Trac and mod_python

Trac supports ?mod_python, which speeds up Trac's response times considerably 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
```

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>
```

Note that the option `TracUriRoot` may or may not be necessary in your setup. Try without first, and if the URLs produced by Trac look wrong or if Trac does not seem to recognize the URLs correctly, add the `TracUriRoot` option.

Configuring authentication works the same as for CGI:

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

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>
```

Setting up multiple projects

The Trac mod_python handler 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`. 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 "/[^/]+/login">
  AuthType Basic
  AuthName "Trac"
  AuthUserFile /var/trac/.htaccess
  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
    <Directory />
        SetHandler mod_python
        PythonHandler trac.web.modpython_frontend
        PythonOption TracEnv /var/trac/myproject
        PythonOption TracUriRoot /
    </Directory>
    <Location /login>
        AuthType Basic
        AuthName "MyCompany Trac Server"
        AuthUserFile /var/trac/myproject/.htusers
        Require valid-user
    </Location>
</VirtualHost>
```

Troubleshooting

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.

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 (3.1.3 or 3.1.4) on Windows, uploading attachments will **not** work. This is a known problem which we can't solve cleanly at the Trac level.

However, there is a workaround for this at the mod_python level, which is to apply the following patch ?attachment:ticket:554:util_py.patch to the (Lib/site-packages)/modpython/util.py file.

If you don't have the `patch` command, that file can be replaced with the ?fixed util.py (fix which, although done prior to the 3.1.4 release, is *not* present in 3.1.4).

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.

See also TracGuide, TracInstall, TracCgi, TracFastCgi