

<b>ID</b>	...	<b>Creation date</b>	June 18, 2009
<b>Platform</b>	S60 3rd Edition, FP1, FP2 S60 5th Edition	<b>Tested on devices</b>	
<b>Category</b>	Qt for Symbian	<b>Subcategory</b>	UI

**Keywords (APIs, classes, methods, functions):** QTextEdit, QSignalMapper

## Overview

This code example shows how to work with QTextEdit. QTextEdit class can be used as a replacement for Symbian class CEikRichTextEditor. The example duplicates all the functionalities from the Forum Nokia example [S60 Platform: Rich Text Editor Example](#).

This snippet can be self-signed. As it does not use any API which require developer/certified signing.

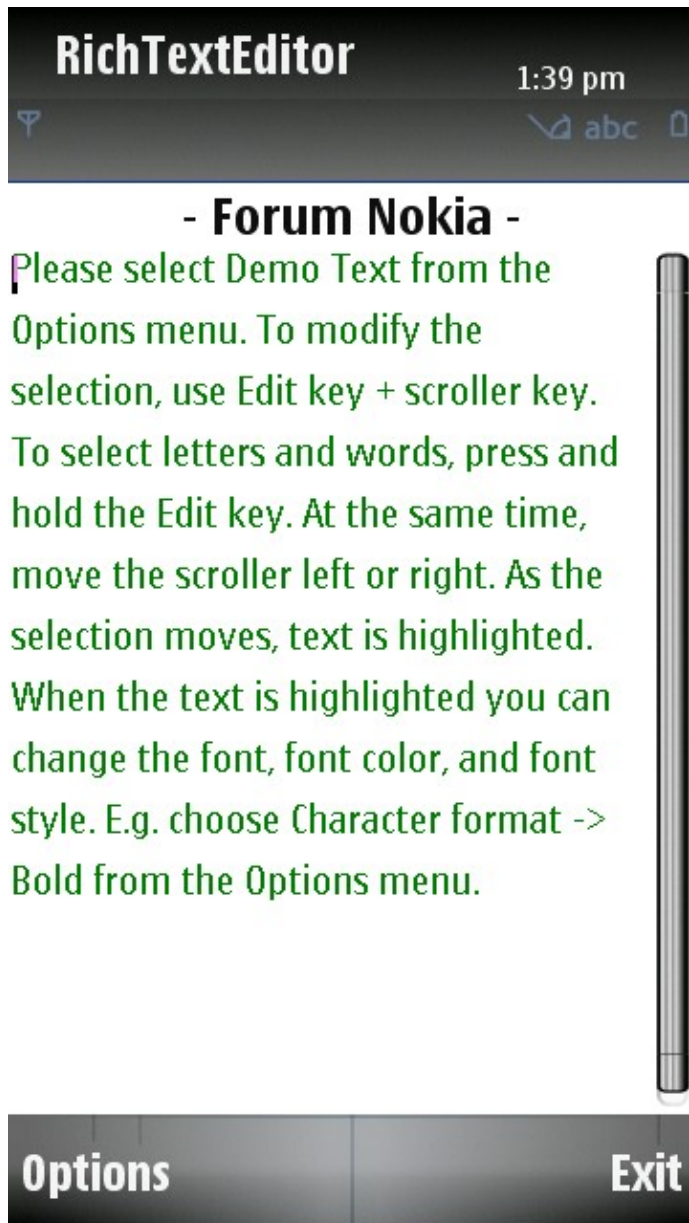
## Note

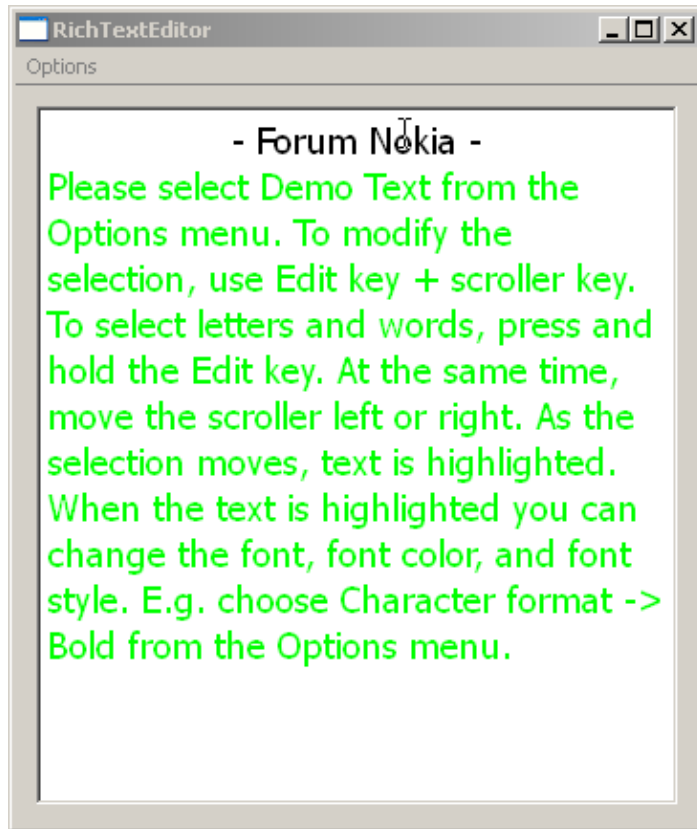
All of the QTextEdit functionality has not been ported to S60 yet.

