

Trac Ticket Queries

Error: Macro TracGuideToc(None) failed

```
'NoneType' object has no attribute 'find'
```

In addition to [reports](#), Trac provides *custom ticket queries*, which display tickets meeting specified criteria.

To configure and execute a custom query, navigate to the *View Tickets* module from the navigation bar, and select the *New Custom Query* link.

Filters

When you first go to the query page, the default filter will display tickets relevant to you:

- If logged in, all open tickets assigned to you.
- If not logged in, but you have specified a name or email address in the preferences, all open tickets where your name or email is in the CC list.
- If not logged in and no name/email is defined in the preferences, then all open issues.

New filters are added using the dropdown lists at the bottom corners of the filters box; 'And' conditions on the left, 'Or' conditions on the right. Filters are removed by clicking the button to the left with the minus sign on the label.

After you have edited your filters, click the *Update* button to refresh your results.

Filters with either a text box or a dropdown menu of options can be added multiple times to perform an *Or* on the criteria. Add additional 'Or's by Clicking the 'And' Dropdown and selecting an item that you have already selected. For example, to select "Milestone is X *Or* Milestone is Y", Select 'Milestone', Select 'X', then click the 'And' Dropdown, select 'Milestone' a second time and select 'Y'.

For text fields such as Keywords and CC the `-` operator can be used to negate a match and double quotes (*since 1.2.1*) can be used to match a phrase. For example, a *contains* match for `word1 word2 -word3 "word4 word5"` matches tickets containing `word1` and `word2`, not `word3` and `word4 word5`.

You can use the controls just below the filters box to group the results based on a field, or display the full description for each ticket.

Keyboard shortcuts are available for manipulating the *checkbox* filters:

- Clicking on a filter row label toggles all checkboxes.
- Pressing the modifier key while clicking on a filter row label inverts the state of all checkboxes.
- Pressing the modifier key while clicking on a checkbox selects the checkbox and deselects all other checkboxes in the filter. Since 1.2.1 this also works for the *Columns* checkboxes.

The modifier key is platform and browser dependent. On Mac the modified key is Option/Alt or Command. On Linux the modifier key is Ctrl + Alt. Opera on Windows seems to use Ctrl + Alt, while Alt is effective for other Windows browsers.

Navigating Tickets

Clicking on one of the query results will take you to that ticket. You can navigate through the results by clicking the *Next Ticket* or *Previous Ticket* links just below the main menu bar, or click the *Back to Query* link to return to the

query page.

You can safely edit any of the tickets and continue to navigate through the results using the *Next/Previous/Back to Query* links after saving your results. When you return to the query *any tickets which were edited* will be displayed with italicized text. If one of the tickets was edited such that *it no longer matches the query criteria*, the text will also be greyed. If **a new ticket matching the query criteria has been created**, it will be shown in bold.

The query results can be refreshed and cleared of these status indicators by clicking the *Update* button again.

Saving Queries

Trac allows you to save the query as a named query accessible from the reports module. To save a query ensure that you have *Updated* the view and then click the *Save query* button displayed beneath the results. You can also save references to queries in Wiki content, as described below.

Note: one way to easily build queries like the ones below is to create and test the queries using Custom Query module. Clicking *Save query* will display the query string for you, all you need to do is remove the extra line breaks.

Note: you must have the **REPORT_CREATE** permission in order to save queries to the list of default reports. The *Save query* button will only appear if you are logged in as a user that has been granted this permission. If your account does not have permission to create reports, you can still use the methods below to save a query.

TracLinks to Queries

You can make a link to a query from any Wiki page using a simple query language to specify the criteria.

```
[query:status=new|assigned|reopened&version=1.0 Active tickets against 1.0]
```

Which is displayed as:

Active tickets against 1.0

Alternatively, you can copy the query string from the browser URL box and paste it into the Wiki link, including the leading ? character:

```
[query:?status=new&status=assigned&status=reopened&group=owner Assigned tickets by owner]
```

Which is displayed as:

Assigned tickets by owner

Query Language

The `query`: TracLinks and the `[[TicketQuery]]` macro both use a mini ?query language? for specifying query filters. Filters are separated by ampersands (&). Each filter consists of the ticket field name, an operator and one or more values. Multiple values are separated using a pipe (|), meaning the filter matches any of the values. To include a literal & or | in a value, escape the character with a backslash (\).

The available operators are:

= the field content exactly matches one of the values

`~=` the field content contains one or more of the values
`^=` the field content starts with one of the values
`$=` the field content ends with one of the values
 All of these operators can also be negated:

`!=` the field content matches none of the values
`!~=` the field content does not contain any of the values
`!^=` the field content does not start with any of the values
`!$=` the field content does not end with any of the values

Filters combining matches and negated matches can be constructed for text fields such as Keywords and CC using the *contains* (`~=`) operator. The `-` operator is used to negate a match and double quotes (*since 1.2.1*) are used for whitespace-separated words in a phrase. For example, `keywords~=word1 word2 -word3 "word4 word5"` matches tickets containing `word1` and `word2`, not `word3` and also `word4 word5`.

<code>status=closed,keywords~=firefox</code>	query closed tickets that contain keyword <code>firefox</code>
<code>status=closed,keywords~=opera</code>	query closed tickets that contain keyword <code>opera</code>
<code>status=closed,keywords~=firefox opera</code>	query closed tickets that contain keywords <code>firefox</code> and <code>opera</code>
<code>status=closed,keywords~=firefox opera</code>	query closed tickets that contain keywords <code>firefox</code> or <code>opera</code>
<code>status=closed,keywords~=firefox,or,keywords~=opera</code>	query closed tickets that contain keyword <code>firefox</code> , or (closed or unclosed) tickets that contain keyword <code>opera</code>
<code>status=closed,keywords~=firefox -opera</code>	query closed tickets that contain keyword <code>firefox</code> , but not <code>opera</code>
<code>status=closed,keywords~=opera -firefox</code>	query closed tickets that contain keyword <code>opera</code> , but no <code>firefox</code>

The date fields `created` and `modified` can be constrained by using the `=` operator and specifying a value containing two dates separated by two dots (`. .`). Either end of the date range can be left empty, meaning that the corresponding end of the range is open. The date parser understands a few natural date specifications like "3 weeks ago", "last month" and "now", as well as Bugzilla-style date specifications like "1d", "2w", "3m" or "4y" for 1 day, 2 weeks, 3 months and 4 years, respectively. Spaces in date specifications can be omitted to avoid having to quote the query string.

<code>created=2007-01-01..2008-01-01</code>	query tickets created in 2007
<code>created=lastmonth..thismonth</code>	query tickets created during the previous month
<code>modified=1weekago..</code>	query tickets that have been modified in the last week
<code>modified=..30daysago</code>	query tickets that have been inactive for the last 30 days

Note that `modified` is the *last modified time*, so `modified` with a date range shows ticket that were *last modified* in that date range. If a ticket was modified in the date range, but modified again after the end date, it will not be included in the results.

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