# Trac and mod\_python

Mod\_python is an <u>Apache</u> module that embeds the Python interpreter within the server, so that web-based applications in Python will run many times faster than traditional CGI and will have the ability to retain database connections. Trac supports <u>mod\_python</u>, which speeds up Trac's response times considerably, especially compared to <u>CGI</u>, and permits use of many Apache features not possible with <u>tracd/mod\_proxy</u>.

## **Table of Contents**

Trac and mod_python	1
Simple configuration: single project	2
Python Egg Cache	3
Configuring Authentication	3
Advanced Configuration	3
Setting the Python Egg Cache	3
Setting the PythonPath	3
Setting up multiple projects	3
Virtual Host Configuration	4
Troubleshooting	4
Login Not Working	5
Expat-related segmentation faults	5
Form submission problems	5
Problem with virtual host configuration	5
Problem with zipped egg	5
Using .htaccess	6
Additional .htaccess help	6
Platform specific issues	6
Win32 Issues	6
OS X issues	6
SELinux issues	6
FreeBSD issues	6
Fedora 7 Issues	7
Subversion issues	7
Page layout issues	7
HTTPS issues	7
Segmentation fault with php5-mhash or other php5 modules	7

### Simple configuration: single project

If you just installed mod\_python, you may have to add a line to load the module in the Apache configuration:

```
LoadModule python_module modules/mod_python.so
```

Note: The exact path to the module depends on how the HTTPD installation is laid out.

On Debian using apt-get:

```
apt-get install libapache2-mod-python libapache2-mod-python-doc
```

Still on Debian, after you have installed mod\_python, you must enable the modules in apache2, equivalent to the above Load Module directive:

a2enmod python

On Fedora use, using yum:

```
yum install mod_python
```

You can test your mod\_python installation by adding the following to your httpd.conf. You should remove this when you are done testing for security reasons. Note: mod\_python.testhandler is only available in mod\_python 3.2+.

```
<Location /mpinfo>
SetHandler mod_python
PythonInterpreter main_interpreter
PythonHandler mod_python.testhandler
# For Apache 2.2
<IfModule !mod_authz_core.c>
   Order allow,deny
   Allow from all
</IfModule>
# For Apache 2.4
<IfModule mod_authz_core.c>
   Require all granted
</IfModule>
</Location>
```

A simple setup of Trac on mod\_python looks like this:

```
<Location /projects/myproject>
SetHandler mod_python
PythonInterpreter main_interpreter
PythonHandler trac.web.modpython_frontend
PythonOption TracEnv /var/trac/myproject
PythonOption TracUriRoot /projects/myproject
# For Apache 2.2
<IfModule !mod_authz_core.c>
  Order allow, deny
  Allow from all
</IfModule>
# For Apache 2.4
<IfModule mod_authz_core.c>
  Require all granted
</IfModule>
</Location>
```

The option TracUriRoot may or may not be necessary in your setup. Try your configuration without it; if the URLs produced by Trac look wrong, if Trac does not seem to recognize URLs correctly, or you get an odd "No handler matched request to..." error, add the TracUriRoot option. You will notice that the Location and TracUriRoot have the same path.

The options available are:

```
# For a single project
PythonOption TracEnv /var/trac/myproject

# For multiple projects
PythonOption TracEnvParentDir /var/trac/myprojects

# For the index of multiple projects
PythonOption TracEnvIndexTemplate /srv/www/htdocs/trac/project_list_template.html

# A space delimitted list, with a "," between key and value pairs.
PythonOption TracTemplateVars key1,val1 key2,val2

# Useful to get the date in the wanted order
PythonOption TracLocale en_GB.UTF8

# See description above
PythonOption TracUriRoot /projects/myproject
```

### **Python Egg Cache**

Compressed Python eggs like Genshi are normally extracted into a directory named .python-eggs in the users home directory. Since Apache's home usually is not writeable, an alternate egg cache directory can be specified like this:

```
PythonOption PYTHON_EGG_CACHE /var/trac/myprojects/egg-cache
```

Or you can uncompress the Genshi egg to resolve problems extracting from it.

#### **Configuring Authentication**

See corresponding section in the TracModWSGI page.

