Qt DOCUMENTATION

Search   Topics >

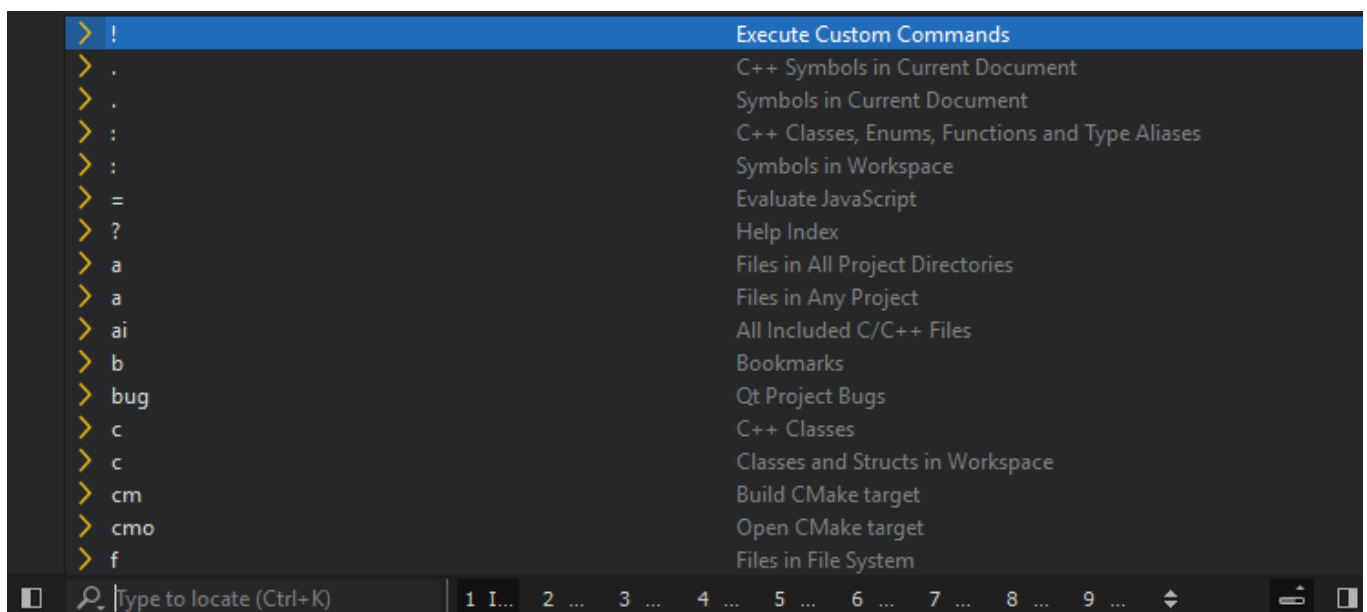Qt Design Studio Manual > Searching with the Locator

Searching with the Locator

You can find the locator in the bottom left of the Qt Design Studio window.
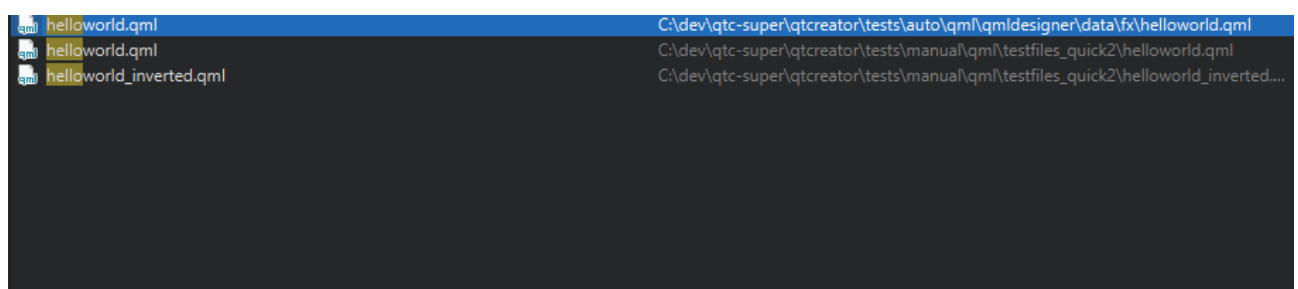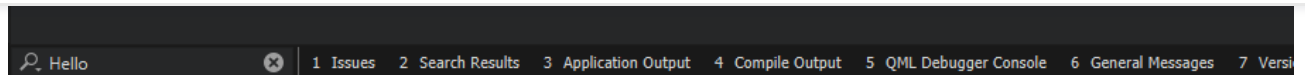

To activate the locator:

> Press **Ctrl+K** (**Cmd+K** on macOS).

