

Trac Ticket Queries

In addition to [reports](#), Trac provides support for *custom ticket queries*, used to display lists of tickets meeting a specified set of 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 and no name/email defined in the preferences then all open issues are displayed.

Current filters can be removed by clicking the button to the right with the minus sign on the label. New filters are added from the pulldown 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 pulldown menu of options can be added multiple times to perform an *or* of the criteria.

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.

Once you've edited your filters click the *Update* button to refresh your results.

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.

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

Using the `[[TicketQuery]]` Macro

The `?TicketQuery` macro lets you display lists of tickets matching certain criteria anywhere you can use [WikiFormatting](#).

Example:

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

This is displayed as:

#5

inverse operation of "calc t make_time" to get time string in numeric format

Just like the [query: wiki links](#), the parameter of this macro expects a query string formatted according to the rules of the simple [ticket query language](#).

A more compact representation without the ticket summaries is also available:

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

This is displayed as:

#5

Finally if you wish to receive only the number of defects that match the query using the count parameter.

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

This is displayed as:

1

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 by placing pipes (|) between the columns like below:

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

This is displayed as:

Results (1 - 3 of 26)

1 2 3 4 5 6 7 8 9 ?

<u>Ticket</u>	<u>Resolution</u>	<u>Summary</u>	<u>Owner</u>	<u>Reporter</u>
#43	invalid	Deconvolution script for S to P Receiver Function (RF)	MarcusWalther	subhrajit.das10
#42	fixed	standard magnitudes	MarcusWalther	SWendt
#38	worksforme	locdiff - segmentation fault in version 5.5 2011	MarcusWalther	sdonner

1 2 3 4 5 6 7 8 9 ?

Full rows

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

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

This is displayed as:

Results (1 - 3 of 26)

1 2 3 4 5 6 7 8 9 ?

<u>Ticket</u>	<u>Resolution</u>	<u>Summary</u>	<u>Owner</u>	<u>Reporter</u>
#43	invalid	Deconvolution script for S to P Receiver Function (RF)	MarcusWalther	subhrajit.das10
Description		I have got a deconvolution script for p to s RF but i need to develop the opposite one (S to P). I am new to this and I dont get the commands being used. I am posting the P to S decon script here.....please suggest me how can I develop the my destined one...		
#42	fixed	standard magnitudes	MarcusWalther	SWendt

Description

Using the [[TicketQuery]] Macro

<u>Ticket</u>	<u>Resolution</u>	<u>Summary</u>	<u>Owner</u>	<u>Reporter</u>
---------------	-------------------	----------------	--------------	-----------------

Improve SH to use adopted poles and zeros for Wood-Anderson simulation filter and updated static magnification.

#38	worksforme	<u>locdiff - segmentation fault in version 5.5 2011</u>	MarcusWalther	sdonner
-----	------------	---	---------------	---------

Hi

I just installed the new version of SH (version 5.5 February-2011, Copyright (C) 2011) Using the utility "locdiff" results in "Segmentation fault".

I tried to recompile everything in the sh/util directory as root with the command make -f Makefile

Description

Result: make libutil.a buttfreq butrec fconcat fereg findevnt flf2flt evalflf catflf locadd locdiff find_next_loc restfreq restrec simfreq simrec q2gse hypo_stlist ev2telex ev2hypo ev2hyposat ev2hypocenter ev4hypocenter ev4hypou ev2view ev2bull autoloc2evt evt2reltimes evt2relonset evt2phasetimes relonset2slo onset_convert check_evt reformat_locsat_out archevent_shc timename evt_copy_and_split evt_time_sort evt_correct_aniso evt_extract_ms qed_ms_make_evt vrs_read_diff travelttime slo2distance statinf fit_travel fit_travel2 crusttrav fit_to_line getresid floatop weather2ascii beamloc stddev find_topo30_id stringop lynx_reformat auto_statlist richter_ml make[1]: Entering directory `/usr/local/sh/sh/util' ar r libutil.a eventdsc.o evtelex.o evhypo.o evhypout.o timelist.o evbullet.o; libutil.a ar: creating libutil.a /bin/sh: libutil.a: command not found make[1]: * [libutil.a] Error 127 make[1]: Leaving directory `/usr/local

But the libutil.a file is existent. I helped myself by copying the executable of an older version from a colleague ...

Cheers

1 2 3 4 5 6 7 8 9 ?

Query Language

query: [TracLinks](#) and the `[[TicketQuery]]` macro both use a mini ?query language? for specifying query filters. Basically, the filters are separated by ampersands (&). Each filter then 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 litteral & or | in a value, escape the character with a backslash (\).

The available operators are:

=	the field content exactly matches the 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
-------------	---

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 left out 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

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