

113 多媒体模块播放音乐

113.1 Qt的多媒体模块

Qt的多媒体模块为录音、录像、拍照、音视频播放，乃至数字收音机等功能提供支持。要在应用程序中使用Qt的多媒体模块，只需在项目文件 (.pro) 中增加“QT += multimedia”即可。

113.2 QSound

QSound类用于播放音频文件，包含如下方法：

```
1 |           QSound::QSound(const QString& filename); // 构造函数，参数为音频文件的
   | 路径
2 |           void QSound::play(); // 槽函数，播放
3 |           void QSound::stop(); // 槽函数，停止
4 | static void QSound::play(const QString& filename); // 静态函数，播放参数中指定的
   | 音频文件
```

113.3 QMediaPlayer

QMediaPlayer类表示通用的媒体播放器，包含如下方法：

```
1 | void QMediaPlayer::setMedia(const QMediaContent& media); // 设置媒体内容
2 | void QMediaPlayer::setVolume(int volume); // 槽函数，设置音量
3 | void QMediaPlayer::play(); // 槽函数，播放
4 | void QMediaPlayer::pause(); // 槽函数，暂停
5 | void QMediaPlayer::stop(); // 槽函数，停止
```

113.4 LAV Filters

如果不能正常播放所指定的多媒体文件，且在调试窗格看到类似“doRender: Unknown error 0x80040266”的错误提示。在排除多媒体文件损坏的前提下，可以认定为系统中缺少DirectShow解码器。毕竟Qt的Windows版本，底层使用的就是DirectShow Player Service。

DirectShow解码器，即LAV Filters，可以从这里下载：<https://lav-filters.en.lo4d.com/download>。

113.5 案例

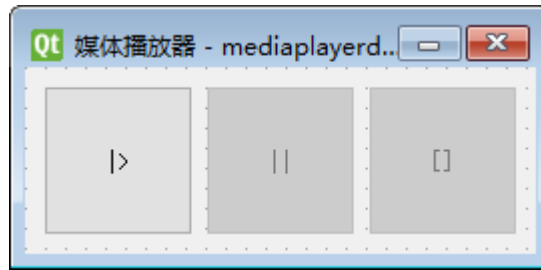
113.5.1 创建项目

通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为MediaPlayer的项目。

C:\Users\Minwei\Projects\Qt\MediaPlayer\MediaPlayer.pro：

```
1 | ...
2 | QT += ... multimedia
3 | ...
```

113.5.2 设计界面



对象	类
MediaPlayerDialog	QDialog
m_btnPause	QPushButton
m_btnPlay	QPushButton
m_btnStop	QPushButton

C:\Users\Minwei\Projects\Qt\MediaPlayer\mediaplayerdialog.ui:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3 <class>MediaPlayerDialog</class>
4 <widget class="QDialog" name="MediaPlayerDialog">
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>255</width>
10 <height>93</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>媒体播放器</string>
15 </property>
16 <layout class="QHBoxLayout" name="m_layout">
17 <item>
18 <widget class="QPushButton" name="m_btnPlay">
19 <property name="sizePolicy">
20 <sizepolicy hsize="Minimum" vsize="Preferred">
21 <horstretch>0</horstretch>
22 <verstretch>0</verstretch>
23 </sizepolicy>
24 </property>
25 <property name="text">
26 <string>|&gt;</string>
27 </property>
28 </widget>
29 </item>
30 <item>
31 <widget class="QPushButton" name="m_btnPause">
32 <property name="enabled">
33 <bool>>false</bool>
34 </property>
35 <property name="sizePolicy">
36 <sizepolicy hsize="Minimum" vsize="Preferred">
```

```

37     <horstretch>0</horstretch>
38     <verstretch>0</verstretch>
39 </sizepolicy>
40 </property>
41 <property name="text">
42     <string>||</string>
43 </property>
44 </widget>
45 </item>
46 <item>
47     <widget class="QPushButton" name="m_btnStop">
48     <property name="enabled">
49     <bool>>false</bool>
50 </property>
51 <property name="sizePolicy">
52     <sizepolicy hsize="Minimum" vsize="Preferred">
53     <horstretch>0</horstretch>
54     <verstretch>0</verstretch>
55 </sizepolicy>
56 </property>
57 <property name="text">
58     <string>[]</string>
59 </property>
60 </widget>
61 </item>
62 </layout>
63 </widget>
64 <resources/>
65 <connections/>
66 </ui>

```

113.5.3 实现功能

C:\Users\Minwei\Projects\Qt\MediaPlayer\mediaplayerdialog.h:

```

1  #ifndef MEDIAPLAYERDIALOG_H
2  #define MEDIAPLAYERDIALOG_H
3
4  #include <QDialog>
5  #include <QMediaPlayer>
6
7  QT_BEGIN_NAMESPACE
8  namespace Ui { class MediaPlayerDialog; }
9  QT_END_NAMESPACE
10
11 class MediaPlayerDialog : public QDialog
12 {
13     Q_OBJECT
14
15 public:
16     MediaPlayerDialog(QWidget *parent = nullptr);
17     ~MediaPlayerDialog();
18
19 private slots:
20     void on_m_mediaPlayer_stateChanged(

```

```

21     QMediaPlayer::State state);
22
23 private:
24     Ui::MediaPlayerDialog *ui;
25     QMediaPlayer* m_mediaPlayer;
26 };
27
28 #endif // MEDIAPLAYERDIALOG_H

```

C:\Users\Minwei\Projects\Qt\MediaPlayer\mediaplayerdialog.cpp:

```

1  #include "mediaplayerdialog.h"
2  #include "ui_mediaplayerdialog.h"
3
4  MediaPlayerDialog::MediaPlayerDialog(QWidget *parent)
5      : QDialog(parent)
6      , ui(new Ui::MediaPlayerDialog)
7      , m_mediaPlayer(new QMediaPlayer(this))
8  {
9      ui->setupUi(this);
10
11     m_mediaPlayer->setMedia(QUrl::fromLocalFile("sax.flac"));
12
13     connect(ui->m_btnPlay, SIGNAL(clicked()), m_mediaPlayer, SLOT(play()));
14     connect(ui->m_btnStop, SIGNAL(clicked()), m_mediaPlayer, SLOT(stop()));
15
16     connect(m_mediaPlayer, SIGNAL(stateChanged(QMediaPlayer::State)),
17            this, SLOT(on_m_mediaPlayer_stateChanged(QMediaPlayer::State)));
18 }
19
20 MediaPlayerDialog::~MediaPlayerDialog()
21 {
22     delete ui;
23 }
24
25 void MediaPlayerDialog::on_m_mediaPlayer_stateChanged(
26     QMediaPlayer::State state)
27 {
28     switch (state)
29     {
30         case QMediaPlayer::PlayingState:
31             ui->m_btnPlay->setEnabled(false);
32             ui->m_btnPause->setEnabled(true);
33             connect(ui->m_btnPause, SIGNAL(clicked()),
34                    m_mediaPlayer, SLOT(pause()));
35             ui->m_btnStop->setEnabled(true);
36             break;
37
38         case QMediaPlayer::PausedState:
39             connect(ui->m_btnPause, SIGNAL(clicked()),
40                    m_mediaPlayer, SLOT(play()));
41             break;
42
43         case QMediaPlayer::StoppedState:
44             ui->m_btnPlay->setEnabled(true);

```

```
45         ui->m_btnPause->setEnabled(false);
46         ui->m_btnStop->setEnabled(false);
47         break;
48     }
49 }
```

113.5.4 测试验证

运行效果如图所示：