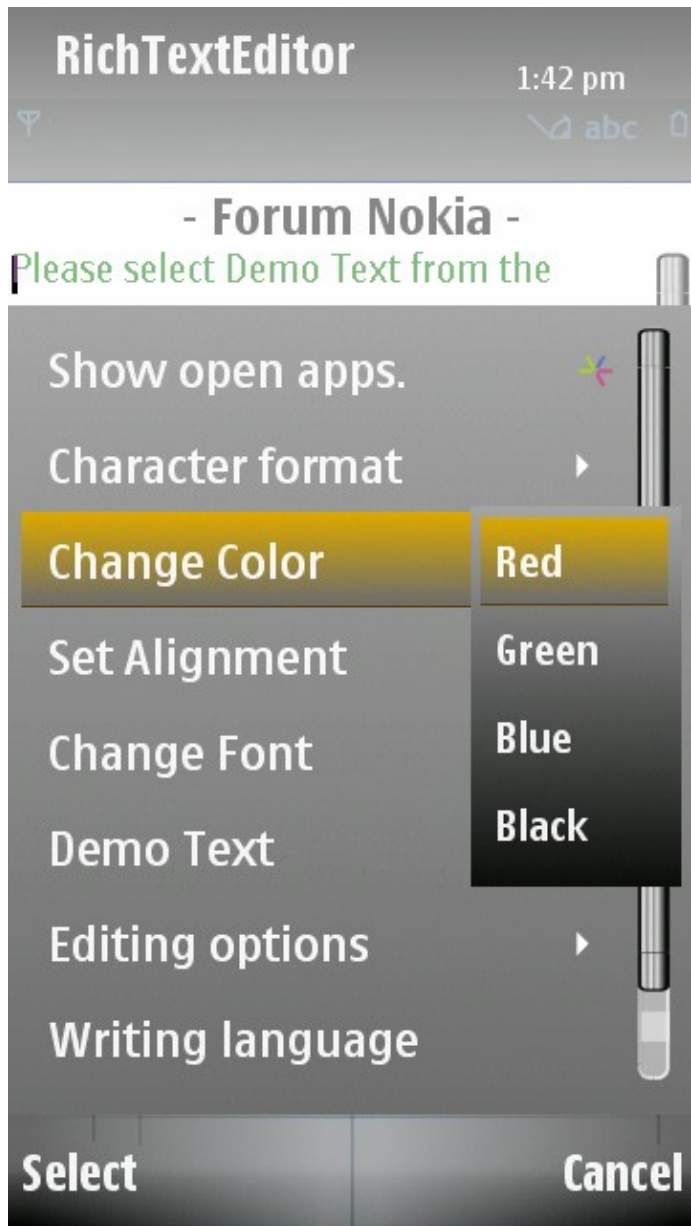
## Preconditions

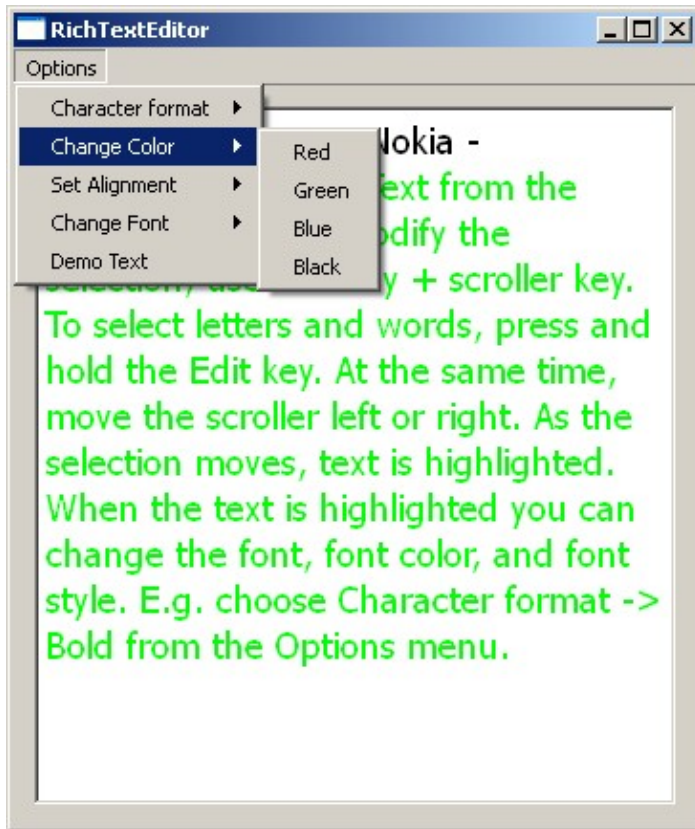
- Download and Install latest version [Qt for Symbian - Installation packages](#) which has links on how to install the latest version

