Live TV with Qt

Description

This is a Live Streaming Application which launches the default Real Player of the device and plays an RTSP Streaming Link. I will show you how to play **You Tube Videos** (Live) in this article. You can also try to play Live TV Channels using other RTSP like **rtsp://server/domain/tvname.sdp**

How to start

1. Lets create a text file and paste this link **rtsp://v4.cache4.c.youtube.com/CkELENy73wIaOAnGbz7NeJeeixMYDSANFEIJbXYtZ29vZ2xlSARSBXdhdGNoWgIDbGlja0xpbmtgqZW17Zb**
2. Save the text file as **youtube.ram**
3. Create a folder with a name **channel**
4. Copy the file **youtube.ram** to **channel**
5. Create a New Qt project, and named as **LiveTVQt**
6. Copy the **channel** folder to **LiveTVQt** (project directory).

LiveTVQt.pro

Use the DEPLOYMENT keyword to copy files into the package file and to the device. All the files from the **channel** folder are copied into the package. When the package is installed on the device, the files are stored on the device's **C:\Data\System\LiveTVChannel** folder.

```
TEMPLATE = app
TARGET = LiveTVQt
QT += core \
    gui \
    network \
    webkit \
    webkit \
    phonon
HEADERS += LiveTVQt.loc \
    LiveTVQt.h
SOURCES += LiveTVQt.rss \
    LiveTVQt_reg.rss \
    main.cpp \
    LiveTVQt.cpp
FORMS += LiveTVQt.ui
RESOURCES +=
symbian:TARGET.UID3 = 0xEA524E4F

myFiles.sources = channel\*
myFiles.path = c:\data\system\LiveTVChannel
DEPLOYMENT += myFiles
```

LiveTVQt.h

```
#ifndef LIVETVQT_H
#define LIVETVQT_H

#include <QtGui/QMainWindow>
#include "ui_LiveTVQt.h"
#include <QProcess>

class LiveTVQt : public QMainWindow
{
    Q_OBJECT
```


(C) Copyright Nokia 2013. All rights reserved.

```
public:
 LiveTVQt(QWidget *parent = 0);
    ~LiveTVQt();
    //function to create menu.
       void createMyMenu();
public Q_SLOTS: //slots to receive action of menu trigger.

    void launchRealPlayerAction();
    void processError(QProcess::ProcessError err);


private:
    Ui::LiveTVQt ui;
    //Actions for menu item
    QAction* menu_launchRealPlayeAction;
   QAction* menu_exitAction;
   QProcess process;
};

#endif // LIVETVQT_H
```

LiveTVQt.cpp

```
#include "LiveTVQt.h"
#include <QDesktopServices>
#include <QUrl>
#include <QMessageBox>

LiveTVQt::LiveTVQt(QWidget *parent)
    : QMainWindow(parent)
{
 ui.setupUi(this);
 setWindowTitle("Live TV");
  createMyMenu();
     QObject::connect( &process, SIGNAL(error(QProcess::ProcessError)),
                       this, SLOT(processError(QProcess::ProcessError)));
}

LiveTVQt::~LiveTVQt()
{

}

void LiveTVQt::createMyMenu()
{


 menu_launchRealPlayeAction = new QAction(tr("You Tube"), this);
 menuBar()->addAction(menu_launchRealPlayeAction);
 connect(menu_launchRealPlayeAction, SIGNAL(triggered()), this, SLOT(launchRealPlayerAction()));



 menu_exitAction = new QAction(tr("Exit"), this);
 menuBar()->addAction(menu_exitAction);
 connect(menu_exitAction, SIGNAL(triggered()), this, SLOT(close()));
}


void LiveTVQt::launchRealPlayerAction()
{
 QDesktopServices::openUrl(QUrl(tr("file:///C:/data/system/LiveTVChannel/youtube.ram")));

}


void LiveTVQt::processError(QProcess::ProcessError err)
{
 switch(err)
 {
 case QProcess::FailedToStart:
  QMessageBox::information(0,"FailedToStart","FailedToStart");
  break;
 case QProcess::Crashed:
  QMessageBox::information(0,"Crashed","Crashed");
  break;
 case QProcess::Timedout:
  QMessageBox::information(0,"FailedToStart","FailedToStart");
  break;
 case QProcess::WriteError:
  QMessageBox::information(0,"Timedout","Timedout");
  break;
 case QProcess::ReadError:
  QMessageBox::information(0,"ReadError","ReadError");
  break;
 case QProcess::UnknownError:
  QMessageBox::information(0,"UnknownError","UnknownError");
  break;
 default:
  QMessageBox::information(0,"default","default");
  break;
 }
}
```

Source Code

The full source code presented in this article is available here File:LiveTVQt.zip

Related Example

BTV Live Source Code: BTV Live

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.