

在编辑模式下工作

本节介绍如何使用“编辑”模式。有关使用边栏的详细信息，请参阅[浏览项目内容](#)。

使用编辑器工具栏

编辑器工具栏位于编辑器视图的顶部。编辑器工具栏是上下文相关的，并显示与编辑器中当前打开的文件相关的项目。



在打开的文件和符号之间导航

使用工具栏、“窗口”菜单项或[键盘快捷键](#)在打开的文件和正在使用的符号之间导航。

要向后或向前浏览位置记录，请点击 **<**（返回）和 **>**（前进）。

若要返回到进行更改的最后一个位置，请选择“窗口”>“转到上次编辑”。

要转到任何打开的文件，请从**打开文件**下拉菜单（1）中选择该文件。要打开包含用于管理打开的文件的命令的上下文菜单，请右键单击工具栏上的文件名或图标。除了“文件”菜单中还提供的命令外，还可以通过选择“复制完整路径”、“复制文件名”或“复制路径和行号”，将当前文件的路径和名称以及光标当前所在的行号复制到剪贴板。

要跳转到当前文件中使用的任何符号，请从符号下拉菜单（2）中选择该符号。默认情况下，符号按它们在文件中出现的顺序显示。右键单击菜单标题，然后选择“按字母顺序排序”以按字母顺序排列符号。

注意： 其他在Qt Creator中导航的便捷方式由**侧边栏**提供。

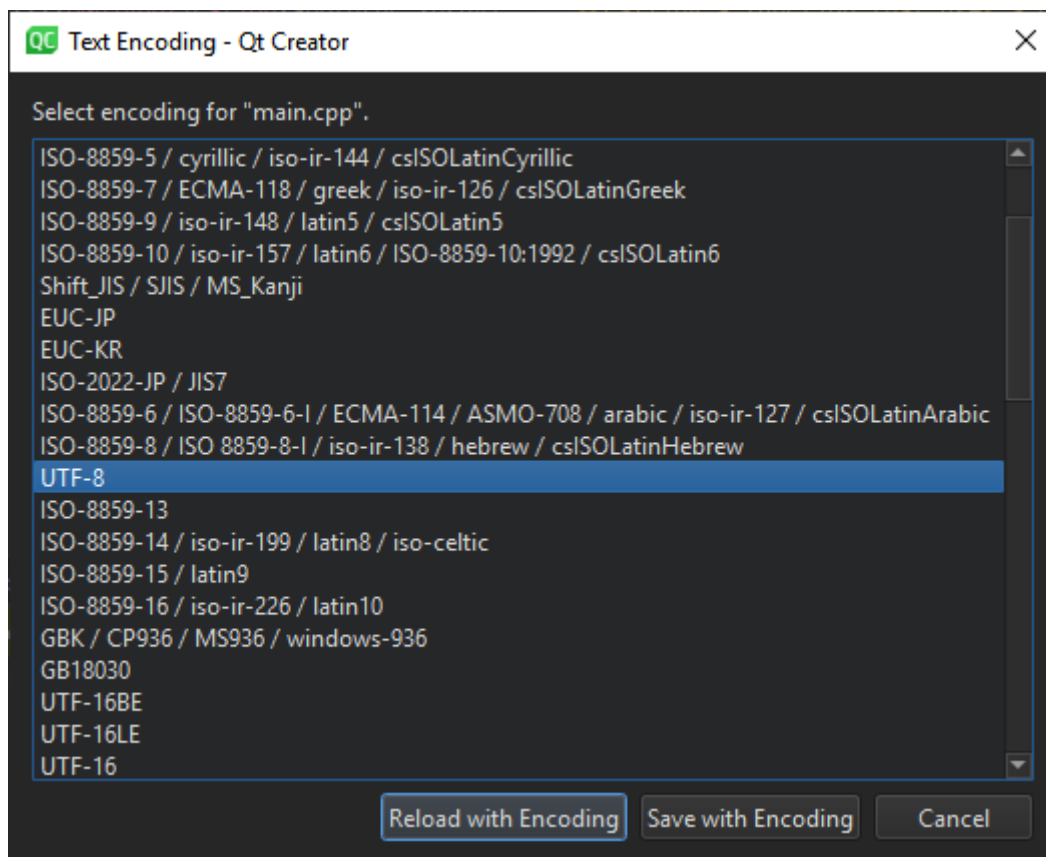
选择解析上下文

在不同的上下文中，代码可能有不同的解释。文件可以由具有不同定义的不同项目或子项目使用，也可以包含在C、C++、目标C或目标C++的上下文中。要更改活动分析上下文，请在“活动分析上下文”菜单（4）中选择可用的**分析上下文**。仅当多个解析上下文可用时，该菜单才可见。若要重置分析上下文，请右键单击菜单以打开上下文菜单，然后选择**清除首选分析上下文**。如果分析项目所需的信息仍然不完整或不正确，请选择**其他预处理器指令**以添加预处理器指令。

更改文本编码

要在编辑器工具栏（5）上显示当前文件的文件编码，请选择**编辑>首选项>“文本编辑器”>显示>显示文件编码**。

要更改文本编码，请在工具栏上单击它，然后在“文本编码”对话框中选择新**编码**：



To reload the file with the selected encoding, select **Reload with Encoding**. To save the file with the new encoding, select **Save with Encoding**.

