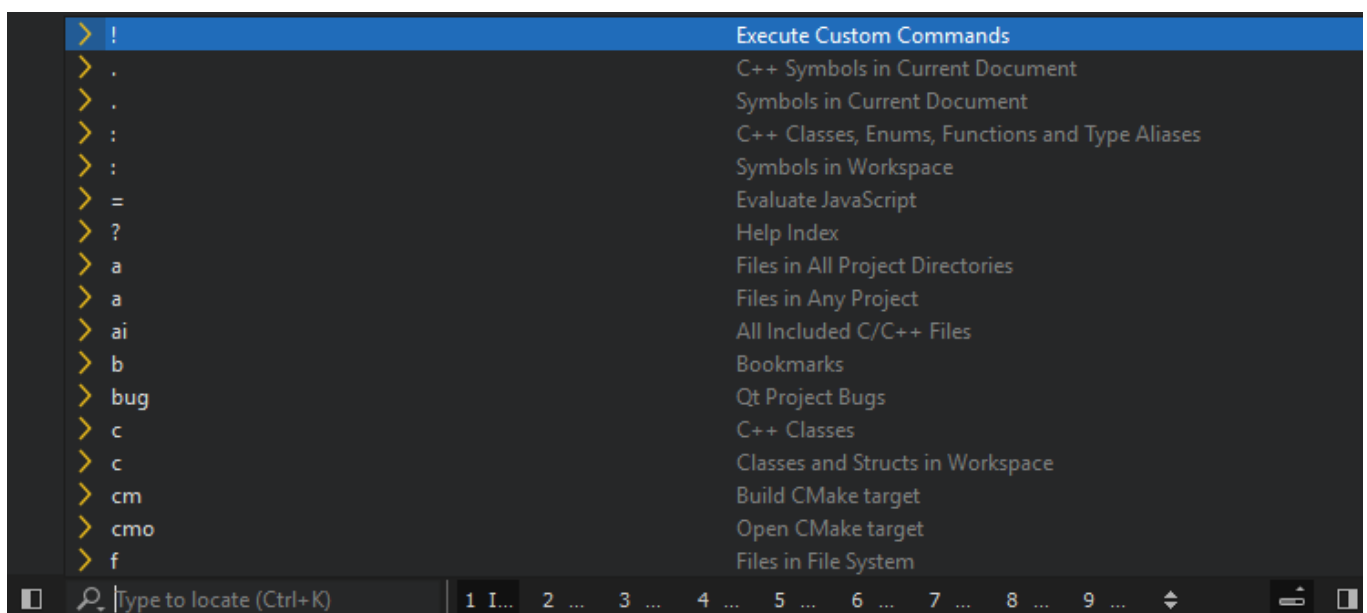


Searching with the Locator

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

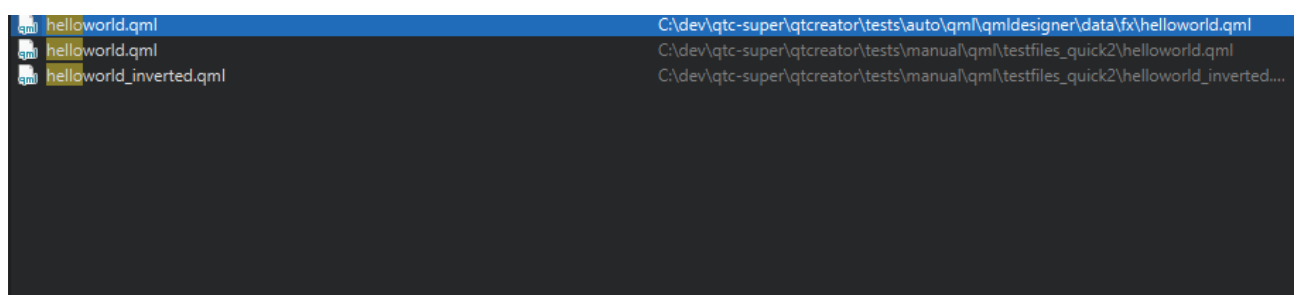


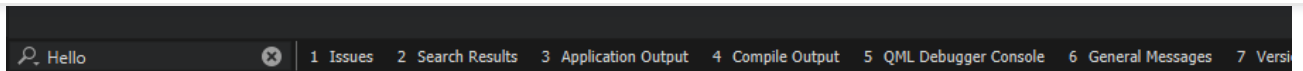
To activate the locator:

- > 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.





