Trac Documentation

Release 0.12-dev

The Trac Team

May 15, 2008

CONTENTS

Release 0.12

Date May 14, 2008

Contents:

CHAPTER

ONE

Trac Installation Guide

Trac is written in the Python programming language and needs a database, SQLite, Postgres, or MySQL. For HTML rendering, Trac uses the Genshi 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.

1.1 Quick Install

If you already have Python 2.5 and setuptools installed, just run this command to install the latest version of Trac:

```
easy_install Trac
```

After this skip down to creating a project environment.

1.2 Requirements

Python >= 2.3 *(If using mod_python or mod_wsgi, 2.5 is preferred.)*

setuptools >= 0.6

Genshi >= 0.4.1

Database See below

ClearSilver >= 0.9.3 **optional** *(Needed only for using older plugins.)*

You will need database bindings for whichever database you plan to use. We highly recommend using SQLite to begin with, and then move to Postgres if you run in to problems.

1.2.1 SQLite

Python 2.5 ships with compatible a version of the bindings included, so if you use it, stop reading here.

SQLite >= 3.3.4

PySQLite >= 2.3.2

It is possible to use the older SQLite 2.x and PySQLite 1.1.x, however you may run in to compatibility problems with some plugins and scripts. In addition you will hit many throughput and locking issues.

1.2.2 Postgres

Postgres >= 8.0 *(>=8.3 requires Trac >=0.11.)*

pyscopg2 Any version

or

pyPgSQL Any version

pyscopg2 is generally faster, and is preferred.

1.2.3 MySQL

Warning: MySQL has several issues that cannot be easily worked around by Trac. As such, it should only be used if there is no other option.

MySQL >= 4.1

MySQLdb >= 1.2.1

1.2.4 Subversion

Using Subversion with Trac is optional, however if you wish to, you will need to install the Subversion Python bindings.

Subversion >= 1.0

Note: You do **NOT** need to install SWIG in order to install the Python bindings.

The SWIG bindings should not be confused with PySVN, which is an unrelated project.

For other version control backends, please see VersioningSystemBackend

1.2.5 Syntax Coloring

Trac optionally supports code syntax coloring in the source browser, and in wiki text. While Pygments is the preferred back-end for this you can install any or all of the following:

Pygments >= 0.6

SilverCity >= 0.9.7, 0.9.5 *(0.9.6 is not compatible.)*

Enscript Any version

1.2.6 Other Libraries

These Python libraries are all optional.

docutils >= 0.3 *(Needed for rendering reStructuredText_)*

pytz Any version *(Needed for displaying a full list of timezones, without it a smaller, internal list will be used.)*

1.3 Installation

easy_install Trac is the preferred method of installing Trac, but you can also use the more traditional style from a source bundle:

```
python setup.py install
```

To install in a non-standard path, use the `--prefix` option.

See Installing Python Modules for full instructions on install Python modules.

1.4 Creating a Project Environment

A *Trac environment* is the back-end 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 *trac.ini* configuration file.

Common paths used for the Trac environment are `/var/trac` and `/srv/trac`.

1.5 Web Server

Trac offers much flexibility with its web server options. If you are not already running a server, the standalone tracd option is generally sufficient for most projects.

1.5.1 Trac on CGI

> **Warning:** The way CGI works requires that the entire Trac application be reloaded on every request. This ends up being very slow. **Use CGI only as a last resort.**

To generate the `trac.cgi` script run:

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

It will be in the `cgi-bin` folder inside the path given.

Apache

Add an alias for the path you want to run Trac at to your VirtualHost:

```
ScriptAlias /trac /path/to/www/trac/cgi-bin/trac.cgi
```

For a multiple environment configuration you can use:

```
ScriptAlias /trac /path/to/www/trac/cgi-bin/trac.cgi
<Location /trac>
    SetEnv TRAC_ENV_PARENT_DIR /path/to/base
</Location>
```

See *Authentication on Apache* to setup authentication.

