Volume 4, Issue 11, November 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Study of Design and Development of Integrated Development Environment using Qt C++ Framework

**Parul*, Jagandeep Sidhu**
Computer Science and Engineering,
Chitkara University, Punjab, India

***Abstract— This article is concerned with the construction and development of a tool known as the integrated development environment (IDE) using Qt C++ framework. An IDE is computer software which basically includes source code editor, a compiler or interpreter or both, build automation tools and a debugger and provides comprehensive facilities to the programmers for the development of the software. The objective function to develop an IDE is to enhance learning efficiency of the beginners as well as helpful tool for the professionals by making it more interactive using Qt C++ framework and it will present a single program where all the development is done which further results in greater productivity. This interactive IDE provides a platform for programming developers to develop various applications on languages like C and C++, also help to reduce the cost of programmer cultivating and shorten the time of technological learning.***

***Keywords— Integrated Development Environment; C Language; C++ Language; Qt C++ Framework; Interactive***

## I. INTRODUCTION

Qt Creator is a cross-platform IDE written in C++ which includes extensive C++ libraries that helps developers to efficiently design and build highly performing applications and user interfaces for several desktop and mobile operating systems like Linux, Mac OS X and Windows operating systems without rewriting the source code [1][2][3]. The IDE with Qt C++ framework can help the programming beginners as well as professionals to fastly grasp the basic language grammars and the drawing functions of languages like C, C++ as it support multimedia and graphics [4]. Developing modern GUI applications with Qt C++ framework is fast, simple and straightforward, and can be achieved by hand coding or by using Qt Designer, Qt's visual design tool. Companies and independent developers like Adobe, Boeing, Google, IBM, Motorola, NASA, Skype and numerous smaller companies and organizations from around the world are joining the Qt development community every day [1][2]. There has been a steady increase in the adoption of Qt Creator. In many places Qt Creator has become the primary IDE for software development.

## II. FUNCTIONS OF QT CREATOR

Functions of Qt Creator are shown as Fig. 1.

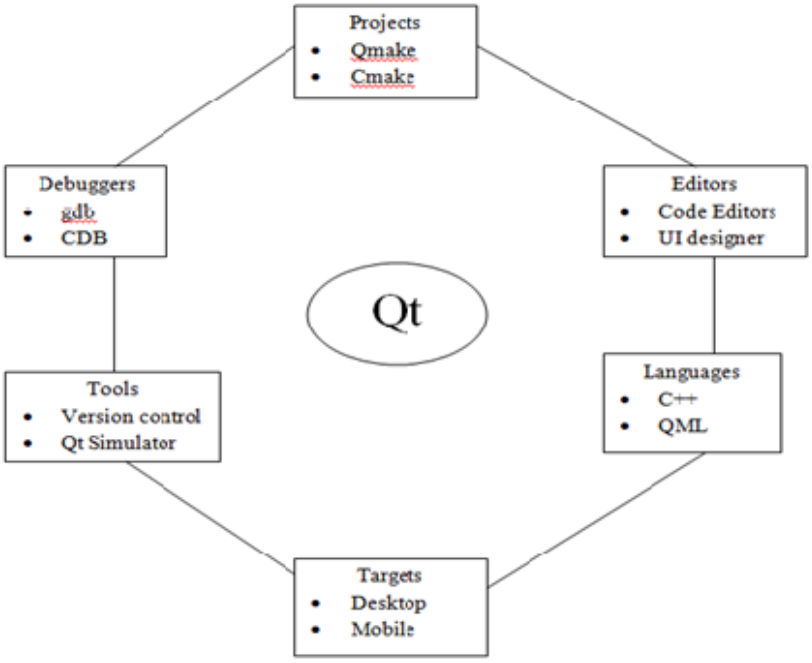



Fig. 1 Functions of Qt Creator

### *A. Projects*

Either a new project can be started or an existing project is imported. Qt Creator automatically integrates all the necessary files according to the type of project. For creating a project, one can group files together, add custom build steps, include resource files and specify settings for running applications. For Graphical User Interface (GUI) application, Qt Creator generates an empty file with the extension .ui that can be modified with the help of Qt Designer. Qt Creator is integrated with two build automation tools Qmake and Cmake.

### *B. Editors*

Qt Creator basically consists of two types of editor 1 Code Editor, 2 Qt Designer. Both these editors are used for designing and building GUI's using widgets. Code Editor is very much different from a text editor. It not only understands plain text but also understands the C++ and QML languages as code. It enables us to write well formatted code as it display inline error and warning messages.

### *C. Languages*

Code editor of Qt Creator can be used to write code in Qt C++ or in the QML programming language. QML declarative language can be used to develop highly dynamic and custom user interfaces using rich set of QML elements.

