

# Trac Logging

## Table of Contents

- [Index](#)
- [Installation](#)
- [Upgrading](#)
- [Configuration](#)
- [Administration](#)
- [Backup](#)
- [Logging](#)
- [Permissions](#)
- [The Wiki](#)
- [Wiki Formatting](#)
- [Timeline](#)
- [Repository Browser](#)
- [Changesets](#)
- [Roadmap](#)
- [Tickets](#)
- [Ticket Queries](#)
- [Reports](#)
- [RSS Support](#)
- [Notification](#)
- [Customization](#)
- [Plugins](#)

Trac supports logging of system messages using the standard [logging module](#) that comes with Python.

Logging is configured in the `[logging]` section in [trac.ini](#).

## Supported Logging Methods

The log method is set using the `log_type` configuration option, which takes any of the following values:

### **none**

Suppress all log messages.

### **file**

Log messages to a file, specified with the `log_file` option in [trac.ini](#).

### **stderr**

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

### **syslog**

(UNIX) Send messages to local syslogd via named pipe `/dev/log`.

### **eventlog**

(Windows) Use the system's NT eventlog for Trac logging.

# Log Levels

The verbosity level of logged messages can be set using the *log\_level* directive in [trac.ini](#). The log level defines the minimum level of urgency required for a message to be logged.

The 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.

---

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