1.5.2 Trac on FastCGI

To generate the `trac.fcgi` script run:

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

It will be in the `cgi-bin` folder inside the path given.

Apache

Add an alias for the path you want to run Trac at to your VirtualHost:

```
ScriptAlias /trac /path/to/www/trac/cgi-bin/trac.fcgi/
```

Note: The trailing slash after the `trac.fcgi` is important.

For a multiple environment configuration you will need to edit the `trac.fcgi` script. Just set **TRAC_ENV_PARENT_DIR** instead of **TRAC_ENV**, and alter the path accordingly.

See *Authentication on Apache* to setup authentication.

LigHTTPD

An example map for using FastCGI on Lighty:

```
fastcgi.server = ("/trac" =>
                   ("trac" =>
                     ("socket" => "/tmp/trac-fastcgi.sock",
                      "bin-path" => "/path/to/www/trac/cgi-bin/trac.fcgi",
                      "check-local" => "disable",
                      "bin-environment" =>
                        ("TRAC_ENV" => "/path/to/env")
                     )
                   )
                 )
```

Note: Be sure you do not have a trailing slash on the path you use to serve Trac. If you want to serve Trac from `/`, use the following FastCGI script instead of the generated one:

```
#!/usr/bin/env python
import tempfile
try:
    from flup.server.fcgi import WSGIServer
except ImportError:
    from trac.web._fcgi import WSGIServer
from trac.web.main import dispatch_request

def application(environ, start_request):
    environ['PATH_INFO'] = environ['SCRIPT_NAME'] + environ['PATH_INFO']
    environ['SCRIPT_NAME'] = ''
    environ['PYTHON_EGG_CACHE'] = tempfile.gettempdir()
    return dispatch_request(environ, start_request)

if __name__ == '__main__':
    WSGIServer(application).run()
```

Authentication

First generate your `trac.htpasswd` file as shown in *Basic Authentication*.

Be sure you are loading `mod_auth` before `mod_fastcgi` in your modules list.

You need to configure both the back-end file and the paths to enforce authentication on:

```
auth.backend = "htpasswd"

# Separated password files for each project
# See "Conditional Configuration"
$HTTP["url"] =~ "^/trac" {
  auth.backend.htpasswd.userfile = "/path/to/trac.htpasswd"
}

# Enable auth on trac URLs, see
auth.require = ("/trac/login" =>
                ("method"  => "basic",
                 "realm"   => "Trac Login",
                 "require" => "valid-user"
                )
               )
```

See Also:

mod_fastcgi mod_fastcgi documentation.

mod_auth mod_auth documentation.

nginx

nginx handles FastCGI slightly differently, as it will not spawn the daemon program itself. You need to start the FastCGI daemon on its own, and then point nginx at it.

An example nginx configuration:

```
location /trac {
    # full path
    if ($uri ~ ^/trac/([^/]+)(/.*)) {
         set $script_name $1;
         set $path_info $2;
    }

    # socket address
    fastcgi_pass   unix:/tmp/trac-fastcgi.sock;

    ## WSGI REQUIRED VARIABLES
    # WSGI application name - trac instance prefix.
    fastcgi_param  SCRIPT_NAME        $script_name;
    fastcgi_param  PATH_INFO          $path_info;

    ## WSGI NEEDED VARIABLES - trac warns about them
    fastcgi_param  REQUEST_METHOD     $request_method;
    fastcgi_param  SERVER_NAME        $server_name;
    fastcgi_param  SERVER_PORT        $server_port;
    fastcgi_param  SERVER_PROTOCOL    $server_protocol;

    # for authentication to work
    fastcgi_param  REMOTE_USER        $remote_user;
}
```

