Connecting Qt signal to QML function

This snippet shows how to connect signals and slots between Qt and QML code. Signals and slots are one way to provide data for the Qt Quick UI and make the UI react to changes in the model.

Preconditions

- Qt 4.7 or higher is installed on your platform.

Qt Project File

```
#To make sure we use declarative
QT += declarative

#To get files deployed on device / emulator
files.sources += ui.qml
DEPLOYMENT += files
```

Source

NOTE: The parameter type in signal and slot is `QVariant` on the C++ side, as the functions in QML are JavaScript and they have no type.

myclass.h

```
#ifndef MYCLASS_H
#define MYCLASS_H

#include <QObject>
#include <QVariant>

class MyClass : public QObject
{
    Q_OBJECT

public:
    MyClass() {}

public slots:
    void getData() {
        QString text("New data");
        emit data(QVariant(text));
    }

signals:
    void data(QVariant data);
};

#endif // MYCLASS_H
```

main.cpp

```
#include <QApplication>
#include <QGraphicsObject>
#include <QDeclarativeView>
#include <myclass.h>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    MyClass myClass;

    QDeclarativeView view;
    view.setSource(QUrl("./ui.qml"));
    view.setResizeMode(QDeclarativeView::SizeRootObjectToView);

    QObject *rootObject = dynamic_cast<QObject*>(view.rootObject());

    QObject::connect(rootObject, SIGNAL(dataRequired()), &myClass, SLOT(getData()));
    QObject::connect(&myClass, SIGNAL(data(QVariant)), rootObject, SLOT(updateData(QVariant)));

#if defined(Q_WS_MAEMO_5)
    view.setGeometry(QRect(0,0,800,480));
    view.showFullScreen();
#elif defined(Q_WS_S60)
    view.setGeometry(QRect(0,0,640,360));
    view.showFullScreen();
#else
    view.setGeometry(QRect(100,100,800, 480));
    view.show();
#endif

    return app.exec();
}
```

ui.qml

```
// ui.qml
import Qt 4.7
```


(C) Copyright Nokia 2013. All rights reserved.

```
Rectangle {
    signal dataRequired;

    function updateData(text) { dataText.text = text }  // slot

    anchors.fill: parent; color: "black"

    Text {
        id: dataText
        anchors.centerIn: parent; color: "white"
    }

    MouseArea {
        anchors.fill: parent
        onClicked: dataRequired()
    }
}
```

Postconditions

The signals and slots in both Qt and QML are connected to each other: when `rootObject`'s `dataRequired` is signalled, `myClass`'s `getData` gets called. Also, when `myClass` data is signalled, `rootObject`'s `updateData` is called.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.