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 an 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 bookmarks (b). For more information, see [Using Bookmarks](#).
- › Locating class (c), enum, function (m), and type alias definitions in your project or anywhere referenced from your project (:)
- › 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)
- › Running text editing macros that you record and save (rm). For more information, see [Using Text Editing Macros](#)
- › Executing JavaScript (=), especially useful for calculations. For more information, see [Executing JavaScript](#).
- › Executing shell commands (!)
- › Executing version control system commands (bzs, cvs, git, hg, or svn). For more information, see [Using Version Control Systems](#)

- 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
 - On Windows: using [Everything](#)
 - On Linux: using the `Locate` command
- Running external tools (x)
- Using CMake to build the project for the current run configuration (cm). For more information, see [Setting up CMake](#).
- Opening the CMakeLists.txt file for the current run configuration in the editor (cmo). This is the same build target as when you select **Build > Build for Run Configuration**.

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.

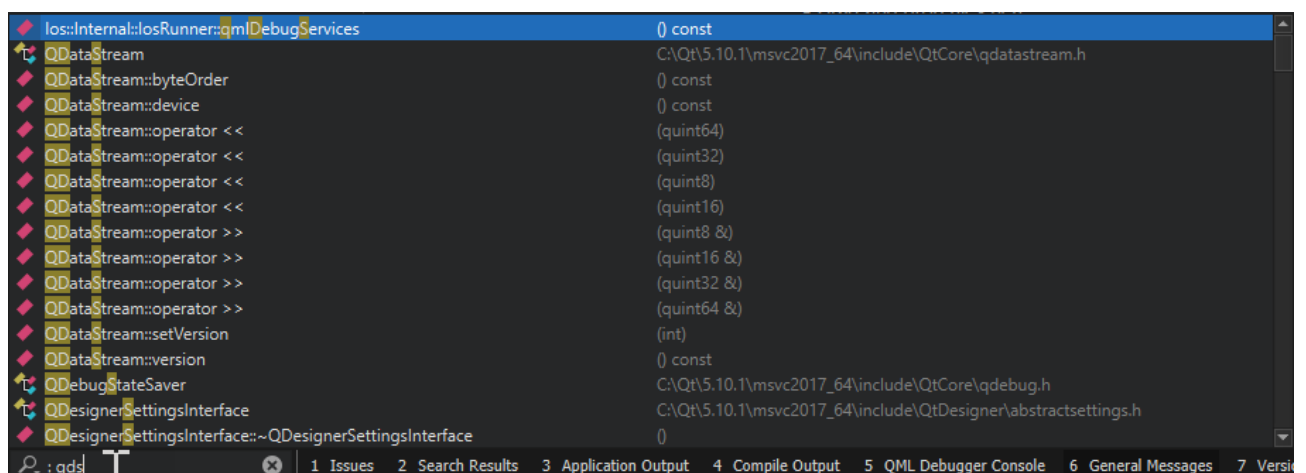
You can also double-click a locator filter in the filter list to use it. You can use the up and down arrow keys or the **Ctrl+P** and **Ctrl+N** keyboard shortcuts to move up and down the list, and then press **Enter** to use the selected filter.

For example, to locate symbols matching [QDataStream](#):

1. Activate the locator.
2. Enter a colon (:) followed by a space and the upper case letters in the symbol name ([QDataStream](#)):

```
: qds
```

The locator lists the results.



Filters locating files also accept paths, such as `tools/*main.cpp`. Filters locating class and function definitions also accept namespaces, such as `Utils::*View`. 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 file in any open project (a).

If locator does not find some files, see [Specifying Project Contents](#) for how to make them known to the locator.

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](#).

