Table of Contents

Trac Installation Guide 3
Requirements 3
For SQLite 3
For PostgreSQL 4
For MySQL 4
Optional Requirements 4
Installing Trac 5
Advanced Options 5
Creating a Project Environment 6
Running the Standalone Server 6
Running Trac on a Web Server 6
Configuring Authentication 7
Using Trac 7

Trac Installation Guide

[TracGuideToc]

Trac is a lightweight project management tool that is implemented as a web-based application. Trac is written in the Python programming language and needs a database (either SQLite, PostgreSQL, or MySQL works). For HTML rendering, Trac uses the ClearSilver templating system.

What follows are generic instructions for installing and setting up Trac and its requirements. While you can find instructions for installing Trac on specific systems at TracInstallPlatforms on the main Trac site, please be sure to first read through these general instructions to get a good understanding of the tasks involved.

Requirements

To install Trac, the following software packages must be installed:

- Python, version >= 2.3.
  - For RPM-based systems you might also need the `python-devel` and `python-xml` packages.
  - If you want to use Trac with Subversion on Windows, note that there are no precompiled Subversion bindings for Python 2.4, so you probably need Python 2.3.
  - 2006-09-20: This seems to be no longer true, see http://subversion.tigris.org/servlets/ProjectDocumentList?folderID=91. For subversion 1.4.0 python 2.4 binary bindings are available.
- ClearSilver, version >= 0.9.3
  - With python-bindings (`./configure –with-python=/usr/bin/python`)
- You also need a database system and the corresponding python drivers for it.

The database can be either SQLite, PostgreSQL or MySQL *(experimental)*.

For SQLite

- SQLite, version 2.8.x or 3.x (preferred)
- PySQLite, version 1.x (for SQLite 2.x) or version 2.x (for SQLite 3.x). For details see PySqlite

*Note: Versions of Trac prior to 0.9 do **not** work with PySQLite 2.x.*

```
Note: It appears that PySQLite 2.x is required for Trac 0.9+/SQLite 3.x if you
plan to use the 'trac-post-commit-hook.py' script available from the 'contrib'
section of the source repository.
```

Note: Users of Mac OS X please take care; the Apple-supplied SQLite contains additional code to support file locking on network filesystems like AFP or SMB. This is not presently (3.3.6) in the mainline sources,

so if you build your own SQLite from source it will not function correctly on such filesystems - typically it gives the error "database is locked". A patch is available for version 3.3.6, based on Apple's code, otherwise you're probably best off using the Apple supplied version (presently 3.1.3).

For PostgreSQL

- PostgreSQL
- psycopg2 or pyPgSQL

Note: PostgreSQL support requires Trac version 0.9 or later.

For MySQL

Warning: MySQL support is currently experimental. That means it works for some people, but has not been tested extensively yet.

- MySQL, version 4.1 or later
- MySQLdb, version 1.2.1 or later

Note: MySQL support requires Trac version 0.10 or later.

Optional Requirements

Version Control System

- Subversion, version >= 1.0. (either 1.2.3 or >= 1.3.1 recommended) and corresponding Python bindings. For troubleshooting, check TracSubversion
  - Trac uses the SWIG bindings included in the Subversion distribution, **not** PySVN (which is sometimes confused with the standard SWIG bindings).
  - If Subversion was already installed without the SWIG bindings, you'll need to re-`configure` Subversion and `make swig-py`, `make install-swig-py`.
- Support for other version control systems is provided via third-parties. See PluginList and VersioningSystemBackend.

Web Server

- A CGI-capable web server (see TracCgi), or
- a FastCGI-capable web server (see TracFastCgi), or
- Apache with mod_python 3.1.3+ (see TracModPython)
  - When installing mod_python the development versions of Python and Apache are required (actually the libraries and header files)

For those stuck with Apache 1.3, it is also possible to get Trac 0.8.4 working with mod_python 2.7 (see TracModPython2.7). This hasn't been tested with Trac 0.9, so it may or may not work.