Selecting Line Ending Style

To switch between Windows line endings (CRLF) and Unix line endings (LF), select the ending style on the editor toolbar (6). To hide this field, select **Edit > Preferences > Text Editor > Display**, and deselect **Display file line ending**.

To set the line endings to use for all projects by default, select **Edit > Preferences > Text Editor > Behavior**, and then select the ending style in the **Default line endings** field.

Editing Selected Lines

The **Edit > Advanced** menu contains options for editing selected lines of text.

To duplicate the selected lines, select **Duplicate Selection**. To format the duplicated lines as a comment, select **Duplicate Selection and Comment**.

To turn selected text into lowercase, select **Lowercase Selection** or press **Alt+U**. To turn it into uppercase, select **Uppercase Selection** or press **Alt+Shift+U**.

To sort selected lines alphabetically, select **Sort Selected Lines** or press **Alt+Shift+S**.

Select **Add Next Occurrence to Selection** or press **Ctrl+D** to add a cursor at the next occurrence of selected text for multi-cursor editing.

Multi-Cursor Editing

To apply a change to several places simultaneously, press and hold **Alt**, and click to place cursors in several places. Any changes you make are applied simultaneously at all the cursor positions.

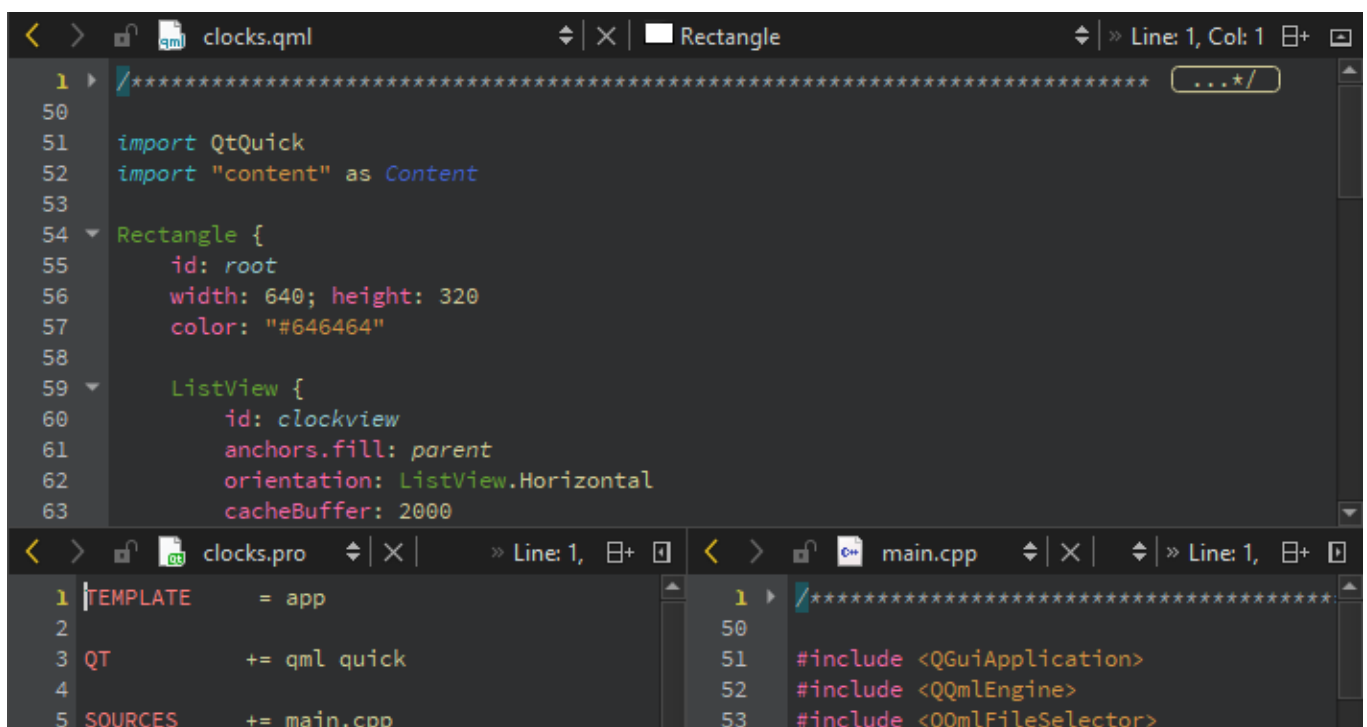
Use the arrow keys to move all the cursors up and down. The **Home** and **End** key move all the cursors to the beginning or to the end of the line.

