

# Trac Plugins

## Error: Macro TracGuideToc(None) failed

```
'NoneType' object has no attribute 'find'
```

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 plugins in Trac, you need to have [?setuptools](#) (version 0.6) installed.

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.

### For All Projects

Plugins that you want to use in all your projects (such as [?WebAdmin](#)) can be installed globally by running:

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

Alternatively, you can just drop the `.egg` file in the Python `site-packages` directory.

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`.)

## 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're 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>
```

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

---

See also [TracGuide](#), [?plugin list](#), [?component architecture](#)