### *D. Targets*

Qt Creator is used to develop, compile and run Qt applications for several target types such as desktop environment (Windows, Linux and Mac OS) and mobile devices.

### *E. Tools*

Two tools Version Control Systems and Qt Simulator are included in Qt Creator. Version Control Systems are used for comparing files with the latest versions stored in the repository. Qt Simulator is used to test Qt applications as it is installed as a part of the Qt Software Development Kit (SDK).

### *F. Debuggers*

Qt includes a debugger plug-in instead of a debugger that acts as an interface between the Qt Creator core and external native debuggers (GNU Symbolic Debugger (gdb) and Microsoft Console Debugger (CDB)) [5].

## III. FEATURES OF QT CREATOR

Qt Creator is easy to install and use. It consists of complete tool chain as it is integrated with GUI layout, forms designer and a visual debugger. It supports multiple compilers like on windows it can use Minimalistic GNU for Windows (MinGW) or Microsoft Studio Visual C++ (MSVC) and on Linux it can use GNU Compiler Collection (GCC). It is extensible through various plug-ins such as syntax highlighting and code completion, static code checking and style hints, context sensitive help, code folding, parenthesis matching. It includes all the features mandatory to develop modern GUI applications with menus, toolbars and dock windows. It supports two types of document interface 1 Single document interface and multiple user interfaces. It is used to store application settings using system registry or text files, allowing items such as user preferences, most recently used files, window and toolbar positions and sizes to be recorded for later use. It also supports multithreading programming which is provided by a collection of classes that is used to represent common constructs, making it possible to write Qt applications that take advantage of threads to perform calculations, long duration tasks, or just to improve responsiveness [6][7].

## IV. APPLICATIONS DEVELOPED ON QT CREATOR

The Application Example is shown as Fig. 2.

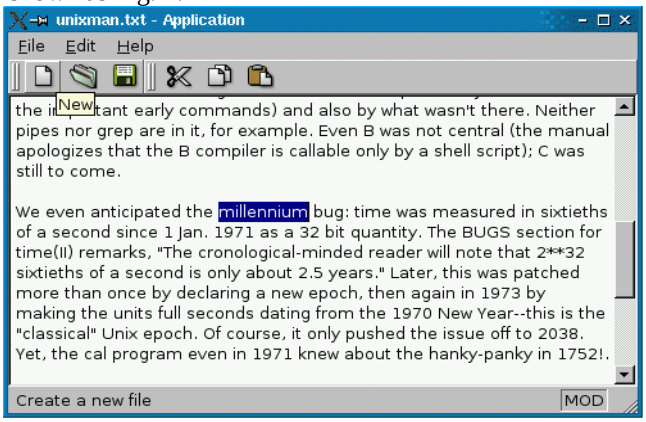



Fig. 2 The Application Example

The Application example depicts how to implement a standard GUI application with menus, toolbars, and a status bar. It is a simple text editor program develops using Plain Text Edit which is an advanced editor used to handle large documents and responds quickly to user input [8].

The Code Editor Example is shown as Fig. 3.

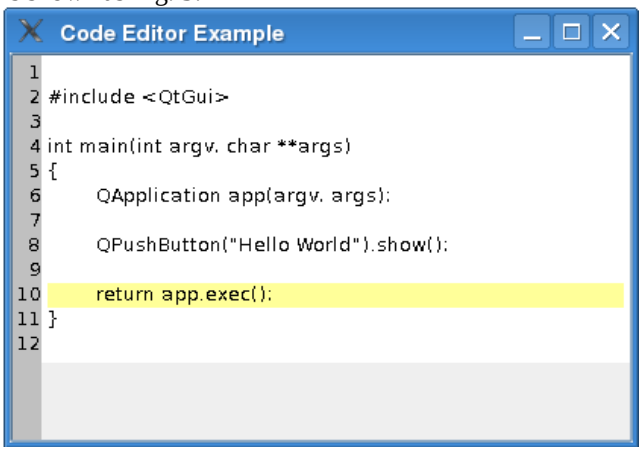
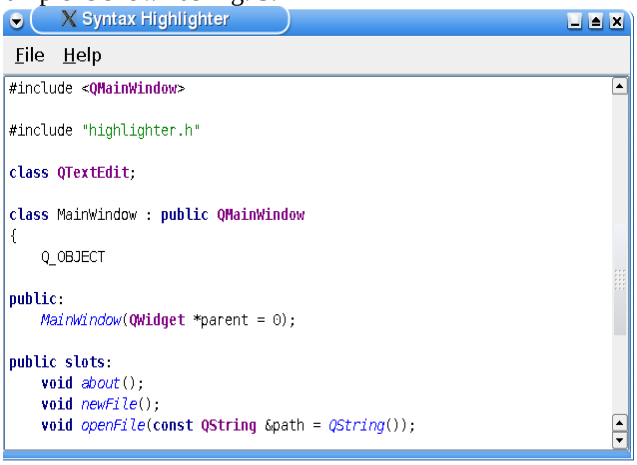



Fig. 3 Code Editor Example

The Code Editor Example is a simple editor with consecutive line numbers and highlights the current line. As can seen from Fig. 3, the Code Editor Example displays the line numbers in an area to the left of the area for editing and also highlight the line containing the cursor [9].

