

Wikiprint Book

Title: チケットクエリ

Subject: SilverFrost - TracQuery

Version: 3

Date: 12/20/25 12:08:38

SilverFrost 目次

チケットクエリ	3
フィルタ	3
チケットナビゲーション	3
クエリを保存する	3
TracLinks を使用する	3
[[TicketQuery]] マクロを使用する	4
table 形式でカスタマイズする	4
全行表示	4
クエリ言語	5

チケットクエリ

[レポート](#)に加えて、Trac は カスタムチケットクエリ があります。 カスタムチケットクエリは、ある一定の基準を満たすチケットのリストを表示するのに使用されます。

カスタムクエリを設定・実行するために、ナビゲーションバーの チケットを見る (英語版では *View tickets*) モジュールに切り替え、 カスタムクエリ (英語版では *Custom Query*) のリンクを選択します。

フィルタ

最初にクエリのページを開くと、デフォルトのフィルタが適用され、自分に関連するチケットの一覧が表示されます。

- ログインしている場合、自分が担当者 ("assigned to") になっている、オープンチケットが表示されます。
 - ログインしていない場合でも、個人設定 ("Preferences") で名前やメールアドレスを設定している場合は、自分が設定した名前やメールアドレスが、関係者 ("CC") に含まれているオープンチケットが表示されます。
 - ログインしておらず、個人設定も保存していない場合、すべてのオープンチケットが表示されます。

現在のフィルタは検索条件の左側にある " - " (マイナス)

のラベルのボタンをクリックすることにより削除することができます。新しいフィルタは、フィルタボックスの下にあるプルダウンリストから追加することができます ('And' 条件は左側に、'Or' 条件は右側に配置されています)。テキストボックスやオプションのプルダウンメニューのフィルタを or 条件として、いくつでも追加することができます。

フィルタボックスの下のフィールドを使用することによって、結果をフィールドごとにグループ化したり、各チケットの完全な説明を表示することができます。

一度フィルタを編集したら、**更新**（英語版では *Update*）ボタンで結果を更新することができます。

チケットナビゲーション

クエリの結果の一つをクリックすると、そのチケットが表示されます。メインメニューの次のチケット (英語版では Next Ticket) または前のチケット (英語版では Previous Ticket) というリンクをクリックすることによって、クエリの結果を行き来することができます。クエリに戻る (英語版では Back to Query) というリンクをクリックすると、クエリのページに戻ります。

あなたは安全にチケットの編集をすることができ、編集を保存した後に 次のチケット/前のチケット/クエリに戻る
の各リンクを使用して、結果を行き来することができます。クエリに戻ったとき、編集済みのチケット
はイタリックのテキストで表示されます。チケットの 1 つが編集され、クエリの基準を満たさなくなった場合、テキストはグレーで表示されます。
クエリの基準を満たす新しいチケットが登録されるると、テキストが太字で表示されます。

クエリの結果は、更新ボタンを押すごとにステータスがクリアされリフレッシュされます。

クリップを保存する

Trac ではクエリをレポートモジュールからアクセスできる名前付きクエリとして保存することができます。クエリを保存するには **更新** (英語版では **Update**) ボタンで表示を更新してから、画面の下の方にある **クエリを保存する** (英語版では **Save query**) ボタンを押下します。また、下記の方法で **Wiki** のコンテンツとしてクエリへの参照を保存することもできます。

Note: 下記の方法で使用するクエリを簡単に組み立てるには、カスタムクエリモジュールでクエリを組み立ててテストしてから、クエリを保存する
を押します。すると、すでに組み立てられたクエリ文字列が得られるので、余計な改行文字を削除するとそのまま使うことができます。

TracLinks を使用する

クエリを保存しておくと、後から保存したクエリを再度実行できます。保存は、Wiki ページにクエリへのリンクを作成することで行います。

表示内容·

1.0 での未解決のチケット

ここでは、検索内容を特定する上で単純なクエリ言語が使用されています。（[クエリ言語](#) 参照）

あるいは、タブリト実行した画面の URL をコピーし、以降の文字列（註記：URL のパラメータ部）を Wiki リンクに貼り付けることができます。

```
[query: ?status=new&status=assigned&status=reopened&group=owner ██████████████████]
```

