Trac Ticket Queries

In addition to reports, Trac provides support for *custom ticket queries*, which can be used to display tickets that meet specified criteria.

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

Filters

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

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

Current filters can be removed by clicking the button to the left with the minus sign on the label. New filters are added from the dropdown lists at the bottom corners of the filters box; 'And' conditions on the left, 'Or' conditions on the right. Filters with either a text box or a dropdown menu of options can be added multiple times to perform an *Or* on the criteria.

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

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

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. Lastly, 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, you can build and test the queries in the Custom report module and when ready - click *Save query*. This will build 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.

Using TracLinks

You may want to save some queries so that you can come back to them later. You can do this by making a link to the query from any Wiki page.

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

Which is displayed as:

> Active tickets against 1.0

This uses a very simple query language to specify the criteria, see Query Language.

Alternatively, you can copy the query string of a query and paste that 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

Customizing the *table* format

You can also customize the columns displayed in the table format (*format=table*) by using *col=<field>*. You can specify multiple fields and what order they are displayed in by placing pipes (`|`) between the columns:

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

This is displayed as:

Ticket	Resolution	Summary	Owner	Reporter
No tickets found				

Full rows

In *table* format you can also have full rows by using *rows=<field>*:

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

This is displayed as:

Ticket	Resolution	Summary	Owner	Reporter
No tickets found				

Query Language

`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. More than one value are separated by a pipe (`|`), meaning that 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 when 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`.

`status=closed,keywords~=firefox`	query closed tickets that contain keyword `firefox`
`status=closed,keywords~=opera`	query closed tickets that contain keyword `opera`
`status=closed,keywords~=firefox opera`	query closed tickets that contain keywords `firefox` and `opera`
`status=closed,keywords~=firefox|opera`	query closed tickets that contain keywords `firefox` or `opera`
`status=closed,keywords~=firefox,or,keywords~=opera`	query closed tickets that contain keyword `firefox`, or (closed or unclosed) tickets that contain keyword `opera`
`status=closed,keywords~=firefox -opera`	query closed tickets that contain keyword `firefox`, but not `opera`
`status=closed,keywords~=opera -firefox`	query closed tickets that contain keyword `opera`, but no `firefox`

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.

`created=2007-01-01..2008-01-01`	query tickets created in 2007
`created=lastmonth..thismonth`	query tickets created during the previous month
`modified=1weekago..`	query tickets that have been modified in the last week
`modified=..30daysago`	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, TracGuide, TicketQuery