection String**

The connection string for an embedded SQLite database is:

sqlite:db/trac.db

#### PostgreSQL Connection String

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

Note that due to the way the above string is parsed, the "/" and "@" characters cannot be part of the password.

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

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

On UNIX, you might want to select a UNIX socket for the transport, either the default socket as defined by the PGHOST environment variable:

postgres://user:password@/database

or a specific one:

postgres://user:password@/database?host=/path/to/socket/dir

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

See the <u>PostgreSQL documentation</u> for detailed instructions on how to administer <u>PostgreSQL</u>. Generally, the following is sufficient to create a database user named tracuser, and a database named trac.

createuser -U postgres -E -P tracuser createdb -U postgres -O tracuser -E UTF8 trac When running createuser you will be prompted for the password for the user 'tracuser'. This new user will not be a superuser, will not be allowed to create other databases and will not be allowed to create other roles. These privileges are not needed to run a trac instance. If no password is desired for the user, simply remove the -P and -E options from the createuser command. Also note that the database should be created as UTF8. LATIN1 encoding causes errors trac's use of unicode in trac. SQL\_ASCII also seems to work.

Under some default configurations (debian) one will have run the createuser and createdb scripts as the postgres user. For example:

```
sudo su - postgres -c 'createuser -U postgres -S -D -R -E -P tracuser'
sudo su - postgres -c 'createdb -U postgres -O tracuser -E UTF8 trac'
```

Trac uses the public schema by default but you can specify a different schema in the connection string:

postgres://user:pass@server/database?schema=yourschemaname

#### MySQL Connection String

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

mysql://johndoe:letmein@localhost:3306/trac

## 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).
- ht docs directory containing web resources, which can be referenced in Genshi templates. (0.11 only)
- log default directory for log files, if logging is turned on and a relative path is given.
- plugins Environment-specific <u>plugins</u> (Python eggs, since <u>0.10</u>)
- templates Custom ClearSilver environment-specific templates. (0.10 only)

- site\_css.cs Custom CSS rules.
- site\_footer.cs Custom page footer.
- site\_header.cs Custom page header.

templates - Custom Genshi environment-specific templates. (0.11 only)

- site.html method to customize header, footer, and style, described in <u>TracInterfaceCustomization#SiteAppearance</u>
- wiki-macros Environment-specific Wiki macros. (0.10 only)

# 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: <u>TracAdmin</u>, <u>TracBackup</u>, <u>TracIni</u>, <u>TracGuide</u>