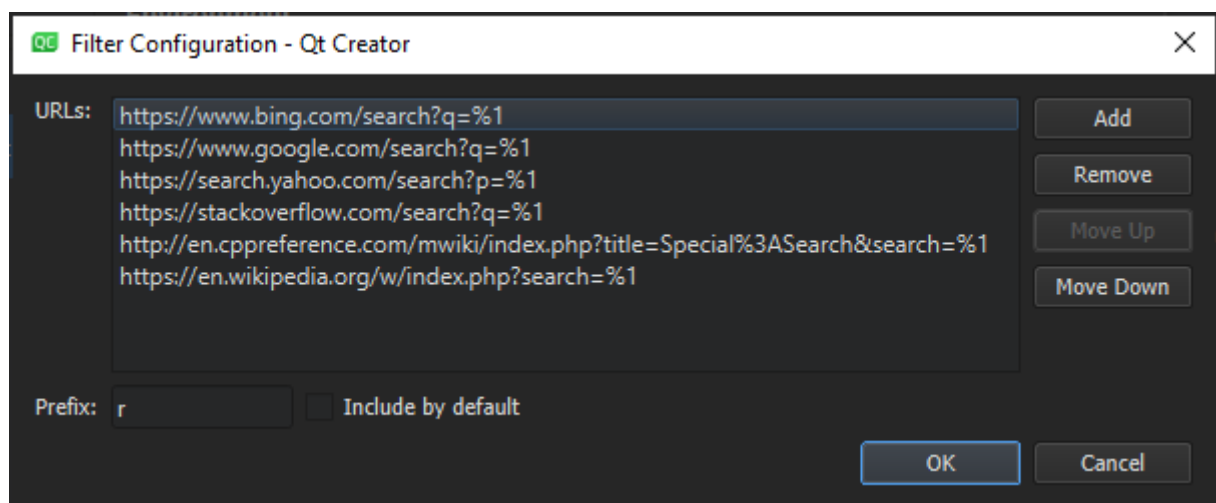
Adding Web Search Engines

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.



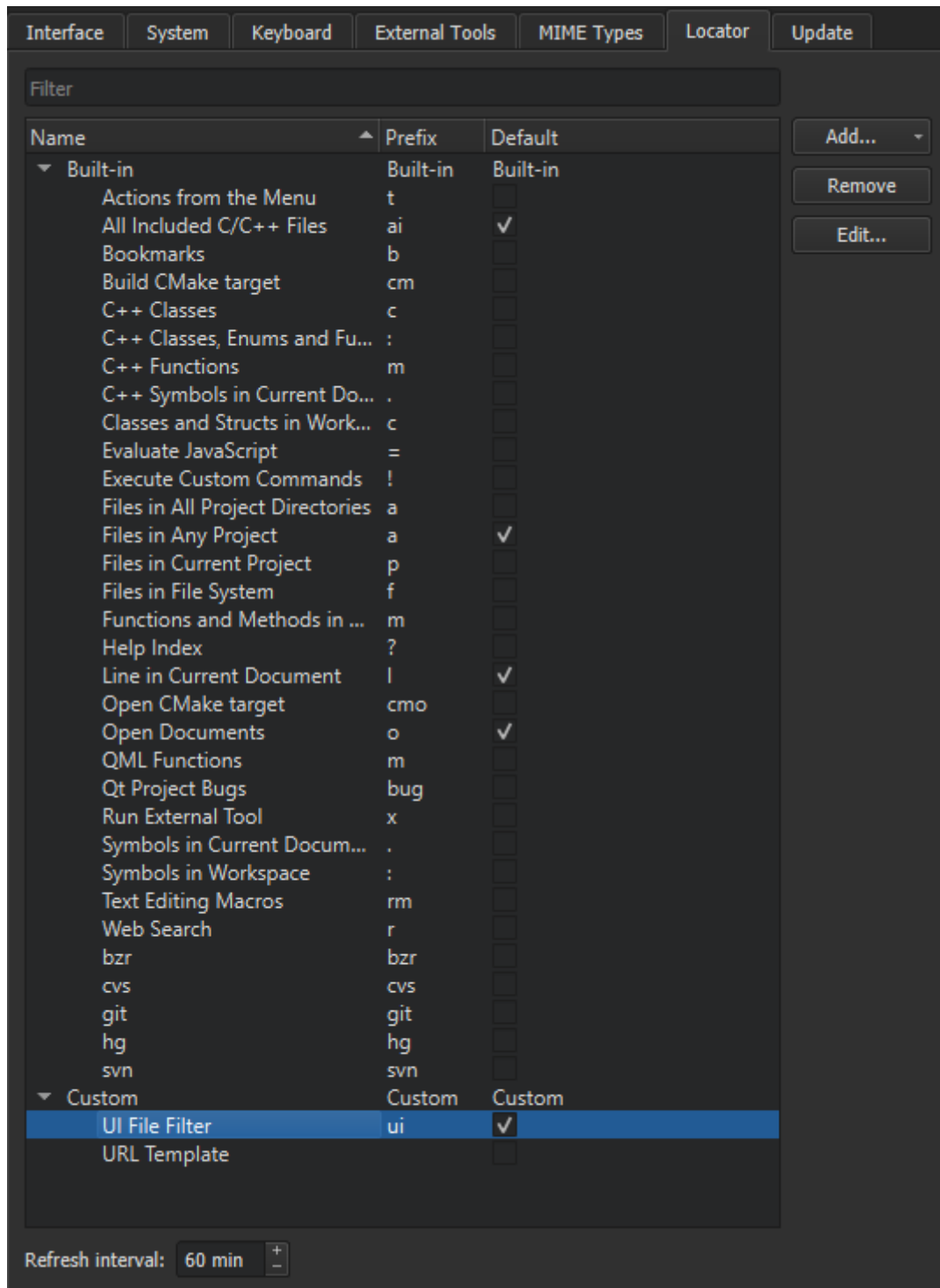
3. Double-click the new entry to specify a URL and a search command. For example, `http://www.google.com/search?q=%1`.
4. Click **OK**.