> Select **Tools** > **Locate**.

> Select **Edit** > **Go to Line**.

> Click the line and column indicator on the editor toolbar.

To open a QML file called *HelloWorld.qml* in the currently open project using the locator:

1. Activate the locator by pressing **Ctrl+K**.

2. Start typing the filename.

Qt DOCUMENTATION

> Hello   ⊗   1 Issues   2 Search Results   3 Application Output   4 Compile Output   5 QML Debugger Console   6 General Messages   7 Versi

3. Move to the filename in the list and press **Enter**.

   The file opens in the editor.

4. To move to a line in the file, enter the line number in the locator.

To move directly to a particular line and column in the document when you open the document, append them to the file name in the locator, separated by plus signs (+) or colons (:). For example, to open *HelloWorld.qml* to line 41 and column 2, enter: `HelloWorld.qml:41:2`.

If the path to a file is very long, it might not fit into the locator window. To view the full path, press **Alt** when the filename is selected or use the handle next to the locator window to increase the window width.

It is also possible to enter only a part of a search string. As you type, the locator shows the occurrences of that string regardless of where in the name of a component it appears. Some locator filters, such as colon and `m`, support *fuzzy* matching, which means that you can enter the uppercase letters to locate a symbol when using camel case or the letters after the underscore when using snake case.

To narrow down the search results, you can use the following wildcard characters:

> To match any number of any or no characters, enter `*`.

> To match a single instance of any character, enter `?`.

Using Locator Filters

The locator enables you to browse not only files, but any items defined by **locator filters**. The filters that are available depend on the file type:

> Locating any open document (`o`)

> Locating files anywhere on your file system (`f`)

> Locating files belonging to your project (`p`), such as source, header, resource, and `.ui` files, or to any project (`a`)

> Locating QML methods (`m`)

> Locating symbols in the current document (`.`)

> Locating a specific line and column in the document displayed in your editor (`l <line_number>: <column_number>`)

> Opening help topics, including Qt documentation (`?`)

> Performing web searches (`r`)

> Executing JavaScript (`=`), especially useful for calculations. For more information, see Executing JavaScript.

> Executing shell commands (`!`)

> Executing version control system commands (`git`). For more information, see Using Git.

> Triggering menu items from the main menu (`t`)

> Searching for issues from the Qt Project Bug Tracker (`bug`).

> Searching for applications, documents, and other files by using platform-specific external tools or commands (`md`). The following tools are used by default, but you can configure the locator to use any other command:

  > On macOS: using Spotlight

Qt DOCUMENTATION

On Linux, using the locate command.

To use a specific locator filter, type the assigned prefix followed by **Space**. The prefix is usually a single character. Then type the search string (typically, a filename or class name) or the command to execute.

You can also double-click a locator filter in the filter list to use it. You can use the up and down arrow keys or the **Ctrl+P** and **Ctrl+N** keyboard shortcuts to move up and down the list, and then press **Enter** to use the selected filter.

For example, to create a new file and open it in the editor, type `f` followed by **Space**, followed by path and file name, and then press **Enter**.

You can use the filter that triggers menu commands to open sessions. Enter `t yoursess` or `t sess yoursess` to trigger **File** > **Sessions** > *yoursessionname*.

By default, the following filters are enabled and you do not need to use their prefixes explicitly:

> Going to a line and column in the current file (`l`).

> Going to an open file (`o`).

> Going to a file in any open project (`a`).

Configuring Locator Filters

If the default filters do not match your use case, you can check whether you can change them. For all filters, you can change the filter prefix and restrict the search to items that match the filter.

To configure a locator filter:

1. In the locator, click ⌕ (**Options**) and select **Configure** to open the **Locator** preferences.
2. Select a filter, and then select **Edit**.
3. Specify the prefix string.
4. To implicitly include the filter even when not typing a prefix as a part of the search string, select **Include by default**.
5. Set other available preferences. For more information, see Adding Web Search Engines.

Adding Web Search Engines

You can use the **Web Search (r)** locator filter to perform web searches. URLs and search commands for Bing, Google, Yahoo! Search, cplusplus.com, and Wikipedia are configured by default.

To find out the format of the search command to use for your favorite web search engine, perform a search in your browser and copy the resulting URL to the locator filter configuration. Replace the search term with the variable %1.

