

Wikiprint Book

Title: Trac のインターフェースをカスタマイズする

Subject: SilverFrost - TracInterfaceCustomization

Version: 3

Date: 12/20/25 09:14:07

## SilverFrost 目次

Trac のインターフェースをカスタマイズする	3
イントロダクション	3
プロジェクトのロゴとアイコン	3
ロゴ	3
アイコン	3
ナビゲーション項目のカスタマイズ	3
サイトの外観	4
プロジェクトリスト	5
プロジェクトテンプレート	6

## Trac のインターフェースをカスタマイズする

### イントロダクション

このページは Trac の外観をカスタマイズする方法をユーザに提案するために書きました。主要な話題は HTML テンプレートと CSS ファイルであり、プログラムコードではありません。ユーザ自身の特定のニーズを満たすために Trac の外観を変更する方法を、ユーザに示すことを意図しています。 Trac の全てのユーザにとって有益な、インターフェース変更の提案は、このページに書くのではなくチケットを使用してください。  
(訳注: 本家サイトのチケットの話です)

### プロジェクトのロゴとアイコン

Trac のインターフェースをカスタマイズする中で、最も簡単なのは、ロゴとサイトアイコンです。両方とも [trac.ini](#) に設定するだけで構成できます。

ロゴやアイコンのイメージは、 Trac Environment フォルダの中の "htdocs" というフォルダに置かなければいけません。 (Note: バージョン 0.9 以前の Trac で作成したプロジェクトでは、このフォルダ自分で作成する必要があります)

Note: 実際は、ロゴとアイコンはサーバのどこかのパスにおいてもかまいません (Web サーバがアクセスできるところならですが)。設定の中でそれらの絶対またはサーバの相対 URL を使用します。

[trac.ini](#) の適切なセクションの構成は以下の通りです:

#### ロゴ

src の設定を site/ に続く画像ファイルの名前に変更してください。 width と height は画像ファイルにあわせて設定を変更してください。(Trac の chrome ハンドラはプロジェクトのディレクトリ htdocs の中のファイル用に "site/" を使用し、Trac インストール時に作成された共通のディレクトリ htdocs 用に "common/" を使用します。) Note: 'site/' はプロジェクト名の代わりに使っているのではなく、文字通り使用されるべき実際のプレフィックスです。例えば、プロジェクトに 'sandbox' という名前を付け、イメージファイルが 'red\_logo.gif' である場合、 'src' には 'sandbox/red\_logo.gif' ではなく、 'site/red\_logo.gif' を設定します。

```
[header_logo]
src = site/my_logo.gif
alt = My Project
width = 300
height = 100
```

#### アイコン

アイコンは .gif か .ico 形式の 32x32 の画像である必要があります。 icon の設定を site/ に続くアイコンファイルの名前に変更してください。アイコンは通常、サイトの URL の横や、 ■■■■■ メニューに表示されます。

```
[project]
icon = site/my_icon.ico
```

Note: Internet Explorer では、ホストのルートディレクトリにある favicon.ico という名前のファイルしか許容しないため、このアイコンは無視されます。プロジェクトアイコンを IE と他のブラウザで共用したい場合、アイコンをホストのドキュメントルートに配置し、 trac.ini から以下のように参照します:

```
[project]
icon = /favicon.ico
```

ブラウザのアドレスバーでのアイコン表示に問題がある場合、アイコンのファイル拡張子の後に "?" (クエスチョンマーク) を置くと回避することができます。

```
[project]
icon = /favicon.ico?
```

### ナビゲーション項目のカスタマイズ

[mainnav] と [metanav] を使用すると、ナビゲーション項目に使用されるテキストとリンクをカスタマイズしたり、無効化することができます（新規項目を追加することはできません）。

以下の例では、Wiki のスタートページへのリンク名を "Home" に変更して、"Help/Guide" を非表示にします。さらに、"View Tickets" エントリを特定のレポートにリンクさせます。

```
[mainnav]
wiki.label = Home
tickets.href = /report/24

[metanav]
help = disabled
```

