Игры и другие типы приложений, которые используют мультимедиа (типа звуков, анимации) должны включать дополнительные файлы мультимедиа вместе с приложением или включать файлы внутрь исполнимого файла.

Вы можете добавлять данные в Ваше приложение как ресурс, а затем извлекать эти данные, когда это необходимо. Эта методика более желательна, так как это может предохранить другие файлы от изменения их и предохранения.

В этой статье будет рассмотрено, как включать звуковые файлы, видео, анимацию и другие двоичные файлы в исполнимый файл Delphi.

Файлы ресурсов (.RES)

Включение нескольких двоичных файлов в исполнимый состоит из 5 шагов:

- Составьте и/или соберите все файлы, которые необходимо включить в EXE
- Создайте файл сценария ресурсов **.RC**, который описывает все ресурсы, исользуемые Вашим приложением
- Скомпилируйте файл сценария ресурсов **.RC**, чтобы создать файл ресурсов **.RES**
- Свяжите скомпилированный файл ресурсов с исполнимым файлом
- Используйте индивидуальные элементы ресурсов

Первый шаг очень прост: просто решите какие файлы Вы хотите хранить в исполнимом. Например, мы будем хранить две *.WAV* песни, одну *.AVI* видео и одну *.MP3* песню.

Прежде, чем продолжить, есть несколько важных заявлений относительно ограничений при работе с ресурсами:

а) Загрузка и выгрузка ресурсов на время выполнения операции. Ресурсы - часть исполнимого файла приложения и загружается во время выполнения приложения.

б) Вся свободная память может использоваться при загрузке/выгрузке ресурсов. Другими словами, нет никаких пределов в количестве ресурсов, загруженных одновременно.

в) Конечно, файл ресурсов увеличивает размер исполнимого. Если Вы хотите уменьшить размер исполнимого файла, рассмотрите размещение ресурсов и части Вашего проекта в *DLL* и пакетах.

Теперь давайте посмотрим, как создать файл, который описывает ресурсы.

Создание файла сценария ресурсов (.RC)

Файл сценария ресурсов - это простой текстовый файл с расширением **.RC**, который перечисляет ресурсы. Файл сценария имеет следующий формат:

ResName1 ResTYPE1 ResFileName1 ResName2 ResTYPE2 ResFileName2 { ... }
ResNameX ResTYPEX ResFileNameX { ... }

ResName определяет или уникальное имя или целое число (ID), который обозначает ресурс.

ResType описывает тип ресурса

ResFileName - полный путь и имя файла, который будет включен в ресурс.

Для создания нового файла сценания ресурсов проделайте следующее:

- Создайте новый текстовый файл в каталоге проектов
- Переименуйте его в *MyRes.rc*

В *MyRes.rc* записано следующее:

```
Clock WAVE "c:mysoundsprojectsclock.wav"  MailBeep WAVE
"c:windowsmedianewmail.wav"  Cool AVI cool.avi  Intro RCDATA introsong.mp3
```

Файл сценария просто определяет ресурсы. В данном формате *MyRes.rc* перечисляет два WAV файла, один AVI и одну песню в MP3 формате. Все инструкции в
.RC
файле связывают имя ресурса, тип и имя файла для данного ресурса. Есть около дюжины предопределенных типов ресурсов. Они включают иконки, точечные рисунки, курсоры, анимацию и т.д.
RCDATA
определяет пользовательский тип данных.
RCDATA
позволяет включать любой ресурс данных для приложения. Пользовательские ресурсы данных разрешают включение двоичных данных непосредственно в исполняемом файле. Например, заявление
RCDATA
выше называет двоичный ресурс приложения
Intro
и определяет файл, который содержит песню в MP3 формате.

Примечание: удостоверьтесь, что у Вас имеются все ресурсы, перечисленные в Вашем файле **.RC**. Если файлы внутри каталога проектов, то Вы не должны включать полный путь к файлам.

Создание файла ресурсов (.RES)

Чтобы использовать ресурсы, определенные в файле сценария ресурсов, мы должны скомпилировать его в **.RES** файл при помощи **Borland Resource Compiler**. Компилятор ресурсов создает новый файл, основанный на содержании файла сценария ресурсов. Этот файл имеет обычно расширение
.RES
. Компоновщик Delphi позже будет повторно форматировать
.RES
файл в файл объекта ресурсов, а затем связывать его с исполнимым файлом приложения.

Bopland Resource Compiler находится в каталоге *Delphi/Bin*. Имя - **BRCC32.exe**. Просто введите в
командной строке

brcc32.exe
и нажмите
Enter
. Компилятор запустится и отобразит справку использования (без параметров в командной строке).

Подготовьте файл *MyRes.rc* и выполните команду (в каталоге проектов):

```
BRCC32.exe MyRes.rc
```

По умолчанию, при компилировании ресурсов, **BRCC32.exe** называет скомпилированный файл с расширением **.RES** с именем
.RC
файла и помещает его в тот же каталог, что и
.RC
.

