Writing Code

The Qt Creator code editor is fully equipped with semantic highlighting, syntax checking, code completion, code indentation, context sensitive help, and in-line error indicators while you are typing.

- Working in Edit Mode

  You can use the editor toolbar to navigate between open files and symbols in use. You can also split the view to work on several files simultaneously, add bookmarks, and move between symbol definitions and declarations.

- Semantic Highlighting

  Qt Creator enables you to write well formatted code by highlighting code elements and blocks. You can use syntax highlighting also for other types of files than C++ or QML.

- Checking Code Syntax

  Qt Creator checks for errors when you write code and displays inline error and warning messages. Similarly, it checks the data structure of an instance of a JavaScript object notation (JSON) entity. In addition, you can run static checks on the QML and JavaScript code in your project to find common problems.

- Completing Code

  Qt Creator anticipates what you are going to write and completes code and code snippets for elements, properties, and IDs.

- Indenting Text or Code

  Qt Creator indents text and code according to rules that you specify separately for files that contain C++, QML, or Nim (experimental) code and for other text files.

- Using Qt Quick Toolbars

  When you edit QML code in the code editor, you specify the properties of QML components. For some properties, such as colors and font names, this is not a trivial task. For example, few people can visualize the color #18793f. To easily edit these properties, you can use the Qt Quick Toolbars.

- Pasting and Fetching Code Snippets

  You can cooperate with others by pasting and fetching snippets of code from a server. For example, you might ask colleagues to review a change that you plan to submit to a version control system.

- Using Text Editing Macros

  When you have a file open in the code editor, you can record a keyboard sequence as a macro. You can then play the macro to repeat the sequence. You can save the latest macro and assign a keyboard shortcut for running it or run it from the locator.

Related Topics

- Comparing Files

  You can use a diff editor to compare two versions of a file and view the differences side-by-side in the **Edit** mode.

- Parsing C++ Files with the Clang Code Model

  The Clang code model provides some of the services previously provided by the built-in C/C++ code model, such as code completion, syntactic and semantic highlighting, diagnostics, tooltips, outline of symbols, and renaming of local symbols.

‹ Coding

Working in Edit Mode ›