### **Advanced Configuration**

## **Setting the Python Egg Cache**

If the Egg Cache isn't writeable by your Web server, you'll either have to change the permissions, or point Python to a location where Apache can write. This can manifest itself as a 500 internal server error and/or a complaint in the syslog.

```
<Location /projects/myproject>
...
PythonOption PYTHON_EGG_CACHE /tmp
...
</Location>
```

### Setting the PythonPath

If the Trac installation isn't installed in your Python path, you will have to tell Apache where to find the Trac mod\_python handler using the PythonPath directive:

```
<Location /projects/myproject>
...
PythonPath "sys.path + ['/path/to/trac']"
...
</Location>
```

Be careful about using the PythonPath directive, and *not* SetEnv PYTHONPATH, as the latter won't work.

### Setting up multiple projects

The Trac mod\_python handler supports a configuration option similar to Subversion's SvnParentPath, called TracEnvParentDir:

```
<Location /projects>
SetHandler mod_python
```

```
PythonInterpreter main_interpreter

PythonHandler trac.web.modpython_frontend

PythonOption TracEnvParentDir /var/trac

PythonOption TracUriRoot /projects

</Location>
```

When you request the /projects URL, you will get a listing of all subdirectories of the directory you set as TracEnvParentDir that look like Trac environment directories. Selecting any project in the list will bring you to the corresponding Trac environment.

If you don't want to have the subdirectory listing as your projects home page you can use a

```
<LocationMatch "/.+/">
```

This will instruct Apache to use mod\_python for all locations different from root while having the possibility of placing a custom home page for root in your DocumentRoot folder.

You can also use the same authentication realm for all of the projects using a <LocationMatch> directive:

```
<LocationMatch "/projects/[^/]+/login">
AuthType Basic
AuthName "Trac"
AuthUserFile /var/trac/.htpasswd
Require valid-user
</LocationMatch>
```

### **Virtual Host Configuration**

Below is the sample configuration required to set up your Trac as a virtual server, ie when you access it at the URLs like

http://trac.mycompany.com:

```
<VirtualHost *>
  DocumentRoot /var/www/myproject
  ServerName trac.mycompany.com
   <Location />
      SetHandler mod_python
      PythonInterpreter main_interpreter
      PythonHandler trac.web.modpython_frontend
      PythonOption TracEnv /var/trac/myproject
      PythonOption TracUriRoot /
   </Location>
   <Location /login>
      AuthType Basic
      AuthName "MyCompany Trac Server"
      AuthUserFile /var/trac/myproject/.htpasswd
      Require valid-user
   </Location>
</VirtualHost>
```

This does not seem to work in all cases. What you can do if it does not:

- Try using <LocationMatch> instead of <Location>.
- <Location /> may, in your server setup, refer to the complete host instead of simple the root of the server. This means that everything (including
  the login directory referenced below) will be sent to Python and authentication does not work, ie you get the infamous Authentication information
  missing error. If this is the case, try using a sub-directory for Trac instead of the root, ie /web/ and /web/login instead of / and /login.
- Depending on apache's NameVirtualHost configuration, you may need to use <VirtualHost \*:80> instead of <VirtualHost \*>.

For a virtual host that supports multiple projects replace TracEnv /var/trac/myproject With TracEnvParentDir /var/trac.

**Note**: DocumentRoot should not point to your Trac project env. As Asmodai wrote on #trac: "suppose there's a webserver bug that allows disclosure of DocumentRoot they could then leech the entire Trac environment".

## **Troubleshooting**

If you get server error pages, you can either check the Apache error log, or enable the PythonDebug option:

```
<Location /projects/myproject>
...
PythonDebug on
</Location>
```

For multiple projects, try restarting the server as well.

#### **Login Not Working**

If you've used <Location /> directive, it will override any other directives, as well as <Location /login>. The workaround is to use negation expression as follows (for multi project setups):

```
#this one for other pages
<Location ~ "/*(?!login)">
 SetHandler mod python
 PythonHandler trac.web.modpython_frontend
 PythonOption TracEnvParentDir /projects
 PythonOption TracUriRoot /
</Location>
#this one for login page
<Location ~ "/[^/]+/login">
 SetHandler mod_python
 PythonHandler trac.web.modpython frontend
 PythonOption TracEnvParentDir /projects
 PythonOption TracUriRoot /
 #remove these if you don't want to force SSL
 RewriteEngine On
 RewriteCond %{HTTPS} off
 RewriteRule (.*) https://%{HTTP_HOST}%{REQUEST_URI}
 AuthType Basic
 AuthName "Trac"
 AuthUserFile /projects/.htpasswd
 Require valid-user
</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. This problem has not been reported for Python 2.5+, so best to upgrade.

### 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 <code>DocumentRoot</code> 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 to download any CSS or images/icons. Use <Location /trac> SetHandler None </Location> to circumvent the problem, though this may not be the most elegant solution.

## Problem with zipped egg

It's possible that your version of mod\_python will not import modules from zipped eggs. If you encounter an ImportError: No module named trac in your Apache logs but you think everything is where it should be, this might be your problem. Look in your site-packages directory; if the Trac module appears as a *file* rather than a *directory*, then this might be your problem. To rectify this, try installing Trac using the --always-unzip option:

