
tracerlib Documentation

Release 0.1

Calvin Spealman

June 06, 2012

CONTENTS

Tracelib is a set of utilities to make tracing Python code easier.

It provides `TracerManager`, which can allow multiple trace functions to coexist. It can easily be enabled and disabled, either manually or as a context manager in a `with` statement.

Tracer classes make handling the different trace events much easier.

```
class TraceExceptions(Tracer):
    def trace_exception(self, func_name, exctype, value, tb):
        print "Saw an exception: %r" % (value,)
```

Tracer is also easily capable of filtering which events it listens to. It accepts both an `events` parameter, a list of trace events it will respond to, and a `watch` parameter, a list of paths it will respond to in the form of `package.module.class.function`.

This can easily wrap a trace function, or you can subclass `Tracer` and implement one of its helpful `trace_*()` methods.

And, a helper class `FrameInspector` which wraps a frame and makes it trivial to inspect the function name and arguments the function had been called with.

```
inspector = FrameInspector(sys._getframe())

print "Called", inspector.func_name
print "args:", inspector.args
print "kwargs:", inspector.kwargs
```


API

1.1 Managing multiple tracers

1.2 Tracer

1.3 StackTracer

1.4 FrameInspector

1.5 TracerManager

INDICES AND TABLES

- *genindex*
- *modindex*
- *search*