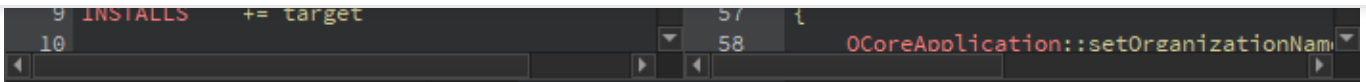
Press and hold **Alt** and double-click strings to select several strings simultaneously.

Press **Esc** to remove all the cursors and selections.


Splitting the Editor View

Split the editor view or open the editor in a new window when you want to work on and view multiple files on the same screen or on multiple screens.





You can view multiple files simultaneously in the following ways:

- To split the editor view into a top and bottom view, select **Window > Split**, press **Ctrl+E, 2**, or select the  (Split) button and then select **Split**.

Split command creates views below the currently active editor view.

- To split the editor view into adjacent views, select **Window > Split Side by Side**, press **Ctrl+E, 3**, or select **Split > Split Side by Side**.


Side by side split command creates views to the right of the currently active editor view.

- To open the editor in a detached window, press **Ctrl+E, 4**, or select **Window > Open in New Window**.

The new window behaves basically in the same way as the editor area in the main window. For example, you can split this window, as well. Documents are opened in the currently active editor window.

To move between split views and detached editor windows, select **Window > Go to Next Split** or **Window** or press **Ctrl+E, O**.

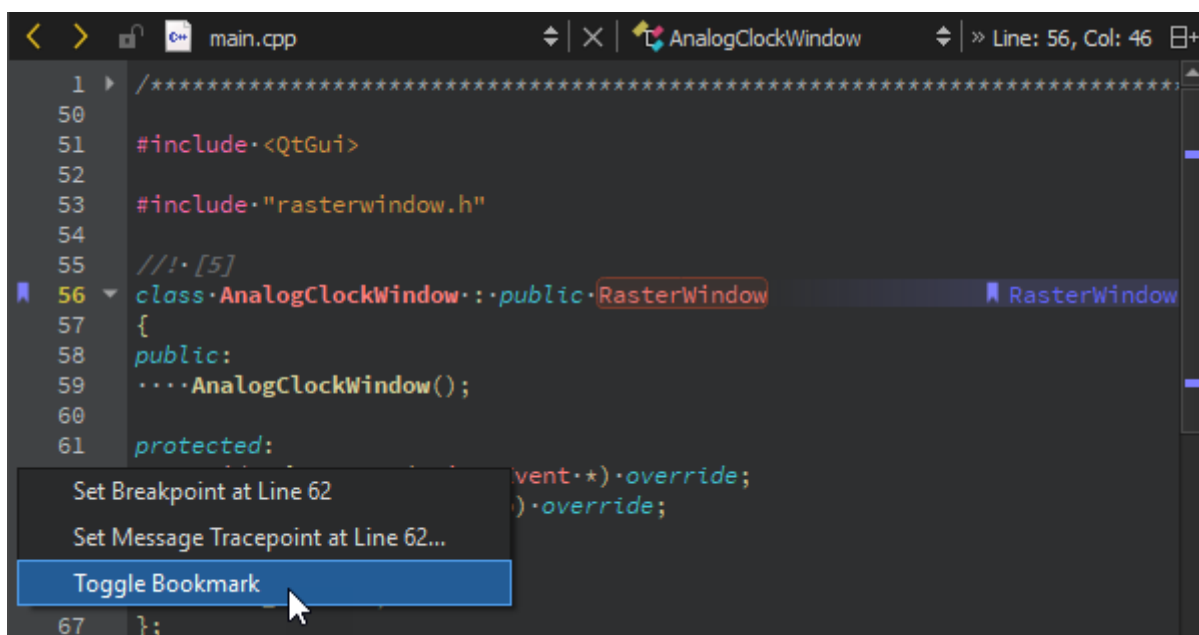
To remove a split view, place the cursor within the view you want to remove and select **Window > Remove Current**

Split, press **Ctrl+E, 0**, or select the  (Remove Split) button. To remove all but the currently selected split view, select **Window > Remove All Splits** or press **Ctrl+E, 1**.

Using Bookmarks

To insert or delete a bookmark in the **Edit** mode:

- Right-click the line number and select **Toggle Bookmark**.
- Press **Shift** and click the left margin at a line.
- Press **Ctrl+M** when the cursor is on a line.