mainnav と metanav についての、より詳細な記述は [TracNavigation](#) を参照してください。

## サイトの外観

Trac はテンプレートエンジンに [Genshi](#) を使用しています。ドキュメントはまだ書かれていませんが、次の tip は動くはずです。

カスタムスタイルシートへのリンクや、独自のヘッダやフッタを追加したい場合、以下のような内容ファイルを、プロジェクトの templates ディレクトリに site.html という名前で作成してください（各 Trac プロジェクトは独自の内容の site.html を持つことができます）。/path/to/env/templates/site.html の例：

```
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:py="http://genshi.edgewall.org/"
      py:strip="">

    <!-- Add site-specific style sheet -->
    <head py:match="head" py:attrs="select('@*')">
        ${select('*|comment()|text())}
        <link rel="stylesheet" type="text/css"
              href="${href.chrome('site/style.css')}" />
    </head>

    <body py:match="body" py:attrs="select('@*')">
        <!-- Add site-specific header -->
        <div id="siteheader">
            <!-- Place your header content here... -->
        </div>

        ${select('*|text())}

        <!-- Add site-specific footer -->
        <div id="sitefooter">
            <!-- Place your footer content here... -->
        </div>
    </body>
</html>
```

XSLT に慣れ親しんだ人であれば、Genshi テンプレートには類似点があるのに気付くかもしれません。しかしながら Trac 固有の機能もあります。例えば \${href.chrome('site/style.css')} は Environment に含まれる htdocs/ にある CSS ファイルへの参照の属性に置き換えられます。同様に、\${chrome.htdocs\_location} は Trac インストール時に作成された共通の htdocs ディレクトリを指定するために使用します。しかし [\[trac\] htdocs\\_location](#) のコンフィグを設定することで、ディレクトリ位置を指定しなおすことができます。

site.html はサイト固有のすべての変更を含んでいる一つのファイルです。通常は、特定のセクション上で py:match を要素 (element) または属性 (attribute) として使用することで、カスタマイズしようとしているページを思い通りにレンダリングすることができます。[メーリングリスト](#) には上記の例の site.html について解説されていますので、参照してください。site.html には変更を行うための py:match セクションをいくつでも記載することができます。これらはすべて Genshi の文法に沿って行います。

<http://genshi.edgewall.org/wiki/Documentation/xml-templates.html> ドキュメントや詳細なシンタックス を参考にしてください。

チケット登録のフォームに導入テキストを表示する（プレビューが非表示のとき）場合は、次の例を追加してください：

```
<form py:match="div[@id='content' and @class='ticket']/form" py:attrs="select('*')">
<py:if test="req.environ['PATH_INFO'] == '/newticket' and (not 'preview' in req.args)">
    <p>Please make sure to search for existing tickets before reporting a new one!</p>
</py:if>
${select('*')}
</form>
```

この例では `req.environ['PATH_INFO']` を使用して、特定のビューだけで変更が行われるようにスコープを限定しています。例えば `site.html` でタイムラインだけで変更を行い、他のセクションには影響を及ぼしたくない場合は、`req.environ['PATH_INFO'] == '/timelime'` を `<py:if>` の `test` 属性に記載します。

より多くの site.html の例が [CookBook/SiteHtml](#) で見ることができます。

style.css の例は [CookBook/SiteStyleCss](#) で見ることができます。

0.10 からアップグレードされた Environment で、かつ `site_newticket.cs` ファイルが既に存在している場合は、ワークアラウンドすることによってテンプレートをロードすることができます - ClearSilver の処理が含まれていない場合に限ります（訳注: `<?cs?>` が含まれていない場合）。また、この場合はただ一つの要素 (`element`) だけがインポートされるので、コンテンツはある種のラッパー (`<div>` ブロックやそれに似た親コンテナ) を必要とします。インクルードするためには `XInclude` の名前空間を指定しなければなりませんが、ドキュメントルート以外にも置くことができます:

```
<form py:match="div[@id='content' and @class='ticket']/form" py:attrs="select('*')"  
      xmlns:xi="http://www.w3.org/2001/XInclude">  
<py:if test="req.environ['PATH_INFO'] == '/newticket' and (not 'preview' in req.args)">  
  <xi:include href="site_newticket.cs"><xi:fallback /></xi:include>  
</py:if>  
${select('*')}  
</form>
```

また、共通のテンプレートディレクトリに、`site.html`（その名前にも関わらず）を置くことができます - [\[inherit\] templates\\_dir](#) オプションを参照してください。新しく、一個のグローバルな `site.html` ファイルに、ヘッダ、フッタ、チケット作成時の tips を組み込むことで、簡単なメンテナンス（および、大きなインストールを行った 0.10 からのバージョンアップのための移行パス）を提供しています。