表示内容:

担当者別着手済みチケット

[[TicketQuery]] マクロを使用する

[TicketQuery](#) マクロは条件にマッチするチケットの一覧を表示します。 [WikiFormatting](#) が使用できるいずれの場所でも使用できます。

例:

```
[[TicketQuery(version=0.6|0.7&resolution=duplicate)]]
```

表示内容:

一致するもののがありません

[query: Wiki リンク](#) 同様、このマクロのパラメータは [クエリ言語](#) に規定されるルールに基づいたクエリ文字列で記述しなければなりません。

チケットのサマリを表示しない、よりコンパクトな表示も利用可能です:

```
[[TicketQuery(version=0.6|0.7&resolution=duplicate, compact)]]
```

表示内容:

一致するもののがありません

クエリにマッチした件数だけを取得したいのならば、 `count` パラメータを使用してください。

```
[[TicketQuery(version=0.6|0.7&resolution=duplicate, count)]]
```

表示内容:

0

table 形式でカスタマイズする

table 形式 (`format=table`) では、 `col=<field>` を使用することで列の表示をカスタマイズすることができます。複数のフィールドを指定したい場合、表示する順に列をパイプ (|) で区切りで指定します:

```
[[TicketQuery(max=3, status=closed, order=id, desc=1, format=table, col=resolution|summary|owner|reporter)]]
```

以下のように表示されます:

チケット	解決方法	概要	担当者	報告者
----------------------	----------------------	--------------------	---------------------	---------------------

一致するものが見つかりませんでした。

全行表示

table 形式では、以下の例のように `rows=<field>` を使用することで、全行表示を行うことができます:

```
[[TicketQuery(max=3, status=closed, order=id, desc=1, format=table, col=resolution|summary|owner|reporter, rows=description)]]
```

以下のように表示されます:

チケット	解決方法	概要	担当者	報告者
----------------------	----------------------	--------------------	---------------------	---------------------

一致するものが見つかりませんでした。

クエリ言語

query: [TracLinks](#) と [[TicketQuery]] マクロは、いずれもクエリのフィルタを特定するために小さな "クエリ言語" を使用します。基本的に、フィルタはアンパサンド (&) で区切られます。それぞれのフィルタはチケットの各フィールドの名前、オペレータ、1つ以上の値で構成されています。値が複数ある場合はパイプ (|) で区切り、フィルタがいずれかの値にマッチすればいいことを意味しています。 & や | の文字そのものを値に含めたい場合、バックスラッシュ (\, 環境によっては円マーク) でエスケープしてください。

オペレーター一覧:

=	フィールドの内容はいずれかの値と完全に一致する
~=	フィールドの内容は1つ以上の値を含む
^=	フィールドの内容が値の一つで始まる
\$=	フィールドの内容が値の一つで終了する

これらのオペレータは否定の意味があります:

!=	フィールドの内容がいずれの値にも一致しない
!~=	フィールドの内容がいずれの値も含んでいない
!^=	フィールドの内容がいずれの値でも始まらない
!\$=	フィールドの内容がいずれの値でも終わらない

日付フィールドである **■■■** (英語版では `created`) と **■■■** (英語版では `modified`) は = オペレータに二つの日付を、二つのドット (...) で区切って指定することができます。日付の範囲の片方が空の場合、空の方向には条件を指定しないことを意味します。また、日付パーサは自然言語に似た構文を解釈します。 "3 weeks ago", "last month", "now" などです。 Bugzilla スタイルの "1d", "2w", "3m", "4y" なども、それぞれ1日、2週間、3ヶ月、4年と解釈できます。クエリ文字列に指定する場合、これらの構文からは空白文字を除外して指定してください。

<code>created=2007-01-01..2008-01-01</code>	2007年に登録されたチケットへのクエリ
<code>created=lastmonth..thismonth</code>	前月に登録されたチケットへのクエリ
<code>modified=1weekago..</code>	1週間以内に更新されたチケットへのクエリ
<code>modified=..30daysago</code>	30日間更新されていないチケットへのクエリ

See also: [TracTickets](#), [TracReports](#), [TracGuide](#)