Подключение файла ресурсов к исполнимому файлу

Мы создали файл ресурсов. В следующем шаге мы должны добавить следующую директиву компилятора в модуле нашего проекта сразу после директивы формы (ниже ключевого слова **implementation**).

```
{$R *.DFM} {$R AboutDelphi.RES}
```

Случайно не удалите *{$R *.DFM}*, так как эта строка кода сообщает Delphi о подключении визуальной части формы. Когда Вы выбираете точечные рисунки для компонентов **SpeedButton, Image** или **Button**, Delphi включает растровый файл, который Вы выбрали как часть ресурса формы.

После того, как .RES файл будет связан с исполняемым файлом, приложение может загружать его ресурсы во время выполнения, когда это будет необходимо. Чтобы фактически использовать ресурс, Вы должны будете сделать несколько вызовов *API*.

Нам нужен будет новый проект с пустой формой. Добавьте директиву *{$R MyRes.res}* в модуль главной формы. Как уже было сказано выше, мы будем иметь дело с функциями Windows API.

Например, метод **LoadFromResourceName** объекта **TBitmap**. Этот метод извлекает указанный растровый ресурс и назначает его объекту
TBitmap
. Это аналогично вызову функции LoadBitmap в Windows API. Как всегда, Delphi улучшил вызов функции API, чтобы удовлетворить наши потребности лучше.

Проигрывание анимации из ресурсов

Чтобы показать анимацию *cool.avi* (который был определен в нашем файле ресурсов), мы будем использовать компонент **TAnimate** из палитры компонентов Win32, который нужно поместить на главную форму. Оставьте его имя по умолчанию -
Animate1
. Мы будем использовать событие
OnCreate
формы, чтобы отобразить анимацию:

```
procedure TForm1.FormCreate(Sender: TObject) ;  begin   with Animate1 do begin
ResName := 'cool';    ResHandle := hInstance;      Active := TRUE;   end;  end;
```

Все просто! Как мы можем видеть, чтобы запустить анимацию из ресурсов, мы должны использовать свойства *ResHandle, ResName* или *ResID* компонента **TAnimate**. После установки
ResHandle
, мы устанавливаем свойство
ResName
, чтобы определить какой ресурс является видеоклипом AVI, который должен будет отображен управлением
TAnimate
. Установка свойства
Active
в
True
просто запускает анимацию.

Проигрывание WAV

Так как мы включили два файла WAV в наше приложение, теперь посмотрим как извлечь звук из файла и запустить его. Поместите кнопку **Button1** на форму и напишите следующий код в обработчик события
OnClick

:

```
uses mmsystem;  { ... }  procedure TForm1.Button1Click(Sender: TObject) ;  var    hFind, hRes: THandle;    Song: PChar;  begin   hFind := FindResource(HInstance, 'MailBeep', 'WAVE') ;   if hFind  0 then begin     hRes:=LoadResource(HInstance, hFind) ;     if hRes  0 then begin  Song:= LockResource(hRes) ;       if Assigned(Song) then SndPlaySound(          Song, snd_ASync or snd_Memory) ;       UnlockResource(hRes) ;     end;     FreeResource(hFind) ;  end;  end;
```

Этот подход использует несколько вызовов API для загрузки ресурса типа WAV по имени *MailBeep* и запустить его. Обратите внимание: Вы можете использовать Delphi, чтобы запустить предопределенные системой звуки.

Проигрывание MP3

Единственный файл MP3 имеет имя **Intro**. Так как ресурс имеет тип **RCDATA**, мы будем использовать другую методику, чтобы извлеч и запустить MP3 песню.
TMediaPlayer
может запустить MP3 файл.

Теперь добавьте компонент **TMediaPlayer** на форму (*MediaPlayer1*) и добавьте **TButton** (*Button2*
). Событие
OnClick
должно быть таким:

```
procedure TForm1.Button2Click(Sender: TObject) ;  var   rStream: TResourceStream;  fStream: TFileStream;   fname: string;  begin   {эта часть извлекает mp3 из exe}   fname := ExtractFileDir(Paramstr(0)) + 'Intro.mp3';   rStream := TResourceStream.Create(hInstance, 'Intro', RT_RCDATA) ;   try    fStream := TFileStream.Create(fname, fmCreate) ;    try  fStream.CopyFrom(rStream, 0) ;    finally    fStream.Free;    end;  finally    rStream.Free;   end;   {эта часть проигрывает mp3}   MediaPlayer1.Close;   MediaPlayer1.FileName:= fname;  MediaPlayer1.Open;  end;
```

Этот код при помощи **TResourceStream** извлекает MP3 из exe и сохраняет его в рабочем каталоге приложения. Имя файла MP3 - Intro.mp3. Затем просто назначьте его свойству *FileName*
компонента
TMediaPlayer
и воспроизведите песню.

Но, присутствует одна небольшая проблема - приложение создает файл MP3 на машине пользователя. Вы можете добавить код, чтобы удалить этот файл перед закрытием прилоожения.

Автор: Zarko Gajic, About.com