To add URLs and search commands to the list:

1. Select **Edit** > **Preferences** > **Environment** > **Locator** > **Web Search (prefix: r)** > **Edit**.
2. Select **Add** to add a new entry to the list.

Qt **DOCUMENTATION**

Prefix: r   ☐ Include by default

OK   Cancel
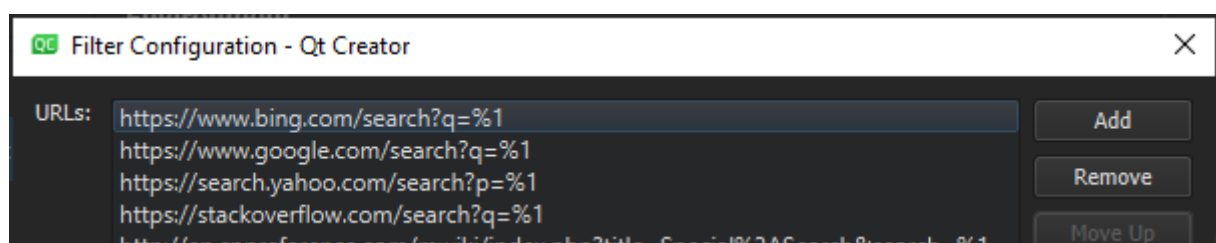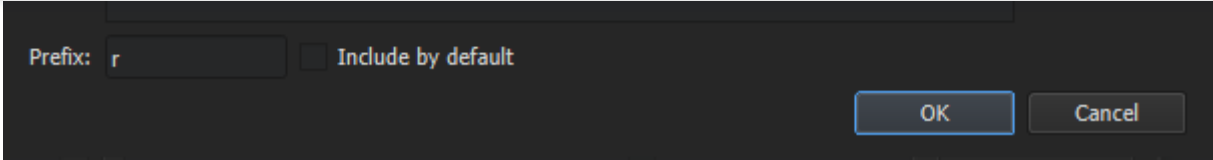
3. Double-click the new entry to specify a URL and a search command. For example,
   `http://www.google.com/search?q=%1`.

4. Click **OK**.

Creating Locator Filters

You can create custom locator filters for finding in a directory structure or on the web.

