

## Trac Logging

Trac supports logging of system messages using Python's [logging module](#).

Logging is configured in the [\[logging\]](#) section of [TracIni](#).

### Supported Logging Methods

The log method is set using the [\[logging\] log\\_type](#) option, which accepts the following values:

#### none

Suppress all log messages.

#### file

Log messages to a file, specified with the [\[logging\] log\\_file](#) option. Relative paths for `log_file` are resolved relative to the environment `log` directory.

#### stderr

Output log entries to console ([tracd](#) only).

#### syslog

(UNIX) Send log messages to the local syslogd via named pipe `/dev/log`. By default, syslog will write them to the `/var/log/messages` file.

#### eventlog

(Windows) Use the system's NT Event Log.

### Log Levels

The verbosity level of logged messages can be set using the [\[logging\] log\\_level](#) option. The log level defines the minimum level of urgency required for a message to be logged, and those levels are:

#### CRITICAL

Log only the most critical (typically fatal) errors.

#### ERROR

Log failures, bugs and errors.

#### WARN

Log warnings, non-interrupting events.

#### INFO

Diagnostic information, log information about all processing.

#### DEBUG

Trace messages, profiling, etc.

Additionally, you can enable logging of SQL statements at debug level. This is turned off by default, as it's very verbose. Set [\[trac\] debug\\_sql = yes](#) to activate.

### Log Format

The output format for log entries can be specified in the [\[logging\] log\\_format](#) option. The format is a string which can contain any of the [Python logging Formatter variables](#). Additionally, the following Trac-specific variables can be used:

#### \$(basename)s

The environment directory name.

#### \$(path)s

The absolute path of the environment directory.

#### \$(project)s

The originating project's name.

Note that variables are identified using a dollar sign (`$(...)`s) instead of percent sign (`%(...)`s).

The default format is:

```
log_format = Trac[$(module)s] $(levelname)s: $(message)s
```

In a multi-project environment where all logs are sent to the same place (e.g. `syslog`), it makes sense to add the project name. In this example we use `basename` since that can generally be used to identify a project:

```
log_format = Trac[${basename}s:${module}s] ${levelname}s: ${message}s
```

---

See also: [TracIni](#), [TracEnvironment](#)