Trac Logging

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` option in trac.ini, 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 all log messages to the local syslogd via named pipe `/dev/log`. By default, syslog will write them to the file /var/log/messages.

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

Log Levels

The verbosity level of logged messages can be set using the `log_level` option in trac.ini. 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.

Note that starting with Trac 0.11.5 you can in addition enable logging of SQL statements, at debug level. This is turned off by default, as it's very verbose (set `[trac] debug_sql = yes` in TracIni to activate).

Log Format

Starting with Trac 0.10.4 (see ?#2844), it is possible to set the output format for log entries. This can be done through the `log_format` option in trac.ini. The format is a string which can contain any of the ?Python logging Formatter variables. Additonally, the following Trac-specific variables can be used:

$(basename)s
: The last path component of the current environment.

$(path)s
: The absolute path for the current environment.

$(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, TracGuide, TracEnvironment