プロジェクトリスト

複数の Trac プロジェクトを動かしているときに、カスタマイズした Genshi テンプレートを使用して、プロジェクトの一覧を表示することができます。

以下に示すのは Trac が使用している、ホストするプロジェクトへのリンクのリストを表示するための基本のテンプレートです。ロードできないプロジェクトについては、エラーメッセージ

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
      xmlns:py="http://genshi.edgewall.org/"
      xmlns:xi="http://www.w3.org/2001/XInclude">
<head>
    <title>██████████</title>
</head>
<body>
    <h1>██████████</h1>
    <ul>
        <li py:for="project in projects" py:choose="">
            <a py:when="project.href" href="$project.href"
               title="$project.description">$project.name</a>
        <py:otherwise>
            <small>$project.name: <em>████</em> <br /> ($project.description)</small>
        </py:otherwise>
    
```

```

</li>
</ul>
</body>
</html>

```

カスタムテンプレートを使用する場合、Web サーバにテンプレートのロケーションの設定を読み込ませる必要があります（確かめてみてください … まだ 0.11 向けに変更していません）：

mod\_wsgi 用：

```
os.environ['TRAC_ENV_INDEX_TEMPLATE'] = '/path/to/template'
```

FastCGI 用：

```
FastCgiConfig -initial-env TRAC_ENV_PARENT_DIR=/parent/dir/of/projects \
               -initial-env TRAC_ENV_INDEX_TEMPLATE=/path/to/template
```

mod\_python 用：

```
PythonOption TracEnvParentDir /parent/dir/of/projects
PythonOption TracEnvIndexTemplate /path/to/template
```

CGI 用：

```
SetEnv TRAC_ENV_INDEX_TEMPLATE /path/to/template
```

TracStandalone の tracd を動かすのに使用するシェルの中で TRAC\_ENV\_INDEX\_TEMPLATE の環境変数を設定する必要があるでしょう：

- Unix

```
$ export TRAC_ENV_INDEX_TEMPLATE=/path/to/template
```

- Windows

```
$ set TRAC_ENV_INDEX_TEMPLATE=/path/to/template
```

## プロジェクトテンプレート

個々の Trac Environment (プロジェクトのインスタンス)

の外観は、同じサーバにホストされている他のプロジェクトとは独立してカスタマイズできます。推奨するのは site.html テンプレート ([サイトの外観 参照](#)) を使う方法です。どのような場合でも可能な限り、この方法にしてください。 site.html を使う場合、変更はオリジナルのテンプレートがレンダリングした結果に対して適用されるので、Trac を今後アップグレードした後も、通常はカスタマイズをそのまま使い続けることができます。 theme.html や他の Trac のテンプレートのコピーを作成する方法の場合、新しい Trac の機能追加や不具合修正の結果、動かなくなってしまったカスタマイズを新しいバージョンに移行する必要があるかもしれません。

リスクを許容して扱う必要がありますが、Trac テンプレートはコピーしてカスタマイズすることもできます。デフォルトの Trac テンプレートはインストールされた Trac egg (/usr/lib/pythonVERSION/site-packages/Trac-VERSION.egg/trac/templates, .../trac/ticket/templates, .../trac/wiki/templates, ...) 内に配置されています。[プロジェクトリスト](#) のテンプレートファイルは index.html が使用されており、各ページに共通する主要なレイアウトを提供するテンプレートは theme.html が使用されます。画像や CSS スタイルシートなどのページの部品は、egg の trac/htdocs ディレクトリに配置されています。

しかし、Trac egg 内部のテンプレートやサイトのリソースは編集しないでください。Trac を再インストールしたときに、カスタマイズの内容が完全に失われてしまいます。代わりに、以下に挙げる方法のいずれかを使ってください：

- カスタマイズが単独のプロジェクトに閉じているのであれば、テンプレートをプロジェクトの templates ディレクトリにコピーしてください
- カスタマイズが複数のプロジェクトに渡るものであるなら、テンプレートを共有のロケーションにコピーし、各プロジェクトからは trac.ini の [inherit] templates\_dir = オプションで、その位置を指定してください

Trac は以下の順序で、テンプレートファイルを探します。まず、プロジェクトの内部を探し、存在しなければ `inherit` で指定された場所、最後に Trac egg の内部を探します。

Trac は通常、パフォーマンスを向上させるために、テンプレートをメモリ上にキャッシュします。変更したテンプレートを適用するためには、サーバプロセスの再起動が必要です。

---

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