Coding

Writing Code

Writing, editing, and navigating in source code are core tasks in application development. Therefore, the code editor is one of the key components of Qt Creator. You can use the code editor in the **Edit** mode.

Finding

Use the incremental and advanced search to search from currently open projects or files on the file system or use the locator to browse through projects, files, classes, functions, documentation and file systems.

Refactoring

Code refactoring is the process of improving and simplifying code without modifying the existing functionality of an application. You can easily find and rename symbols and apply predefined actions to refactor code.

Beautifying Source Code

Beautifying code means applying indentation and style to source code files. You can use the Artistic Style, ClangFormat, or Uncrustify tool to format source files.

Configuring the Editor

You can change the fonts, colors, highlighting, and indentation. If you are used to the Vim editor, you can even run the main editor in a manner similar to it in the **FakeVim** mode.

Related Topics

Using Language Servers

The language client provides code completion, highlighting of the symbol under cursor, and jumping to the symbol definition for other programming languages besides C++. In addition, it integrates diagnostics from the language server.

Editing MIME Types

Modeling

You can use the model editor to create Universal Modeling Language (UML) style models with structured and behavioral diagrams that provide different views of your system and store them in XML format.

Editing State Charts

You can use Qt Creator to create applications that embed state machines. A project wizard creates State Chart XML (SCXML) files with boilerplate code that you can edit using an experimental SCXML editor. You can use the classes in the Qt SCXML module to embed state machines created from the files in Qt applications.

‹ Optimizing Applications for Mobile Devices

Writing Code ›