Table of Contents

Trac and mod_python 3
Simple configuration 3
Configuring Authentication 3
Setting the PythonPath 4
Setting up multiple projects 4
Virtual Host Configuration 5
Troubleshooting 5
Form submission problems 6
Problem with virtual host configuration 6
Using .htaccess 6
Win32 Issues 6
OS X issues 6
SELinux issues 6
FreeBSD issues 7
Subversion issues 7

Trac and mod_python

[TracGuideToc]

Trac supports mod_python, which speeds up Trac's response times considerably, especially compared to CGI, and permits use of many Apache features not possible with tracd/mod_proxy.

Simple configuration

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.

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.

```
<Location /mpinfo>
   SetHandler mod_python
   PythonHandler mod_python.testhandler
</Location>
```

A simple setup of Trac on mod_python looks like this:

```
<Location /projects/myproject>
   SetHandler mod_python
   PythonHandler trac.web.modpython_frontend
   PythonOption TracEnv /var/trac/myproject
   PythonOption TracUriRoot /projects/myproject
</Location>
```

The option `**TracUriRoot**` may or may not be necessary in your setup. Try your configuration with out it, and if the URLs produced by Trac look wrong or if Trac does not seem to recognize the URLs correctly, add the `**TracUriRoot**` option. You will notice that the `Location` and `**TracUriRoot**` have the same path.

Configuring Authentication

Configuring authentication works just like for CGI:

```
<Location "/projects/myproject/login">
  AuthType Basic
  AuthName "myproject"
  AuthUserFile /var/trac/myproject/.htpasswd
  Require valid-user
</Location>
```

Setting the PythonPath

If the Trac installation isn't installed in your Python path, you'll 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
  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 (i.e. when you access it at the URLs like http://trac.mycompany.com):

```
<VirtualHost * >
    DocumentRoot /var/trac/myproject
    ServerName trac.mycompany.com
    <Location />
        SetHandler mod_python
        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>
```

For a virtual host that supports multiple projects replace “`TracEnv`” /var/trac/myproject with “`TracEnvParentDir`” /var/trac/

Troubleshooting

In general, 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>
```

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 reposperms]

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.

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`).

See also TracGuide, TracInstall, TracCgi, TracFastCgi

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

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

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