Table of Contents

Trac Plugins 3
Requirements 3
Installing a Trac Plugin 3
For a Single Project 3
For All Projects 4
Setting up the Plugin Cache 5
About hook scripts 5
Troubleshooting 6
Is setuptools properly installed? 6
Did you get the correct version of the Python egg? 6
Is the plugin enabled? 6
Check the permissions on the egg file 6
Check the log files 6

Trac Plugins

[TracGuideToc]

Since version 0.9, Trac supports plugins that extend the built-in functionality. The plugin functionality is based on the component architecture.

Requirements

To use egg based plugins in Trac, you need to have setuptools (version 0.6) installed.

Plugins can also consist of a single `.py` file dropped into either the environment or global `plugins` directory *(since 0.10)*.

To install `setuptools`, download the bootstrap module ez_setup.py and execute it as follows:

```
$ python ez_setup.py
```

If the `ez_setup.py` script fails to install the setuptools release, you can download it from PyPI and install it manually.

Installing a Trac Plugin

For a Single Project

Plugins are packaged as Python eggs. That means they are ZIP archives with the file extension `.egg`. If you have downloaded a source distribution of a plugin, you can run:

```
$ python setup.py bdist_egg
```

to build the `.egg` file.

Once you have the plugin archive, you need to copy it into the `plugins` directory of the project environment. Also, make sure that the web server has sufficient permissions to read the plugin egg.

Note that the Python version that the egg is built with must match the Python version with which Trac is run. If for instance you are running Trac under Python 2.3, but have upgraded your standalone Python to 2.4, the eggs won't be recognized.

For All Projects

With an .egg file

Some plugins (such as WebAdmin) are downloadable as a `.egg` file which can be installed with the `easy_install` program:

```
easy_install TracWebAdmin-0.1.1dev_r2765-py2.3.egg
```

If `easy_install` is not on your system see the Requirements section above to install it. Windows users will need to add the `Scripts` directory of their Python installation (for example, `C:\Python23\Scripts`) to their `PATH` environment variable (see easy_install Windows notes for more information).

If Trac reports permission errors after installing a zipped egg and you would rather not bother providing a egg cache directory writable by the web server, you can get around it by simply unzipping the egg. Just pass `–always-unzip` to `easy_install`:

```
easy_install --always-unzip TracWebAdmin-0.1.1dev_r2765-py2.3.egg
```

You should end up with a directory having the same name as the zipped egg (complete with `.egg` extension) and containing its uncompressed contents.

Trac also searches for globally installed plugins under `$prefix/share/trac/plugins` *(since 0.10)*.

From source

If you downloaded the plugin's source from Subversion, or a source zip file you can install it using the included `setup.py`:

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

Enabling the plugin

Unlike plugins installed per-environment, you'll have to explicitly enable globally installed plugins via trac.ini. This is done in the `components` section of the configuration file, for example:

```
[[:components]]
webadmin.* = enabled
```

The name of the option is the Python package of the plugin. This should be specified in the documentation of the Plugin, but can also be easily find out by looking at the source (look for a top-level directory that contains a file named `__init__.py`.)

Note: After installing the plugin, you may need to restart Apache.

Setting up the Plugin Cache

Some plugins will need to be extracted by the Python eggs runtime (`pkg_resources`), so that their contents are actual files on the file system. The directory in which they are extracted defaults to the home directory of the current user, which may or may not be a problem. You can however override the default location using the `PYTHON_EGG_CACHE` environment variable.

To do this from the Apache configuration, use the `SetEnv` directive as follows:

```
SetEnv PYTHON_EGG_CACHE /path/to/dir
```

This works whether you are using the CGI or the mod_python front-end. Put this directive next to where you set the path to the Trac environment, i.e. in the same `<Location>` block.

For example (for CGI):

```
<Location /trac>
  SetEnv TRAC_ENV /path/to/projenv
  SetEnv PYTHON_EGG_CACHE /path/to/dir
</Location>
```

or (for mod_python):

```
<Location /trac>
  SetHandler mod_python
  ...
  SetEnv PYTHON_EGG_CACHE /path/to/dir
</Location>
```

Note: this requires the `mod_env` module

For FastCGI, you'll need to `-initial-env` option, or whatever is provided by your web server for setting environment variables.

About hook scripts

If you have set up some subversion hook scripts that call the Trac engine - such as the post-commit hook script provided in the `/contrib` directory - make sure you define the `PYTHON_EGG_CACHE` environment variable within these scripts as well.

Troubleshooting

Is setuptools properly installed?

Try this from the command line:

```
$ python -c "import pkg_resources"
```

If you get **no output**, setuptools **is** installed. Otherwise, you'll need to install it before plugins will work in Trac.

Did you get the correct version of the Python egg?

Python eggs have the Python version encoded in their filename. For example, `MyPlugin-1.0-py2.4.egg` is an egg for Python 2.4, and will **not** be loaded if you're running a different Python version (such as 2.3 or 2.5).

Also, verify that the egg file you downloaded is indeed a ZIP archive. If you downloaded it from a Trac site, chances are you downloaded the HTML preview page instead.

Is the plugin enabled?

If you install a plugin globally (i.e. *not* inside the `plugins` directory of the Trac project environment) you will have to explicitly enable it in trac.ini. Make sure that:

- you actually added the necessary line(s) to the `components` section
- the package/module names are correct
- if you're reference a module (as opposed to a class), you've appended the necessary ".*"
- the value is "enabled", not e.g. "enable"

Check the permissions on the egg file

Trac must of course be able to read the file. Yeah, you knew that 😉

Check the log files

Enable logging in Trac, set the log level to `DEBUG` and then watch the log file for messages about loading plugins.

See also TracGuide, plugin list, component architecture

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

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

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