```
easy_install --always-unzip Trac-0.12b1.zip
```

#### 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.

This also works out-of-box, with following trivial config:

```
SetHandler mod_python

PythonInterpreter main_interpreter

PythonHandler trac.web.modpython_frontend

PythonOption TracEnv /system/path/to/this/directory

PythonOption TracUriRoot /path/on/apache
```

AuthType Basic
AuthName "ProjectName"
AuthUserFile /path/to/.htpasswd
Require valid-user

The TracUriRoot is obviously the path you need to enter to the browser to get to Trac, eg domain.tld/projects/trac.

#### Additional .htaccess help

If you are using the .htaccess method you may have additional problems if your Trac directory is inheriting .htaccess directives from another. This may also help to add to your .htaccess file:

```
<IfModule mod_rewrite.c>
RewriteEngine Off
</IfModule>
```

### Platform specific issues

#### 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, so please upgrade mod\_python to fix this.

### SELinux issues

If Trac reports something like: Cannot get shared lock on db.lock, then the security context on the repository may need to be set:

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

See also ■How do I set repository permissions correctly?

## FreeBSD issues

The FreeBSD ports have both the new and old versions of mod\_python and SQLite, but earlier versions of pysglite and mod\_python won't integrate:

- pysqlite requires threaded support in Python
- mod\_python requires a threadless install.

Apache2 does not automatically support threads 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 the line:

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

#### Fedora 7 Issues

Make sure you install the 'python-sqlite2' package as it seems to be required for TracModPython, but not for tracd.

#### 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 <u>TracStandalone</u>, chances are that you forgot to add the path to the Python bindings with the <u>PythonPath</u> directive. A 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 are using Subversion libraries that are binary incompatible with the Apache ones and 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).

You also need a recent version of mod\_python in order to avoid a runtime error (argument number 2: a 'apr\_pool\_t \*' is expected) due to the default usage of multiple sub-interpreters. Version 3.2.8 should work, though it's probably better to use the workaround described in #3371, in order to force the use of the main interpreter:

```
PythonInterpreter main_interpreter
```

This is also the recommended workaround for other issues seen when using the Python bindings for Subversion within mod\_python (<u>#2611</u>, <u>#3455</u>). See in particular Graham Dumpleton's comment in <u>#3455</u> explaining the issue.

#### Page layout issues

If the formatting of the Trac pages look weird, chances are that the style sheets governing the page layout are not handled properly by the web server. Try adding the following lines to your Apache configuration:

```
Alias /myproject/css "/usr/share/trac/htdocs/css"
<Location /myproject/css>
    SetHandler None
</Location>
```

Note: For the above configuration to have any effect it must be put after the configuration of your project root location /myproject />.

**Note:** Do not enable python optimizations using the directive PythonOptimize On. When optimizations are enabled the page header/footer and documentation for macros and plugins will be hidden. An error will be raised in Trac 1.0.11 and later when optimizations are enabled.

### **HTTPS** issues

If you want to run Trac fully under https you might find that it tries to redirect to plain http. In this case just add the following line to your Apache configuration:

```
<VirtualHost *>
  DocumentRoot /var/www/myproject
  ServerName trac.mycompany.com
  SetEnv HTTPS 1
   ....
</VirtualHost>
```

### 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.

Some people also have troubles when using PHP5 compiled with its own third party libraries instead of system libraries. Check Django segmentation fault.

See also:  $\underline{TracGuide}$ ,  $\underline{TracInstall}$ ,  $\underline{ModWSGI}$ ,  $\underline{FastCGI}$ ,  $\underline{\blacksquare TracNginxRecipe}$