[西祠胡同社区 - 城市生活社区门户](http://www.xici.net/main.asp?url=/u16539771/d104902644.htm)

如何使用 Qt Phonon

09-11-28 12:55  发表于：[《南京嵌入式开发圈》](http://www.xici.net/b1123098/board.asp)　分类：未分类

前言

Phonon严格来说其实非为Qt的library，Phonon原本就是KDE 4的开放原始码多媒体API，後来与Qt合并与开发，所以简单来说就是Qt使用Phonon这个多媒体框架来提供一般影音多媒体档案的播放，而这些影音多媒体来源可以是档案、网路串流或是指到一个档案的QUrl。

Phonon是一个跨平台多媒体框架，能够在Qt应用程式中使用与播放影音多媒体内容。

## [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=2)] Phonon的架构

整体来说，Phonon的架构只需要记住以下的三东西：

* **media object**

Phonon的基础，用於管理多媒体来源。来源可能是影音档等，而能够提供基本的播放控制，例如开始、暂停或结束。而提供多媒体资料给media object的则为media source，在给media object之前通常是raw data，再由media object进行转换。

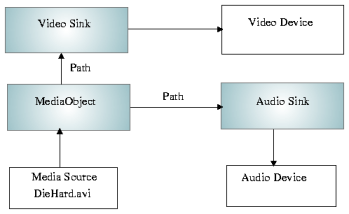
* **sinks**

输出多媒体，例如在widget上播放影片或是输出至音效卡(播放音乐)。通常sink是一个播放的装置(例如音效卡等)。而sink只接受media object来的资料，由media object控制播放；而由sink来处理这些多媒体

* **paths**

用来连接Phonon的物件，意即media object与sink之间的连接。

可从下图看出彼此的关系(资料来源：[Phonon Overview](http://doc.trolltech.com/4.5/phonon-overview.html))：



所以整个播放影音的流程就是首先由media object开始播放，接著把媒体串流经由path送至sink，sink会经由音效卡等装置重新播放(play back)影音。

## [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=3)] 安装

QtSoftware官方网站是说Phonon预设会跟在安装Qt时一并安装，不过我不论是从Qt SDK或从source code重新build，都没有包含Phonon模组，目前我是用另外一种方式来安装Phonon，就是使用套件管理工具来从套件包来安装，只需要安装下列的套件：

sudo apt-get install libphonon-dev libphonon4 phononbackend-gstreamer

安装完毕之後，就可以使用Phono模组罗。

而与其他Qt应用程式一样，若有使用到Phonon函式库的应用程式在build的时候都需要额外设定使用Phonon模组，必须要在qmaek project file中加入：

QT += phonon

## [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=4)] 使用

在此部分将会开始介绍如何使用Phonon这套函式库，并且会列出一些小范例来demo，首先就先来介绍几个简单好用的类别。

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=5)] VideoPlayer Class

Phonon有提供很多类别可供使用，其中最简单的，莫过於VideoPlayer这个类别了。

VideoPlayer widget如它的名字一样，就是用来播放video，而且使用起来相当简易，而且功能也不会缺少，包括播放、暂停与停止。

而一开始早先提到的MediaObject等类别与VideoPlayer在使用上有什么差异呢？如果不需要更复杂的功能，例如建立一个media graph，你只需要能够播放影音档案的话，其实使用VideoPlayer类别即可达到你的要求。

而另外值得一提的就是，VideoPlayer大部分函式都是非同步，所以载入media source并不会马上播放多媒体档案，只有在呼叫函式play( )之後才会播放。

要怎么使用这个类别呢？ 其实相当简单，下面就是程式码片段：

VideoPlayer \*player = new VideoPlayer(Phonon::VideoCategory, parentWidget);

player->play(url);

在实体化VideoPlayer类别物件时，可以在constructor就载入多媒体种类与要放在哪个widget中(即为 parentWidget)，而media source可以利用函式load()来载入或是在play()时载入，而载入方式可直接从档案或是从网路位址。

以下就是一个简单的影音播放功能小程式：

#include <QApplication>

#include <QWidget>

#include <phonon>

#include <QUrl>

int main(int argc, char \*argv[]) {

QApplication app(argc, argv);

QWidget \*widget = new QWidget;

widget->setWindowTitle("Video Player");

widget->resize(400,400);

Phonon::VideoPlayer \*player = new Phonon::VideoPlayer(Phonon::VideoCategory, widget);

player->load(Phonon::MediaSource("../Puppet.mpg"));

player->play();

widget->show();

return app.exec();

}

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=6)] MediaObject Class

MediaObject类别主要提供一个能够处理媒体播放的介面。

MediaObject可说是处理多媒体档案最基本的一部份，它接受并管理来自於MediaSource的媒体档案。而媒体播放、暂停与停止都是由它来控制；而在此之前，media物件必须要与output node连接，如早先所讲的，这个nodes主要将媒体输出至底层的硬体，例如音效卡或显示卡等，而所需要的output node则是根据多媒体的内容而所不同，目前Phonon有两种output node；

* AudioOutput－声音播放
* VideoWidget－影像播放

如果MediaSource包含声音与影像的话，这两种node都必须要连接至media物件。

就这个类别来说，有几个函式是你必须要知道的，分别是

* setCurrentSource()：设定MediaObject的多媒体来源，而来源可以是网路上的影音档(利用QUrl来存取)或是本机档案(利用QString)，使用上相当简单：

QUrl url("http://www.example.com/music.ogg");

media->setCurrentSource(url);

* play()：开始播放多媒体资料
* pause()：暂停播放
* stop()：停止播放

以下为一个简单的程式片段，说明如何使用：

Phonon::MediaObject \*mediaObject = new Phonon::MediaObject(this);