Creating Locator Filters

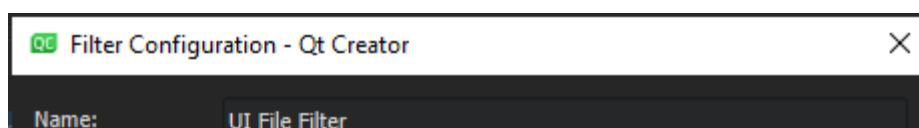
To quickly access files not directly mentioned in your project, you can create your own directory filters. That way you can locate files in a directory structure you have defined.

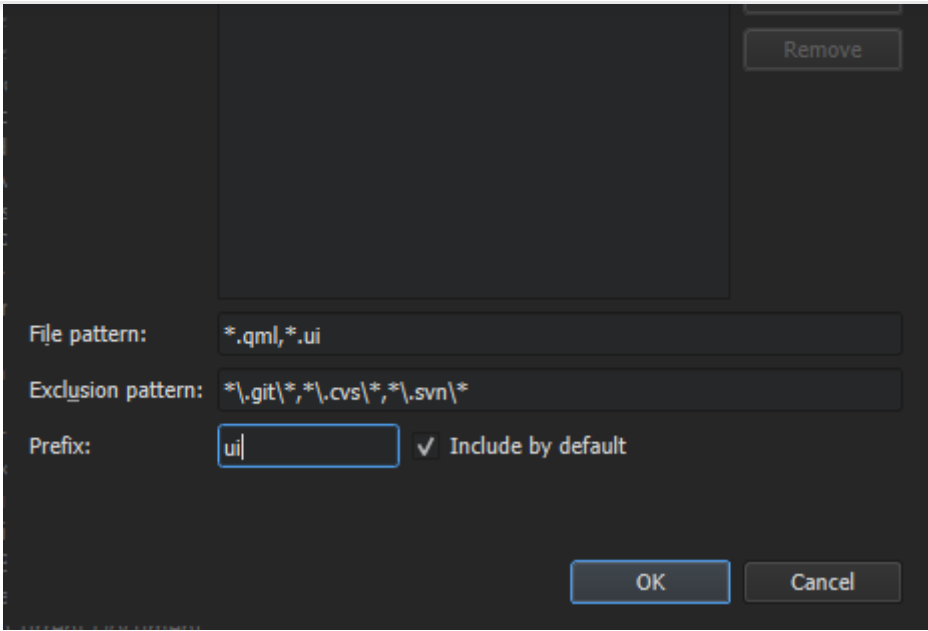
To create custom locator filters:

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



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.





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 Creator 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:

1. Select **Edit > Preferences > Environment > Locator**.
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)	Function	Returns the absolute value of x
		Purpose

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

[< Finding and Replacing](#)
[Refactoring >](#)

© 2022 The Qt Company Ltd. Documentation contributions included herein are the copyrights of their respective owners. The documentation provided herein is licensed under the terms of the [GNU Free Documentation License version 1.3](#) as published by the Free Software Foundation. Qt and respective logos are trademarks of The Qt Company Ltd in Finland and/or other countries worldwide. All other trademarks are property of their respective owners.


[Contact Us](#)
[Company](#)
[Licensing](#)



Newsroom

Careers

Office Locations

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](#)