To quickly access files not directly mentioned in your project, you can create your own directory filters. That way you can locate files in a directory structure you have defined.
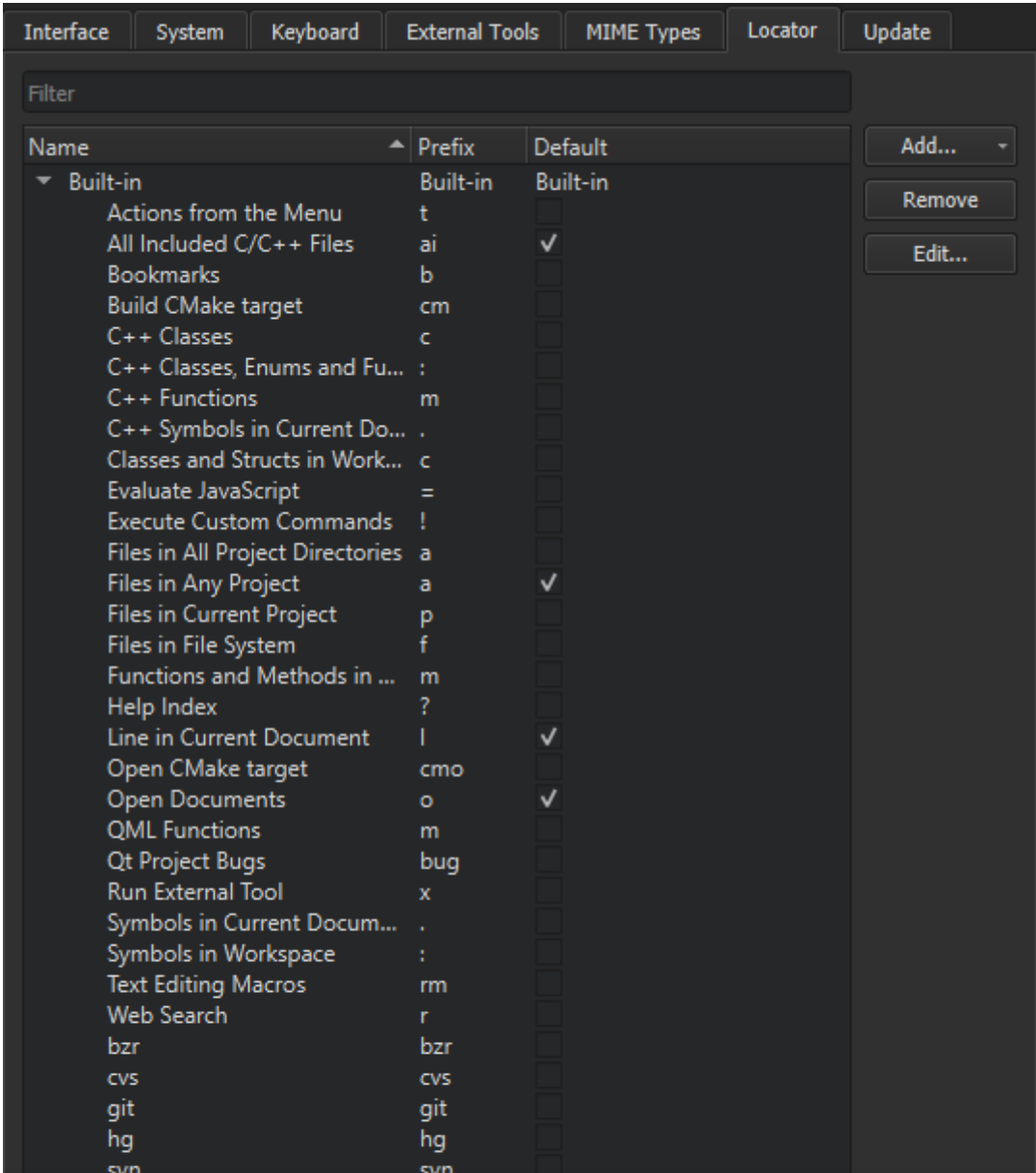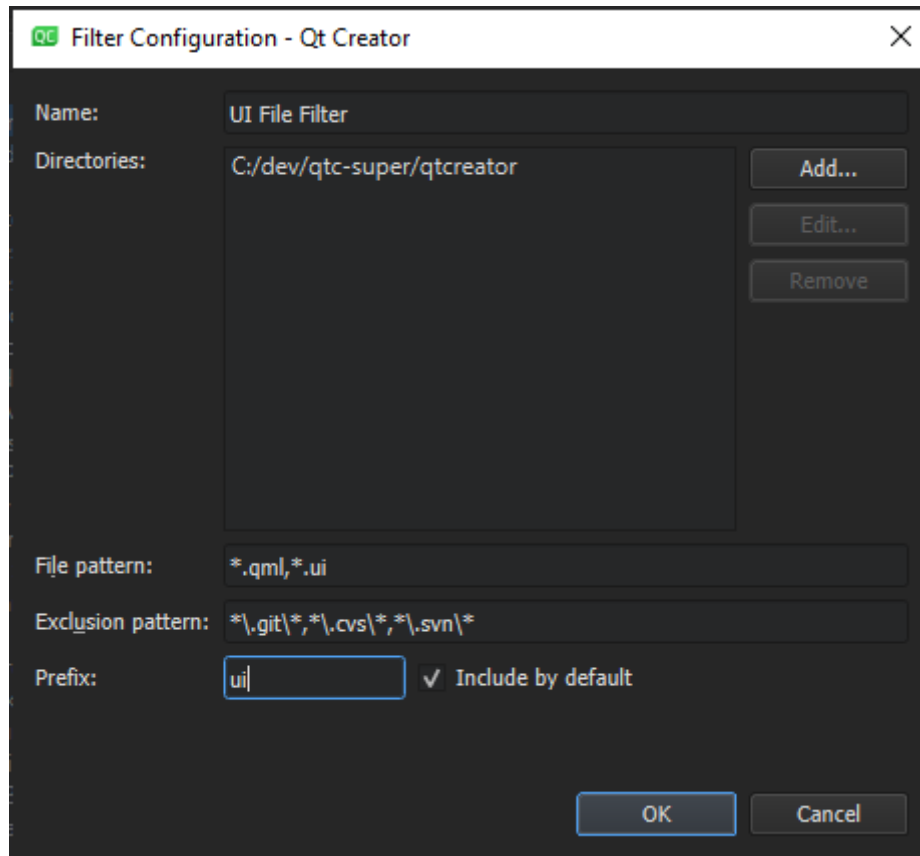
To create custom locator filters:

1. In the locator, select **Options** > **Configure** to open the **Locator** preferences.

| Interface | System | Keyboard | External Tools | MIME Types | Locator | Update |