## Images









## Source

### QtRichTextEditor.zip

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QtGui/QMainWindow>
#include <QSignalMapper>

namespace Ui
{
    class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = 0);
    ~MainWindow();

private slots:
    void setColor(int color);
    void setWeight(bool bold);
    void setStrike(bool strike);
    void setAlignment(int alignment);
    void setFontSize(int size);

```

## QtRichTextEditor

```
private:
    Ui::MainWindow *ui;
    QSignalMapper *colorMapper;
    QSignalMapper *alignmentMapper;
    QSignalMapper *fontMapper;

};

#endif // MAINWINDOW_H

#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent), ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    colorMapper = new QSignalMapper(this);
    alignmentMapper = new QSignalMapper(this);
    fontMapper = new QSignalMapper(this);

    connect(ui->actionBlack, SIGNAL(triggered()), colorMapper, SLOT(map()));
    connect(ui->actionBlue, SIGNAL(triggered()), colorMapper, SLOT(map()));
    connect(ui->actionGreen, SIGNAL(triggered()), colorMapper, SLOT(map()));
    connect(ui->actionRed, SIGNAL(triggered()), colorMapper, SLOT(map()));

    colorMapper->setMapping(ui->actionBlack, 0x00000000);
    colorMapper->setMapping(ui->actionBlue, 0x000000FF);
    colorMapper->setMapping(ui->actionGreen, 0x0000FF00);
    colorMapper->setMapping(ui->actionRed, 0x00FF0000);

    connect(ui->actionLeft, SIGNAL(triggered()), alignmentMapper, SLOT(map()));
    connect(ui->actionRight, SIGNAL(triggered()), alignmentMapper, SLOT(map()));
    connect(ui->actionCenter, SIGNAL(triggered()), alignmentMapper, SLOT(map()));
    connect(ui->actionJustify, SIGNAL(triggered()), alignmentMapper, SLOT(map()));

    alignmentMapper->setMapping(ui->actionLeft, Qt::AlignLeft);
    alignmentMapper->setMapping(ui->actionRight, Qt::AlignRight);
    alignmentMapper->setMapping(ui->actionCenter, Qt::AlignHCenter);
    alignmentMapper->setMapping(ui->actionJustify, Qt::AlignJustify);

    connect(ui->actionNormal, SIGNAL(triggered()), fontMapper, SLOT(map()));
    connect(ui->actionSmall, SIGNAL(triggered()), fontMapper, SLOT(map()));
    connect(ui->actionLarge, SIGNAL(triggered()), fontMapper, SLOT(map()));

    fontMapper->setMapping(ui->actionNormal, 10);
    fontMapper->setMapping(ui->actionSmall, 8);
    fontMapper->setMapping(ui->actionLarge, 14);

    connect(colorMapper, SIGNAL(mapped(int)), this, SLOT(setColor(int)));
    connect(alignmentMapper, SIGNAL(mapped(int)), this, SLOT(setAlignment(int)));
    connect(fontMapper, SIGNAL(mapped(int)), this, SLOT(setFontSize(int)));

    connect(ui->actionBold, SIGNAL(toggled(bool)), this, SLOT(setWeight(bool)));
    connect(ui->actionStrikethrough, SIGNAL(toggled(bool)), this, SLOT(setStrike(bool)));
}

MainWindow::~MainWindow()
{
    delete alignmentMapper;
    delete colorMapper;
}
```

## QtRichTextEditor

```
    delete fontMapper;
    delete ui;
}

void MainWindow::setColor(int color)
{
    QRgb c(color);
    ui->textEdit->setTextColor(c);
}

void MainWindow::setWeight(bool bold)
{
    ui->textEdit->setFontWeight(bold ? QFont::Bold : QFont::Normal);
}

void MainWindow::setStrike(bool strike)
{
    QFont font = ui->textEdit->currentFont();
    font.setStrikeOut(strike);
    ui->textEdit->setFont(font);
}

void MainWindow::setAlignment(int alignment)
{
    Qt::Alignment a(alignment);
    ui->textEdit->setAlignment(a);
}

void MainWindow::setFontSize(int size)
{
    qreal s(size);
    ui->textEdit->setFontPointSize(s);
}
```

## External Links

- QTextEdit: [QTextEdit reference](#)
- QSignalMapper: [QSignalMapper reference](#)