The Syntax Highlighter Example is shown as Fig. 3.




Syntax Highlighting is used to highlight parts of the text written on a plain text editor by using certain colours like blue is for keywords, orange is for operators. It helps the user to read the text and identify syntax errors [10].

## V. GUI COMPONENTS

Qt Creator contains a rich set of standard widgets that are used to create GUI for applications. Widget includes Layouts, Buttons, Items Views, Containers, Input Widgets, and Display Widget.

### *A. Layouts*

Layouts are used to align or displayed all the objects in a proper manner whenever form is previewed or used in an application. They will also ensure that whenever the form is resized, all the objects placed in a form must also be resized correctly. Layouts are divided into four parts 1 Horizontal and Vertical Layout, 2 Grid Layout, 3 Splitter Layout and 4 Form Layout. Horizontal and vertical layouts ensure that widgets within are aligned horizontally and vertically respectively. Horizontal and vertical layouts are the simplest way to arrange objects on a form. Grid layout provides freedom to designer to arrange widgets on the form. This kind of layout is complex than horizontal and vertical layouts. Splitter Layouts arrange objects on the form either in horizontal layout or in vertical layout but also allow user to adjust the amount of space allocated to each object. Form Layout handles widgets by using two-column form, one is used to hold labels and another is used to hold field widgets such as line edits.

### *B. Buttons*

Buttons are of six types Push Button, Tool Button, Radio Button, Check Box, Command Link Button, and Button Box. Push Button are used to provide events like clicked, pressed, released, toggled and destroyed to perform specific actions. Tool Button are used to provide events like clicked, pressed, released, toggled, destroyed and triggered to perform specific actions. A radio Button allows user to select the desired and single choice out of many choices. Check Boxes allows user to select multiple choices out of many choices. Command Link Button has same events as that of a Push Button and whenever user press a command link button, it will redirect to a different page just like in a web browser. Button Box is used to combine two buttons in a single box in a situation where a choice has to make out of two.

### *C. Item Views*

Items Views are of two types model based and item based. Model based are further divided into four parts List View, Tree View, Table View, Column View. In a List View, multiple items displayed as a list and user can select one or more option. Tree View is used to display the data in a hierarchical form such as file directory. Table view is to display the data in the form of a table. List Widget is item based which is used to edit the list items manually.

### *D. Containers*

Containers are of eight types Group Box, Scroll Area, Tool Box, Tab Widget, Stacked Widget, Frames, and Multiple document interface area. Group Box is a collection of checkboxes and radio buttons with similar purposes. Scroll Area provides horizontal and vertical scroll bars which allows user to access parts of document that is greater than the widgets. Tool Box contains a series of pages or compartments. Tab Widget is used to divide the content of a widget into different sections but only one section is viewed at any given time. Stacked Widgets contains one or more widgets in which only the topmost layer is visible. Frames are the placeholder in which widgets like Plain text Edit, Label, Graphics View, Horizontal Scroll Bar, Vertical Scroll Bar etc can be placed. Multiple document interfaces is an area in which multiple child windows reside a single parent window.

### *E. Input Widgets*

Input Widgets includes Combo Box, Line Edit, Text Edit, Plain Text Edit, and Horizontal and Vertical Scroll Bar. Combo Box is a combination drop-down list box and a single-line editable textbox. Line Edit is single-line edit box which is used for displaying data input. Text Edit is multi-line edit box which is used for displaying data input. Plain Text Edit is an advanced editor used to handle large documents and responds quickly to user input. Horizontal Scroll Bar and Vertical Scroll Bar allows user to access parts of document that is greater than the widgets.

### *F. Display Widgets*

Display Widgets includes label which is used to display text or image [1] [2].

## VI. PROBLEM DEFINITION

The objective function for developing an IDE using Qt C++ framework is to enhance learning efficiency of the beginners as well as helpful tool for the professionals by making it more interactive by adding a widget Graphics View. Graphics View will display the address allocated to each variable as well as value of each variable using graphical items like rectangle.

## VII. DESIGN AND DEVELOPMENT OF THE USER INTERFACE

The Graphical User Interface (GUI) is layered out as Fig. 4 after abstracting [4].

