QML SMShelper Plugin

This article creates a C++ extension to QML (`SMSplugin`) to expose Qt Messaging APIs to QML environment. Note it is also possible to launch the standard SMS editor from QML using Qt.openUrlExternally, as described in How to use Qt.openUrlExternally in QML.

Pro file

The following capabilities and libraries are required:

```
symbian:TARGET.CAPABILITY = NetworkServices
CONFIG += mobility
MOBILITY = messaging
```

Header file

smshelper.h applying this macro to definitions of member functions to allow them to be invoked via the **meta-object system**

```
#ifndef SMSHELPER_H
#define SMSHELPER_H

#include <QObject>
#include <qmessage.h>
#include <qmessageservice.h>

QTM_USE_NAMESPACE


#ifdef Q_OS_SYMBIAN
#  define SENDSMS_ENABLED
#elif defined(Q_WS_MAEMO_5) || defined(Q_WS_MAEMO_6)
#  define SENDSMS_ENABLED
#endif

class SMSHelper : public QObject
{
    Q_OBJECT

public:
    explicit SMSHelper(QObject *parent = 0);
    ~SMSHelper();

    Q_INVOKABLE bool sendsms(QString phonenumber, QString message);

signals:
    void stateMsg(const QString &statemsg);
    void errorMsg(const QString &errormsg);
    void debugMsg(const QString &dbgmsg);

private slots:
    void messageStateChanged(QMessageService::State s);
#ifndef SENDSMS_ENABLED
    void signalFinishedState() { emit stateMsg("FinishedState");};
#endif
private:
    QMessageService iMessageService;
    QMessageManager iManager;
    QMessageService::State state;
};

#endif // SMSHELPER_H
```

Source file

smshelper source code

```
#include "smshelper.h"

#ifndef SENDSMS_ENABLED
#include <QTimer>
#endif

SMSHelper::SMSHelper(QObject *parent) :
    QObject(parent)
{
    state = QMessageService::InactiveState;
    connect(&iMessageService, SIGNAL(stateChanged(QMessageService::State)), this, SLOT(messageStateChanged(QMessageService::State)))
}

SMSHelper::~SMSHelper()
{
}


bool SMSHelper::sendsms(QString phonenumber, QString message)
{

#ifdef SENDSMS_ENABLED
    if (!QMessageAccount::defaultAccount(QMessage::Sms).isValid())
    {
        emit errorMsg("No messageaccount for sms sending.");
```


(C) Copyright Nokia 2013. All rights reserved.

```
        return false;
    }

    if (state == QMessageService::InactiveState || state == QMessageService::FinishedState)
    {
        QMessage sms;

        sms.setType(QMessage::Sms);
        sms.setParentAccountId(QMessageAccount::defaultAccount(QMessage::Sms));
        sms.setTo(QMessageAddress(QMessageAddress::Phone, phonenumber));
        sms.setBody(message);

        return iMessageService.send(sms);
    }
    else
    {
        return false;
    }

#else
    QTimer::singleShot(1000,this,SLOT(signalFinishedState()));
    return true;
#endif
}

void SMSHelper::messageStateChanged(QMessageService::State s)
{
    state = s;

    if (s == QMessageService::InactiveState)
    {
        emit stateMsg("InactiveState");
    }
    else if (s == QMessageService::ActiveState)
    {
        emit stateMsg("ActiveState");
    }
    else if (s == QMessageService::CanceledState)
    {
        emit stateMsg("CanceledState");
    }
    else if (s == QMessageService::FinishedState)
    {
        emit stateMsg("FinishedState");
    }
    else
    {
         emit stateMsg(QString("QMessageService::%1").arg(s));
    }
}
```

To register the C++ type in the QML system with the name SMSHelper include following code in your main.cpp file

```
qmlRegisterType<SMSHelper>("SMSHelper",1,0,"SMSHelper");
```

Import sms plugin in QML

```
import SMSHelper 1.0
```

Example use QML

```
SMSHelper
    {
        id: smshelper
}
```

SMShelper developer APIs

```
//Constructor and destructor
SMSHelper(QObject *parent = 0);
~SMSHelper();

public slots:
sendsms(QString phonenumber, QString message);

signals:
void stateMsg(const QString &statemsg);
void errorMsg(const QString &errormsg);
void debugMsg(const QString &dbgmsg);
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.