Archived:Applying a Qt style sheet to an application via QApplication

Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (using C++ for the Qt app UI) is deprecated.

Overview

This example shows you how to apply a style sheet application-wide.

Preconditions

- You know how to Using resources in Qt.

Applying a Qt style sheet to an application via QApplication

```
#include "yourwindow.h"

#include <QtGui>
#include <QApplication>

#include <QFile>
#include <QString>
#include <QTextStream>

void loadStyleSheet() {
    /* Let's use QFile and point to a resource... */
    QFile data(":/style.qss");
    QString style;
    /* ...to open the file */
    if(data.open(QFile::ReadOnly)) {
        /* QTextStream... */
        QTextStream styleIn(&data);
        /* ...read file to a string. */
        style = styleIn.readAll();
        data.close();
        /* We'll use qApp macro to get the QApplication pointer
         * and set the style sheet application wide. */
        qApp->setStyleSheet(style);
    }
}

int main(int argc, char *argv[]) {
    QApplication a(argc, argv);
    YourWindow* w = new YourWindow();
    /* Load style sheet */
    loadStyleSheet();
    w->showMaximized();
    int returnValue = a.exec();
    w->deleteLater();
    return returnValue;
```


(C) Copyright Nokia 2014. All rights reserved.

```
}
```

Postconditions

Now you can style your application with Qt style sheets.

See also

- Using resources in Qt
- QFile documentation
- QTextStream documentation
- QApplication documentation
- qApp macro
- Qt style sheet documentation
- Archived:Creating a gradient background for a QPushButton with style sheet

(C) Copyright Nokia 2014. All rights reserved.