

Trac Storage - The Environment

Trac uses a directory structure and a database for storing project data.

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 where your subversion repository is located.

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, 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` and `?PostgreSQL` 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 instead, you'll have to use a different connection string. For example, to connect to a 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`.

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.
-

See also: [TracAdmin](#), [TracBackup](#), [TracIni](#), [TracGuide](#)