And a modified `trac.fcgi` script:

```
#!/usr/bin/env python
import os
import tempfile

sockaddr = '/tmp/trac-fastcgi.sock'
os.environ['TRAC_ENV'] = '/path/to/env'
os.environ['PYTHON_EGG_CACHE'] = tempfile.gettempdir()

try:
     from trac.web.main import dispatch_request
     import trac.web._fcgi

     fcgiserv = trac.web._fcgi.WSGIServer(dispatch_request,
          bindAddress = sockaddr, umask = 7)
     fcgiserv.run()

except SystemExit:
    raise
except Exception, e:
    print 'Content-Type: text/plain\r\n\r\n',
    print 'Oops...'
    print
    print 'Trac detected an internal error:'
    print
    print e
    print
    import traceback
    import StringIO
    tb = StringIO.StringIO()
    traceback.print_exc(file=tb)
```

```
    print tb.getvalue()
```

Authentication

To add authentication, first setup the `trac.htpasswd` file as shown in *Basic Authentication*.

Then add the following in the `location`:

```
auth_basic "Trac Login";
auth_basic_user_file /path/to/trac.htpasswd;
```

See Also:

ngx_http_fastcgi_module Documentation for the `fastcgi_*` configuration options.

1.5.3 Trac on mod_python

Apache

First create a handler for Trac:

```
<Location /trac>
  SetHandler mod_python
  PythonHandler trac.web.modpython_frontend
  PythonInterpreter main
  PythonOption TracEnv /path/to/env
  PythonOption TracUriRoot /trac
  SetEnv PYTHON_EGG_CACHE /tmp
</Location>
```

`PythonInterpreter` needs to be set to the same string in all VirtualHosts using Trac, though the actual value is unimportant. '`PythonOption TracUriRoot`' needs to be set to the same path as in the Location.

For a multiple environment configuration you can use '`PythonOption TracEnvParentDir`'.

See *Authentication on Apache* to setup authentication.

Changing the Python path

If Trac, or other modules, are not installed in the standard path, you can use the `PythonPath` option to add additional folders:

```
PythonPath "['/new/path'] + sys.path"
```

See Also:

Subversion bindings from source Add `/usr/lib/svn-python`.

virtualenv Add `/path/to/virtualenv/lib/python2.X/site-packages`.

Example

A full example of a mod_python and mod_dav_svn configuration:

```
<VirtualHost *:80>
    ServerName example.com
    ServerAlias www.example.com
    ServerAdmin webmaster@example.com

    # Note: This folder should exist, but will generally be empty
    DocumentRoot /srv/example.com/htdocs
    <Directory /srv/example.com/htdocs>
        Order allow,deny
        Allow from all
    </Directory>

    # Host the main Trac instance at /
    <Location />
        SetHandler mod_python
        PythonHandler trac.web.modpython_frontend
        PythonInterpreter main
        PythonOption TracEnv /srv/example.com/tracs/main
        PythonOption TracUriRoot /
        SetEnv PYTHON_EGG_CACHE /tmp
    </Location>

    # Host all others at /projects/$PROJECT
    <Location /projects>
        PythonOption TracEnv ""
        PythonOption TracEnvParentDir /srv/example.com/tracs
    </Location>

    # Handle logins on both /login and /projects/$PROJECT/login
    <LocationMatch ^(/projects/[^/]+)?/login>
        AuthType Basic
        AuthName "example.com Login"
        AuthUserFile /srv/example.com/htpassd
        Require valid-user
    </LocationMatch>

    # Host subversion for all projects at /svn
    <Location /svn>
        DAV svn
        SVNParentPath /srv/example.com/repos
        SVNListParentPath on

        AuthType Basic
        AuthName "example.com Login"
        AuthUserFile /srv/example.com/htpassd
        # Allow anonymous checkout
        <LimitExcept GET PROPFIND OPTIONS REPORT>
            Require valid-user
        </LimitExcept>
    </Location>
</VirtualHost>
```

Troubleshooting

In general, if you get server error pages, you can either check the Apache error log, or enable the `PythonDebug` option:

```
<Location /trac>
    ...
    PythonDebug on
</Location>
```

Expat-related segmentation faults This problem will most certainly hit you on Unix when using Python 2.4. In Python 2.4, some version of Expat (an XML parser library written in C) is used, and if Apache is using another version, this results in segmentation faults. As Trac 0.11 is using Genshi, which will indirectly use Expat, that problem can now hit you even if everything was working fine before with Trac 0.10.