Other Python Utilities

- setuptools, version >= 0.6 for using plugins (see TracPlugins)
- docutils, version >= 0.3.9 for WikiRestructuredText.
- SilverCity and/or Enscript for syntax highlighting.
  - Note that SilverCity 0.9.6 has a bug that breaks Python syntax highlighting in Trac. Until an update is made available, we recommend using version 0.9.5.

Attention: The various available versions of these dependencies are not necessarily interchangable, so please pay attention to the version numbers above. If you are having trouble getting Trac to work please double-check all the dependencies before asking for help on the MailingList or IrcChannel.

Please refer to the documentation of these packages to find out how they are best installed. In addition, most of the platform-specific instructions also describe the installation of the dependencies. Keep in mind however that the information there might concern older versions of Trac than the one you're installing (in particular, there are still some pages that are about Trac 0.8).

Installing Trac

Like most Python programs, the Trac Python package is installed by running the following command at the top of the source directory:

```
$ python ./setup.py install
```

```
Note: you'll need root permissions or equivalent for this step.
```

This will byte-compile the python source code and install it in the `site-packages` directory of your Python installation. The directories `cgi-bin`, `templates`, `htdocs`, `wiki-default` and `wiki-macros` are all copied to `$prefix/share/trac/`. `conf` and `plugins` stub directories are also created under `$prefix/share/trac/` (*since 0.10*).

The script will also install the trac-admin command-line tool, used to create and maintain project environments, as well as the tracd standalone server.

Advanced Options

To install Trac to a custom location, or find out about other advanced installation options, run:

```
$ python ./setup.py install --help
```

Also see Installing Python Modules for detailed information.

Specifically, you might be interested in:

```
$ python ./setup.py install --prefix=/path/to/installdir
```

Creating a Project Environment

A Trac environment is the backend storage where Trac stores information like wiki pages, tickets, reports, settings, etc. An environment is basically a directory that contains a human-readable configuration file and various other files and directories.

A new environment is created using trac-admin:

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

trac-admin will prompt you for the information it needs to create the environment, such as the name of the project, the type and the path to an existing source code repository, the database connection string, and so on. If you're not sure what to specify for one of these options, just leave it blank to use the default value. The database connection string in particular will always work as long as you have SQLite installed. Leaving the path to the source code repository empty will disable any functionality related to version control, but you can always add that back when the basic system is running.

Also note that the values you specify here can be changed later by directly editing the TracIni configuration file.

```
Note: The user account under which the web server runs will require write
permissions to the environment directory and all the files inside.
```

Running the Standalone Server

After having created a Trac environment, you can easily try the web interface by running the standalone server tracd:

```
$ tracd --port 8000 /path/to/myproject
```

Then, fire up a browser and visit `http://localhost:8000/`. You should get a simple listing of all environments that tracd knows about. Follow the link to the environment you just created, and you should see Trac in action.

Running Trac on a Web Server

Trac provides three options for connecting to a “real” web server: CGI, FastCGI and mod_python. For decent performance, it is recommended that you use either FastCGI or mod_python.

Configuring Authentication

The process of adding, removing, and configuring user accounts for authentication depends on the specific way you run Trac. The basic procedure is described in the "Adding Authentication" section on the TracCgi page. To learn how to setup authentication for the frontend you're using, please refer to one of the following pages:

- TracStandalone if you use the standalone server, `tracd`.
- TracCgi if you use the CGI or FastCGI methods.
- TracModPython if you use the mod_python method.

Using Trac

Once you have your Trac site up and running, you should be able to browse your subversion repository, create tickets, view the timeline, etc.

Keep in mind that anonymous (not logged in) users can by default access most but not all of the features. You will need to configure authentication and grant additional permissions to authenticated users to see the full set of features.

// Enjoy! //

The Trac Team

See also: TracGuide, TracCgi, TracFastCgi, TracModPython, TracUpgrade, TracPermissions

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

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

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