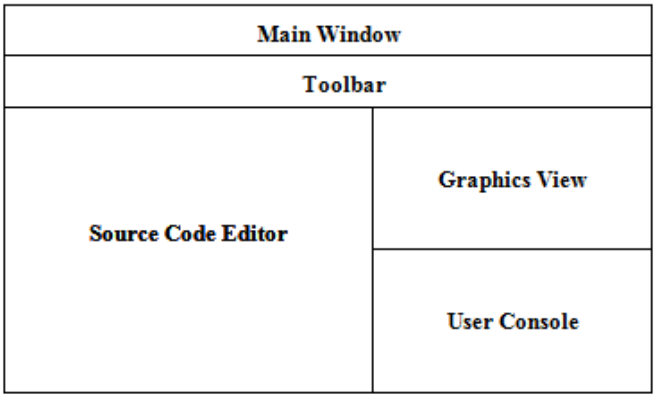



Fig. 4 Graphical User Interface layout

GUI of the IDE is formed by using two layouts Vertical and Horizontal and by some widgets such as Frame, Plain Text Edit, Label, Graphics View, Horizontal Scroll Bar, Vertical Scroll Bar. Horizontal and vertical layouts ensure that widgets within are aligned horizontally and vertically respectively [11]. Frames are the placeholder in which widgets like Plain text Edit, Label, Graphics View, Horizontal Scroll Bar, Vertical Scroll Bar etc can be placed [12]. Plain Text Edit is an advanced editor used to handle large documents and responds quickly to user input [13]. Label is used to display text or image [14]. Graphics View is used for managing large number of 2D graphical items [15]. Horizontal Scroll Bar and Vertical Scroll Bar allows user to access parts of document that is greater than the widgets [16].

## VIII. CONCLUSIONS AND FUTURE WORK

In this research work the IDE is developed and constructed using Qt C++ framework to become more interactive. The IDE will enhance learning efficiency of the beginners as well as helpful tool for the professionals. The IDE will have the ability to develop high-end as well as low-end programs quickly. It will help the beginners to fastly grasp the basic grammar of the language. It reduces the cost of programmer cultivating and shorten the time of technological learning [4][17][18][19]. With further studies and efficient programming, GUI can be modified.

**REFERENCES**

[1] *Qt 4.6 Whitepaper*, Nokia Corporation, 2009.
[2] *Qt 3.3 Whitepaper*, Trolltech.
[3] (2014) Qt Creator Manual. [Online]. Available: http:// qt-project.org/doc/qtcreator-2.8/
[4] Hu Yanju, "Design and Development of C Language Integrated Development Environment with Data Accessing," *International Forum on Information Technology and Applications*, pp. 46–48, 2010.
[5] (2014) Qt Creator. [Online]. Available: http:// developer.nokia.com/community/wiki/Qt_Creator
[6] (2014) IDE Overview. [Online]. Available: http://qt-project.org/doc/qtcreator-2.8/creator-overview.html
[7] (2014) Qt Creator. [Online]. Available: http:// qt-project.org/wiki/Category:Tools::QtCreator
[8] (2014) Application Example. [Online]. Available:http:// qt-project.org/doc/qt-4.8/mainwindows-application.html
[9] (2014) Code Editor Example. [Online]. Available:http://qt-project.org/doc/qt-4.8/widgets-codeeditor.html
[10] (2014) Syntax Highlighter Example. [Online]. Available:http://qt-project.org/doc/qt-4.8/richtext-syntaxhighlighter.html
[11] (2014) Using Layouts in Qt Designer. [Online]. Available: http:// qt-project.org/doc/qt-4.8/designer-layouts.html
[12] (2014) QFrame Class Reference. [Online]. Available: http://qt-project.org/doc/qt-4.8/qframe.html#details
[13] (2014) QPlainTextEdit Class Reference. [Online]. Available: http://qt-project.org/doc/qt-4.8/qplaintextedit.html
[14] (2014) QLabel Class Reference. [Online]. Available: http://qt-project.org/doc/qt-4.8/qlabel.html#details
[15] (2014) Graphics View Framework. [Online]. Available: http://qt-project.org/doc/qt-4.8/graphicsview.html
[16] (2014) QScrollBar Class Reference. [Online]. Available: http://qt-project.org/doc/qt-5/QScrollBar.html#details
[17] R.B. Kline, and A. Seffah, "Evaluation of integrated software development environments: Challenges and results from three empirical studies," *International Journal of Human-Computer Studies*, pp. 607–627, July, 2005.
[18] Peng Gao, Lan-Fen Lin, Ming Cai, and Jin-Xiang Dong, "Study of Integrated Product Modeling in Cooperative Development Environment," *The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings*, 2003, pp. 250–254.
[19] Ying-Kui Gu, and Hong-Zhong Huang, "An Integrated Product and Process Development Model Supporting Entire Life Cycle," *The 8th International Conference on Computer Supported Cooperative Work in Design Proceedings*, 2003, pp. 558–562.