Using Qt environment macros to differentiate platforms

This article needs to be updated: If you found this article useful, please fix the problems below then delete the {{ArticleNeedsUpdate}} template from the article to remove this warning.

Reasons: hamishwillee (01 Sep 2011)
Article is accurate but should be extended to include newer Qt Environment macros (including Q_WS_MAEMO_6, Q_WS_MEEGO etc.)

Overview

This code snippet demonstrates how to write platform-specific code using the `Q_OS_SYMBIAN`, `Q_WS_MAEMO_5`, and `Q_WS_SIMULATOR` environment macros. You can use the macros with Qt code if you need to, for example, define a different screen size for each platform, or use each platform's native language (for example, Symbian C++ in Symbian and C in Maemo), and still be able to build the project for all targets. Note that in the latter case you need to declare in the project file which platform-specific libraries you use.

In the project file, platform-specific code must be placed in scopes. For Symbian, use `symbian`; for Maemo, use `maemo5`; and for Simulator, use `simulator`. Note that scope `win32` is valid for both Simulator and Windows (desktop target) so when writing code only for Windows use scope `win32: !simulator`.

Preconditions

None.

Source

platformlabel.pro

```
QT      += core gui

TARGET   = platformlabel
TEMPLATE = app

SOURCES += main.cpp

symbian {
    # To lock the application orientation
    LIBS += -lcone -leikcore -lavkon
}
```

main.cpp

```
#include <QApplication>
#include <QLabel>

#ifdef Q_OS_SYMBIAN
// Need these includes to lock orientation in Symbian
#include <eikenv.h>
#include <eikappui.h>
#include <aknenv.h>
#include <aknappui.h>
#endif

int main(int argc, char *argv[])
```


(C) Copyright Nokia 2014. All rights reserved.

```
{
    QApplication app(argc, argv);
    QLabel label;

#ifdef Q_OS_SYMBIAN
    // Lock orientation to portrait in Symbian
    CAknAppUi* appUi = dynamic_cast<CAknAppUi*> (CEikonEnv::Static()->AppUi());
    TRAP_IGNORE(
        if(appUi) {
            appUi->SetOrientationL(CAknAppUi::EAppUiOrientationPortrait);
        }
    );
#endif

#ifdef Q_WS_MAEMO_5
    // Lock orientation to portrait in Maemo
    label.setAttribute(Qt::WA_Maemo5PortraitOrientation, true);
#endif

#if defined(Q_OS_SYMBIAN)
    // Symbian^3 specific code can be defined using the SV_S60_5_2 macro
    if (QSysInfo::s60Version() == QSysInfo::SV_S60_5_2) {
        label.setText("Symbian^3");
    }
    else {
        label.setText("Symbian");
    }
#elif defined(Q_WS_MAEMO_5)
    label.setText("Maemo");
#elif defined(Q_WS_SIMULATOR)
    label.setText("Simulator");
#else
    label.setText("Some other platform");
#endif

#if defined(Q_OS_SYMBIAN)
    label.showMaximized();
#else
    // On Maemo and Simulator QMainWindow (to which the only QLabel will map behind the
scene)
    // is automatically shown as maximised and we don't need to ifdef platform specific
code
    // for it.
    label.show();
#endif

    return app.exec();
}
```

Postconditions

The code snippet demonstrated the use of environment macros to write platform-specific code.

(C) Copyright Nokia 2014. All rights reserved.

(C) Copyright Nokia 2014. All rights reserved.