Name	▲	Prefix	Default		
▼ Built-in		Built-in	Built-in		Add... ▾
Actions from the Menu		t			Remove
All Included C/C++ Files		ai	✓		Edit...
Bookmarks		b			
Build CMake target		cm			
C++ Classes		c			
C++ Classes, Enums and Fu...		:			
C++ Functions		m			
C++ Symbols in Current Do...		.			
Classes and Structs in Work...		c			
Evaluate JavaScript		=			
Execute Custom Commands		!			
Files in All Project Directories		a			
Files in Any Project		a	✓		
Files in Current Project		p			
Files in File System		f			
Functions and Methods in ...		m			
Help Index		?			
Line in Current Document		l	✓		
Open CMake target		cmo			
Open Documents		o	✓		
QML Functions		m			
Qt Project Bugs		bug			
Run External Tool		x			
Symbols in Current Docum...		.			
Symbols in Workspace		:			
Text Editing Macros		rm			
Web Search		r			
bzr		bzr			
cvs		cvs			
git		git			
hg		hg			
svn		svn			

Qt **DOCUMENTATION**

Refresh interval: 60 min

2. Select **Add** > **Files in Directories** to add a directory filter or **URL Template** to add a URL filter. The settings to specify depend on the filter type.

Filter Configuration - Qt Creator

Name:	UI File Filter

Directories: C:/dev/qtc-super/qtcreator

Add...
Edit...
Remove

File pattern: *.qml,*.ui

Exclusion pattern: *\.git*,*\.cvs*,*\.svn*

Prefix: ui ✓ Include by default

OK   Cancel

3. In the **Name** field, enter a name for your filter.

4. In the **Directories** field, select at least one directory. The locator searches directories recursively.

5. In the **File pattern** field, specify file patterns to restrict the search to files that match the pattern. Use a comma separated list. For example, to search for all `.qml` and `.ui.qml` files, enter `*.qml,*.ui.qml`

6. In the **Exclusion pattern** field, specify file patterns to omit files from the search.

7. In the **Prefix** field, specify the prefix string.

   To implicitly include the filter even when not typing a prefix as a part of the search string, select **Include by default**.

8. Select **OK**.

Configuring Locator Cache

The locator searches the files matching your file pattern in the directories you have selected and caches that information. The cache for all default filters is updated as you write your code. By default, Qt Design Studio updates the filters created by you once an hour.

To update the cached information manually, select **Options** > **Refresh** in the locator.

To set a new cache update time:

Qt DOCUMENTATION

2. In **Refresh interval**, define new time in minutes.

Executing JavaScript

The locator provides access to a JavaScript interpreter, that can be used to perform calculations.

Beside simple mathematical operations, like ((1 + 2) * 3), the following built-in functions exist:

Function	Purpose
abs(x)	Returns the absolute value of x
acos(x)	Returns the arccosine of x, in radians
asin(x)	Returns the arcsine of x, in radians
atan(x)	Returns the arctangent of x, in radians
atan2(x, y)	Returns the arctangent of the quotient of its arguments
bin(x)	Returns the binary representation of x
ceil(x)	Returns the value of x rounded up to the next integer
cos(x)	Returns the cosine of x (x is in radians)
exp(x)	Returns the value of E to the power of x
e()	Returns Euler's number E (2.71828...)
floor(x)	Returns the value of x rounded down to the next integer
hex(x)	Returns the hexadecimal representation of x
log(x)	Returns the natural logarithm (base E) of x
max([value1[, value2[, ...]]])	Returns the highest value of the given numbers
min([value1[, value2[, ...]]])	Returns the lowest value of the given numbers
oct(x)	Returns the octal representation of x
pi()	Returns PI (3.14159...)
pow(x, y)	Returns the value of x to the power of y
random()	Returns a random number between 0 and 1
round(x)	Returns the value of x rounded to the next integer
sin(x)	Returns the sine of x (x is in radians)
sqrt(x)	Returns the square root of x
tan(x)	Returns the tangent of x (x is in radians)

DOCUMENTATION

The Qt
Company

Contact Us

Company

About Us

Investors

Newsroom

Careers

Office Locations

Licensing

Terms & Conditions

Open Source

FAQ

Support

Support Services

Professional Services

Partners

Training

For Customers

Support Center

Downloads

Qt Login

Contact Us

Customer Success

Community

Contribute to Qt

Forum

Wiki

Downloads

Marketplace

© 2022 The Qt Company   Feedback   Sign In