Launching a native application from Qt

This code snippet shows how to launch a native application from Qt. The example has been tested launching the Symbian camera application on the Symbian platform, but the code should work generically.

Header

The `ProcessHandler` starts the process and listens for its states.

```
#include <QObject>
#include <QProcess>

class ProcessHandler : public QObject
    {
    Q_OBJECT

    public:
        ProcessHandler(QObject *parent = 0);
        ~ProcessHandler();

    public:
        void StartCameraApp();

    public Q_SLOTS:
        void stateChanged(QProcess::ProcessState state);
        void error(QProcess::ProcessError error);

    private:
        QProcess*   process;
    };
```

Source

Create `QProcess` and start listening for process state changes.

```
ProcessHandler::ProcessHandler(QObject *parent)
: QObject(parent)
    {
    process = new QProcess(this);
    QObject::connect(process, SIGNAL(stateChanged(QProcess::ProcessState)),
                     this, SLOT(stateChanged(QProcess::ProcessState)));
    }
```

Class descruction:

```
ProcessHandler::~ProcessHandler()
    {
    if (process->state() != QProcess::NotRunning)
        {
        // Close process if it is running
        process->close();
        }
    }
```

Start the Symbian native camera `Cameraapp.exe`:

```
void ProcessHandler::StartCameraApp()
    {
    QString program = "Cameraapp.exe";
    process->start(program);
    }
```

`QProcess` signals the error to the error slot:

```
void ProcessHandler::error(QProcess::ProcessError error)
    {
    switch (error)
        {
        case QProcess::FailedToStart:
        case QProcess::Crashed:
        case QProcess::Timedout:
        case QProcess::ReadError:
        case QProcess::WriteError:
        case QProcess::UnknownError:
            {
            // TODO: Handle error
            break;
            }
        default:
            {
            break;
            }
        };
    }
```


(C) Copyright Nokia 2013. All rights reserved.

QProcess signals the process state changes to the stateChanged slot.

```
void ProcessHandler::stateChanged(QProcess::ProcessState state)
    {
    switch (state)
        {
        case QProcess::NotRunning:
        case QProcess::Starting:
        case QProcess::Running:
            {
            // TODO: do what you want with
            // different states
            break;
            }
        default:
            {
            break;
            }
        };
    }
```

Postconditions

The Symbian native camera is started.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.