See Graham Dumpleton's detailed explanation and workarounds for the issue.

Form submission problems If you're experiencing problems submitting some of the forms in Trac (a common problem is that you get redirected to the start page after submission), check whether your `DocumentRoot` contains a folder or file with the same path that you mapped the mod_python handler to. For some reason, mod_python gets confused when it is mapped to a location that also matches a static resource.

Problem with virtual host configuration If the <Location /> directive is used, setting the `DocumentRoot` may result in a "403 (Forbidden)" error. Either remove the `DocumentRoot` directive, or make sure that accessing the directory it points is allowed (in a corresponding `<Directory>` block).

Using <Location /> together with `SetHandler` resulted in having everything handled by mod_python, which leads to not being able download any CSS or images/icons. I used <Location /trac> '`SetHandler None`' </Location> to circumvent the problem, though I do not know if this is the most elegant solution.

Using .htaccess Although it may seem trivial to rewrite the above configuration as a directory in your document root with a `.htaccess` file, this does not work. Apache will append a "/" to any Trac URLs, which interferes with its correct operation.

It may be possible to work around this with mod_rewrite, but I failed to get this working. In all, it is more hassle than it is worth. Stick to the provided instructions. :)

Win32 Issues If you run trac with mod_python < 3.2 on Windows, uploading attachments will not work. This problem is resolved in mod_python 3.1.4 or later, so please upgrade mod_python to fix this.

OS X issues When using mod_python on OS X you will not be able to restart Apache using **apachectl restart**. This is apparently fixed in mod_python 3.2, but there's also a patch available for earlier versions here.

SELinux issues If Trac reports something like: ''Cannot get shared lock on db.lock" The security context on the repository may need to be set:

```
chcon -R -h -t httpd_sys_content_t PATH_TO_REPOSITORY
```

See Also:

Subversion FAQ How do I set repository permissions correctly?

FreeBSD issues Pay attention to the version of the installed mod_python and sqlite packages. Ports have both the new and old ones, but earlier versions of pysqlite and mod_python won't integrate as the former requires threaded support in python, and the latter requires a threadless install.

If you compiled and installed apache2, apache wouldn't support threads (cause it doesn't work very well on FreeBSD). You could force thread support when running ./configure for apache, using –enable-threads, but this isn't recommended. The best option seems to be adding to `/usr/local/apache2/bin/ennvars`:

```
export LD_PRELOAD=/usr/lib/libc_r.so
```

Subversion issues If you get the following Trac Error '`Unsupported version control system "svn"`' only under mod_python, though it works well on the command-line and even with TracStandalone, chances are that you forgot to add the path to the Python bindings with the *PythonPath* directive. (The better way is to add a link to the bindings in the Python `site-packages` directory, or create a `.pth` file in that directory.)

If this is not the case, it's possible that you're using Subversion libraries that are binary incompatible with the apache ones (an incompatibility of the `apr` libraries is usually the cause). In that case, you also won't be able to use the svn modules for Apache (`mod_dav_svn`).

Segmentation fault with php5-mhash or other php5 modules You may encounter segfaults (reported on debian etch) if php5-mhash module is installed. Try to remove it to see if this solves the problem. See debian bug report 411487.

Some people also have troubles when using php5 compiled with its own 3rd party libraries instead of system libraries. Check here.