Phonon::VideoWidget \*videoWidget = new Phonon::VideoWidget(this);

Phonon::createPath(mediaObject, videoWidget);

Phonon::AudioOutput \*audioOutput =

new Phonon::AudioOutput(Phonon::VideoCategory, this);

Phonon::createPath(mediaObject, audioOutput);

mediaObject->play();

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=7)] Phonon::createPath()

这是相当重要的一个函式，主要用於建立一个Path，连接两个MediaNodes，就是source与sink。

其实它的主要用途是在更进阶的部分，就是在使用到media graph，不过如果只是利用Phonon来播放多媒体影音档，其实只要记得它是用来连接source与输出装置即可。

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=8)] AudioOutput Class

AudioOutput类别主要是用来把多媒体的声音送到声音输出装置。所以它能够经由类似喇叭等输出装置来播放声音，稍早有提过，多媒体资料的来源必须要经过Phonon::createPath()由MediaObject连接。

Phonon::MediaObject \*mediaObject = new Phonon::MediaObject(this);

mediaObject->setCurrentSource(Phonon::MediaSource("/mymusic/barbiegirl.wav"));

Phonon::AudioOutput \*audioOutput =

new Phonon::AudioOutput(Phonon::MusicCategory, this);

Phonon::Path path = Phonon::createPath(mediaObject, audioOutput);

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=9)] VideoWidget Class

VideoWidget类别提供能够显示出影片的widget。

VideoWidget类别会在QWidget上播放多媒体串流的影像，跟AudioOutput一样，必须使用 Phonon::createPath()来与MediaObject连接。你可以利用一些函式来控制在QWidget中的VideoWidget显示的大小，你可以利用setAspectRatio()或setScaleMode()来控制，而它们接收的参数可以到网站上察看，使用方式如下(预设是使用 aspect ratio)：

videowidget->setAspectRatio(Phonon::VideoWidget::AspectRatioAuto);

videowidget->setScaleMode(Phonon::VideoWidget::ScaleAndCrop);

当然也有提供函式让影片进入或退出全萤幕模式。以下为一个简短的程式码范例：

MediaObject \*media = new MediaObject(parent);

VideoWidget \*vwidget = new VideoWidget(parent);

Phonon::createPath(media, vwidget);

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=10)] SeekSlider Class

SeekSlider类别提供一个可滑动的slider来设定多媒体串流播放的位置。所以它会连接到MediaObject，并控制串流目前的位置。

以下是一个使用的范例：

Phonon::MediaObject \*moo = new Phonon::MediaObject;;

Phonon::AudioOutput \*device = new Phonon::AudioOutput;

Phonon::createPath(moo, device);

moo->setCurrentSource(

QString("/home/gvatteka/Music/Lumme-Badloop.ogg"));

Phonon::SeekSlider \*slider = new Phonon::SeekSlider;

slider->setMediaObject(moo);

slider->show();

moo->play();

### [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=11)] VolumeSlider Class

VolumeSlider widget提供可以控制声音装置音量的widget。

用法其实与上面的SeekSlider类似，使用范例如下：

honon::AudioOutput \*audioOutput = new Phonon::AudioOutput(Phonon::MusicCategory);

Phonon::createPath(mediaObject, audioOutput);

Phonon::VolumeSlider \*volumeSlider = new Phonon::VolumeSlider;

volumeSlider->setAudioOutput(audioOutput);

## [[编辑](http://support.oss.org.tw/wiki/index.php?title=Qt_Phonon&action=edit&section=12)] 范例

再看过上面的一些类别介绍之後，其实就可以显一个简单的媒体播放器了，请看范例程式码：

#include <QApplication>

#include <QWidget>

#include <phonon>

#include <QUrl>

#include <QObject>

#include <QVBoxLayout>

#include <QHBoxLayout>

#include <QLabel>

int main(int argc, char \*argv[]) {

QApplication app(argc, argv);

QWidget \*widget = new QWidget;

widget->setWindowTitle("Media Player");

widget->resize(400,400);

Phonon::MediaObject \*media = new Phonon::MediaObject;

media->setCurrentSource(Phonon::MediaSource("../Puppet.mpg"));

Phonon::VideoWidget \*vwidget = new Phonon::VideoWidget(widget);

Phonon::createPath(media, vwidget);

vwidget->setAspectRatio(Phonon::VideoWidget::AspectRatioAuto);

Phonon::AudioOutput \*aOutput = new Phonon::AudioOutput(Phonon::VideoCategory);

Phonon::createPath(media, aOutput);

QLabel \*label = new QLabel("Volume: ");

Phonon::VolumeSlider \*volumeSlider = new Phonon::VolumeSlider;

volumeSlider->setAudioOutput(aOutput);

volumeSlider->setSizePolicy(QSizePolicy::Maximum, QSizePolicy::Maximum);

Phonon::SeekSlider \*seekSlider = new Phonon::SeekSlider;

seekSlider->setMediaObject(media);

QHBoxLayout \*hLayout = new QHBoxLayout;

hLayout->addWidget(label);

hLayout->addWidget(volumeSlider);

hLayout->addStretch();

QVBoxLayout \*vLayout = new QVBoxLayout;

vLayout->addWidget(vwidget);

vLayout->addWidget(seekSlider);

vLayout->addLayout(hLayout);

widget->setLayout(vLayout);

widget->show();

media->play();

return app.exec();

}

上面的程式码就完成了一个简易的多媒体播放器，当然还有很多部分还需要改进，不过在此仅就Phonon的使用上做介绍。执行之後的画面如下所示：

由於VideoWidget已经嵌入到QWidget了，所以你调整视窗大小的话，影片播放的视窗也会跟著调整。