

ログ

Trac は Python の標準 [ロギングモジュール](#) (訳注: [日本語ドキュメント](#)) を使用したシステムメッセージのログ出力に対応しています。

ログは [trac.ini](#) の [logging] セクションで設定することができます。

対応しているログの種類

ログの出力方法は [trac.ini](#) の log_type オプションで設定されます。以下の値が使用できます:

none

すべてのログメッセージを抑制する

file

ログをファイルに出力する。 [trac.ini](#) の log_file ディレクティブで指定する。 log_file の中の相対パスは [TracEnvironment](#) の log ディレクトリへの相対パスとして解決されます

stderr

コンソールにすべてのログを出力する ([tracd](#) のみ)

syslog

(UNIX) 名前付きパイプ /dev/log を通してすべてのログメッセージをローカルの syslog に送信する。 syslog はデフォルトでファイル /var/log/messages に出力される

eventlog

(Windows) イベントログに Trac のログを出力する

ログレベル

出力するログの冗長レベルは [trac.ini](#) の log_level

オプションで指定します。ログレベルは出力するログメッセージの最低限のレベルを定義します。レベルには下記の種類があります:

CRITICAL

最も重要なエラーのみ。たいていは致命的なメッセージです

ERROR

処理失敗、バグ、エラー

WARN

警告、処理を中断するほどではないイベント

INFO

診断メッセージ。すべてのプロセスについてのログ情報

DEBUG

トレースメッセージ、プロファイリングなど

Note: Trac 0.11.5 以降で SQL 文をログに出力できるようになりました。非常に冗長なログになるので、デフォルトでは OFF に成っています ([trac] debug_sql =yes を [TracIni](#) に設定することで有効化できます)。

ログの出力フォーマット

Trac 0.10.4 以降 ([#2844](#) を参照) では、 [trac.ini](#) の log_format

オプションを使用することで、ログエントリの出力フォーマットを設定することが可能です。フォーマットは [Python ロギングフォーマット変数](#) を含むことができる文字列です。そのうえ、以下の Trac 特有の変数を使用することができます:

\$(basename)s

Environment のベースネーム

\$(path)s

Environment の絶対パス

\$(project)s

プロジェクト名

Note: 変数には、パーセント記号 (%(...))s) ではなく、ドル記号 (\$(...))s) を使用します。

デフォルトのフォーマットは以下の通りです:

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

以下は、ログにプロジェクト名を出力する例です (全てのログが同じ場所 (例えば `syslog`) に出力される複数プロジェクト環境で役に立ちます)。この例では、プロジェクトを特定するのに `basename` を使用しています:

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

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