To add a note to a bookmark:

- Select **Tools > Bookmarks > Edit Bookmark**.
- Press **Ctrl+Shift+M**.
- Right-click a bookmark and select **Edit Bookmark** in the context menu.

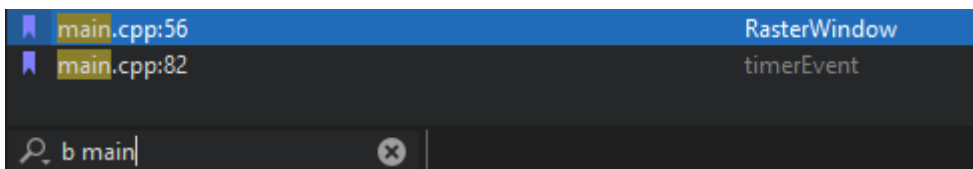
To view the note, move the mouse pointer over the bookmark or open the **Bookmarks** view in the [sidebar](#).

Navigating Bookmarks

To go to the previous bookmark in the current session, select **Tools > Bookmarks > Previous Bookmark** or press **Ctrl+,**.

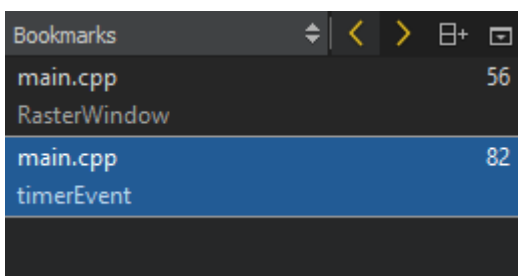
To go to the next bookmark in the current session, select **Tools > Bookmarks > Previous Bookmark** or press **Ctrl+,**.

To use the locator to go to a bookmark, press **Ctrl+K** (or **Cmd+K** on macOS) to open the locator. Enter *b* and a space to display a list of bookmarks. To filter the bookmarks by line number or a text string, enter the number or string after the space. Double-click a bookmark in the list to go to it in the editor.



Viewing Bookmarks

Bookmarks are listed in the **Bookmarks** view in the sidebar. To move between bookmarks, select the **Previous Bookmark** or **Next Bookmark** button or use the keyboard shortcuts.



Moving to Symbol Definition or Declaration

You can move directly to the definition or the declaration of a symbol in the **Edit** mode by holding the **Ctrl** key and clicking the symbol. If you have multiple splits opened, you can open the link in the next split by holding **Ctrl** and **Alt** while clicking the symbol.

To enable this moving function, select **Edit > Preferences > Text Editor > Behavior > Enable mouse navigation**.

There are several additional ways of moving between symbol definitions and declarations. All the functions described below are also available from the **Tools > C++** menu. The functions supported for QML and JavaScript code are available from the **Tools > QML/JS** menu.

You can select the symbol and press **F2**, or right-click the symbol and select **Follow Symbol Under Cursor** to move to its definition or declaration. To follow the symbol in the next split, select **Follow Symbol Under Cursor in Next Split**. Following symbols is supported for namespaces, classes, functions, variables, include statements, and

right-click and select **Switch between Function Declaration/Definition or Open Function Declaration/Definition in Next Split**. For example, this allows you to navigate from anywhere within a function body directly to the function declaration.

Links are opened in the same split by default. To open links in the next split, prepend **Ctrl+E** to the shortcut. For example, press **Ctrl+E,F2** to follow the symbol in the next split. If necessary, the view is automatically split. To change the default behavior, select **Edit > Preferences > Text Editor > Display > Always open links in another split**. Additional symbols are displayed and switching between definition and declaration is done in another split. If you change the default behavior, the shortcuts for opening link targets in the next split are used to open them in the current split.

To switch between C++ header and source files, right-click anywhere in a file and select **Switch Header/Source** or **Open Corresponding Header/Source in Next Split**. You can also press **F4** or **Ctrl+E,F4**, respectively.

Reparsing Externally Changed Files

If source files are modified from outside Qt Creator, the opened files will be reparsed automatically. For all other files, you can use **Tools > C++ > Reparse Externally Changed Files** to update the code model.

Inspecting the Code Model

When you [report a bug](#) that is related to the C++ code model, the Qt Creator developers might ask you to write information about the internal state of the code model into a log file and to deliver the file to them for inspection.

To view information about the C++ code model in the **C++ Code Model Inspector** dialog and write it to a log file, select **Tools > Debug Qt Creator > Inspect C++ Code Model** or press **Ctrl+Shift+F12**.

Qt Creator generates the code model inspection log file in a temporary folder.

Qt Creator underlines semantic errors in olive in the C++ code editor. To check the correct paths for includes that are not resolved or that are resolved to the wrong file, select **Project Parts > Header Paths**.

[< Writing Code](#)

[Semantic Highlighting >](#)

© 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](#)