

QtRotatelImage

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

Keywords (APIs, classes, methods, functions): QPixmap, QSignalMapper, QFileDialog::getOpenFileName, QFileDialog::getSaveFileName

Overview

This code example shows how to work with QPixmap class. User can load any image, rotate it and then save it with a new name.

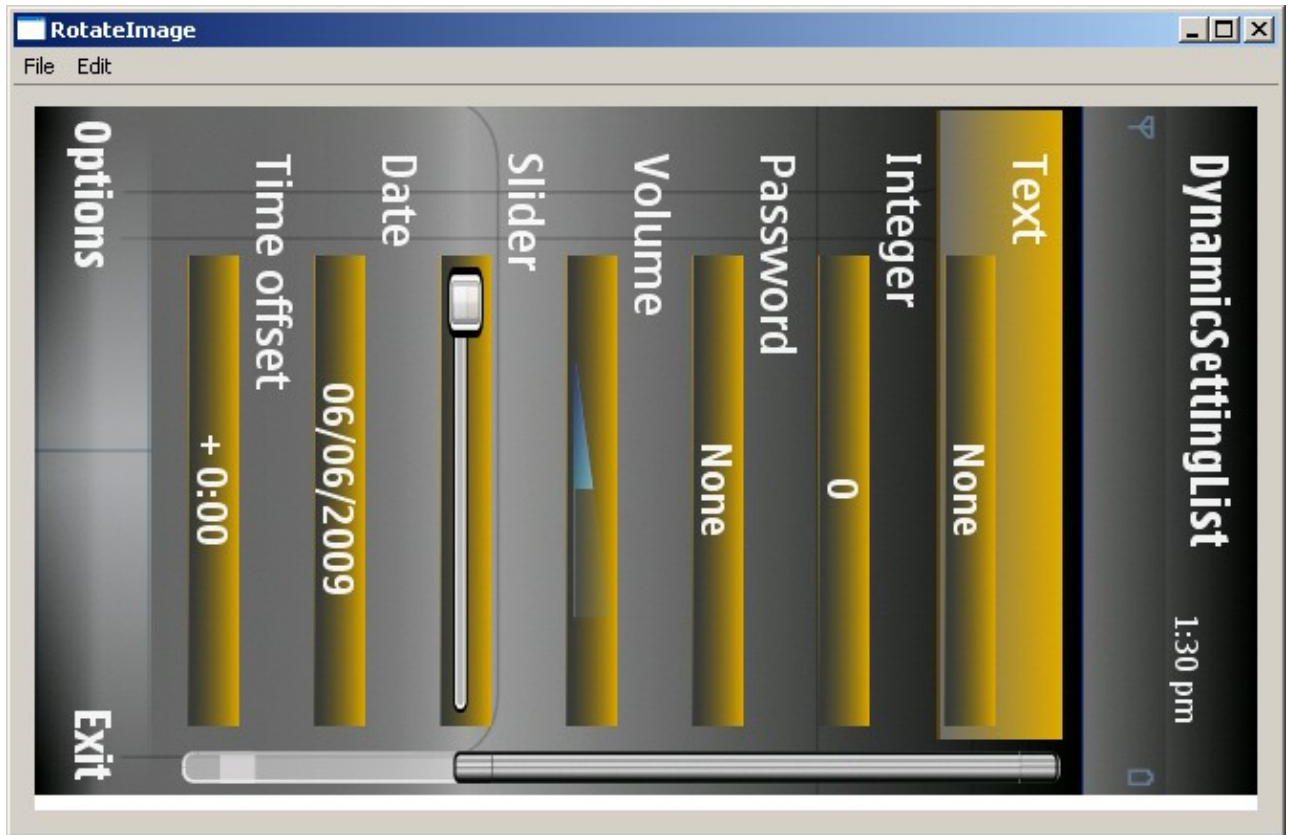
This snippet can be self-signed. As it does not use any API which require developer/certified signing.

Preconditions

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

Images





Source

[QtRotateImage.zip](#)

```
#include <QtGui/QApplication>
#include "mainwindow.h"

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

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

namespace Ui
{
    class MainWindow;
}

class MainWindow : public QMainWindow
{
```

QtRotateImage

```
Q_OBJECT

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

private slots:
    void loadImage();
    void saveImage();
    void rotateImage(int angle);

private:
    Ui::MainWindow *ui;
    QSignalMapper *rotateMapper;
    QPixmap pixmap;
};

#endif // MAINWINDOW_H

#include <QFileDialog>
#include <QPixmap>

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

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

    rotateMapper = new QSignalMapper(this);

    connect(ui->actionLoad, SIGNAL(triggered()), this, SLOT(loadImage()));
    connect(ui->actionSave, SIGNAL(triggered()), this, SLOT(saveImage()));

    connect(ui->actionRotate_90, SIGNAL(triggered()), rotateMapper, SLOT(map()));
    connect(ui->actionRotate_180, SIGNAL(triggered()), rotateMapper, SLOT(map()));
    connect(ui->actionRotate_270, SIGNAL(triggered()), rotateMapper, SLOT(map()));

    rotateMapper->setMapping(ui->actionRotate_90, 90);
    rotateMapper->setMapping(ui->actionRotate_180, 180);
    rotateMapper->setMapping(ui->actionRotate_270, 270);

    connect(rotateMapper, SIGNAL(mapped(int)), this, SLOT(rotateImage(int)));
}

MainWindow::~~MainWindow()
{
    if(rotateMapper)
    {
        delete rotateMapper;
    }
    delete ui;
}

void MainWindow::loadImage()
{
    QString fileName = QFileDialog::getOpenFileName(this,
        tr("Open Image"), "/", tr("Image Files (*.png *.jpg *.bmp)"));
    if ( fileName.isNull() == false )
    {

```

QtRotateImage

```
        if ( pixmap.load(fileName) == true )
        {
            ui->label->setPixmap(pixmap);
        }
    }

void MainWindow::saveImage()
{
    if ( pixmap.isNull() == false )
    {
        QString fileName = QFileDialog::getSaveFileName(this,
            tr("Save Image"), "/", tr("Image Files (*.png *.jpg *.bmp)"));
        if ( fileName.isNull() == false )
        {
            pixmap.save(fileName);
        }
    }
}

void MainWindow::rotateImage(int angle)
{
    int h = pixmap.height();
    int w = pixmap.width();

    QPixmap p = pixmap.transformed(
        QTransform().translate(w/2, h/2).rotate(angle).translate(-w/2, -h/2));
    ui->label->setPixmap(p);
    pixmap=p;
}
```

External Links

- QPixmap: [QPixmap reference](#)
- QSignalMapper: [QSignalMapper reference](#)
- QFileDialog: [QFileDialog reference](#)