1.5.4 Authentication on Apache

Basic Authentication

Create the htpasswd file using the program of the same name:

```
htpasswd -c trac.htpasswd $USERNAME
```

Then add the following to your VirtualHost:

```
<Location /trac/login>
    AuthType Basic
    AuthName "Trac Login"
    AuthUserFile /path/to/trac.htpasswd
    Require valid-user
</Location>
```

The `AuthName` can be set to whatever you like, and will shown to the user in the authentication dialog in their browser.

In a multiple environment setup, you can use the following to use the same authentication on all environments:

```
<LocationMatch /trac/[^/]+/login>
    AuthType Basic
    AuthName "Trac Login"
    AuthUserFile /path/to/htpasswd
    Require valid-user
</LocationMatch>
```

See Also:

Authentication, Authorization and Access Control Apache guide to setting up authentication.

mod_auth_basic Documentation for mod_auth_basic.

Digest Authentication

Create the htdigest file as with basic:

```
htdigest -c trac.htdigest realm $USERNAME
```

`realm` needs to match the value of `AuthName` used in the configuration.

Then add the following to your VirtualHost:

```
<Location /trac/login>
    AuthType Digest
    AuthName "realm"
    AuthDigestFile /path/to/trac.htdigest
    Require valid-user
</Location>
```

You can use the same `LocationMatch` as above for multiple environments.

See Also:

mod_auth_digest Documentation for mod_auth_digest.

LDAP Authentication

You can use `mod_authnz_ldap` to authenticate against an LDAP directory.

Add the following to your VirtualHost:

```
<Location /trac/login>
    AuthType Basic
    AuthName "Trac Login"
    AuthBasicProvider ldap
    AuthLDAPURL "ldap://127.0.0.1/dc=example,dc=com?uid?sub?(objectClass=inetOrgPerson)"
    AuthzLDAPAuthoritative Off
    Require valid-user
</Location>
```

You can also require the user be a member of a certain LDAP group, instead of just having a valid login:

```
Require ldap-group CN=Trac Users,CN=Users,DC=example,DC=com
```

Windows Active Directory

You can use LDAP as a way to authenticate to a AD server.

Use the following as your LDAP URL:

```
AuthLDAPURL "ldap://directory.example.com:3268/DC=example,DC=com?sAMAccountName?sub?(objectClass=user
```

You will also need to provide an account for Apache to use when checking credentials. As this password will be listed in plaintext in the config, you should be sure to use an account specifically for this task:

```
AuthLDAPBindDN ldap-auth-user@example.com
AuthLDAPBindPassword "password"
```

See Also:

mod_authnz_ldap Documentation for mod_authnz_ldap.

mod_ldap Documentation for mod_ldap, which provides connection pooling and a shared cache.

LdapPlugin Store *Trac permissions* in LDAP.

SSPI Authentication

If you are using Apache on Windows, you can use mod_auth_sspi to provide single-sign-on. Download the module from its webpage and then add the following to your VirtualHost:

```
<Location /trac/login>
    AuthType SSPI
    AuthName "Trac Login"
    SSPIAuth On
    SSPIAuthoritative On
    SSPIDomain MyLocalDomain
    SSPIOfferBasic On
    SSPIOmitDomain Off
    SSPIBasicPreferred On
    Require valid-user
</Location>
```

Using the above, usernames in Trac will be of the form `DOMAIN\username`, so you may have to re-add permissions and such. If you do not want the domain to be part of the username, set '`SSPIOmitDomain On`' instead.

Note: Version 1.0.2 and earlier of mod_auth_sspi do not support SSPIOmitDomain and have bug in basic authentication. >= 1.0.3 is recommended.

See Also:

mod_auth_sspi Apache 2.x SSPI authentication module.

Some common problems with SSPI authentication #1055, #1168, #3338

1.6 Further Configuration

Once you have your Trac site up and running, you should be able to view the wiki, browse your subversion repository, 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

CHAPTER

TWO

Indices and tables

- *Index*
- *Module Index*
- *Search Page*

INDEX