

## 第十七讲 **Windows**高级编程

---

吕松茂

lsmtech@163.com

## 内容提要

---

- ⚙ 动态链接库
- ⚙ 自定义组件
- ⚙ OLE

## 内容提要

---

 动态链接库

⚙ 自定义组件

⚙ OLE

## 一、动态链接库

---

### 1.1 动态链接库（DLL）简介

动态链接库(Dynamic Link Library, 简称DLL)是一些编译过的可执行程序模块, 它包含代码, 数据或资源, 可以在应用程序或其他DLL中调用动态链接库的文件扩展名一般为.dll, 也可以是.dev (设备驱动程序)、.sys (系统文件) 和.fon (字体文件), DLL可以实现多个应用程序共享代码和资源。

## 一、动态链接库

---

### 动态链接库（DLL）工作原理

使用DLL的动态链接并不是将库代码拷贝，只是在程序中记录了函数的入口点和接口，在程序执行时才将库代码装入内存。所以不管多少程序使用了DLL，内存中都只有该DLL的一个副本。当没有程序使用它时，系统就将它移出内存，减少了对内存和磁盘的要求。

动态链接库属于Windows可执行文件，但它又不是EXE文件，它不像EXE文件那样可以直接执行，DLL文件中包含的可执行代码是由EXE文件调用的。

## 一、动态链接库

---

### 动态链接库（DLL）特点

DLL最大的特点就是它的代码在运行期间被动态地链接至调用它的程序中。它不用重复编译或链接，一旦装入内存，DLL函数可以被系统中的任何正在运行的应用程序所使用，他们共享该DLL函数的单一拷贝。

DLL中一般由程序通用的过程、函数等构成，当然也可以包括各种资源。在创建Windows应用程序时，链接过程并不把DLL文件中的例程链接到程序上，只有当EXE文件运行并需要调用一个DLL文件中的函数或过程时，Windows 才在DLL中寻找被调用函数并把它的地址传递给调用程序。

## 一、动态链接库

---

### 动态链接库（DLL）的应用范围

在如下几种情况下，调用DLL是合理的：

- ✿ 不同的程序使用相同的DLL，这样只需要将DLL在内存中装载一次，节省了内存的开销。
- ✿ 当某些内容需要升级的时候，使用DLL只需要改变DLL就可以了，而不需要把整个程序都进行变动
- ✿ 由于DLL是独立于语言的，所以，当不同语言习惯的人共同开发一个大型项目的时候，使用DLL便于程序系统的交流。当然，Delphi开发的DLL也可以在诸如Visual BASIC，C++等系统中使用。

## 一、动态链接库

### 1.2 创建DLL

通过Delphi主菜单的【File】|【New】|【Other】在弹出的【New】对话框中，选择DLL Wizard图标，单击【OK】按钮，系统将自动创建一个DLL项目。



2012-4-5

第十七讲：Windows高级编程

8/26



## 一、动态链接库

```
library Project1;  
  
( Important note about DLL memory management: ShareMem must be the  
  first unit in your library's USES clause AND your project's (select  
  Project-View Source) USES clause if your DLL exports any procedures or  
  functions that pass strings as parameters or function results. This  
  applies to all strings passed to and from your DLL--even those that  
  are nested in records and classes. ShareMem is the interface unit to  
  the BORLNDMM.DLL shared memory manager, which must be deployed along  
  with your DLL. To avoid using BORLNDMM.DLL, pass string information  
  using PChar or ShortString parameters. )  
  
uses  
  SysUtils,  
  Classes;  
  
{$R *.res}  
  
begin  
end.
```

## 一、动态链接库

---

### DLL和Application项目文件的格式对比

**DLL**项目文件主要格式:

Library 项目名;

Uses 子句;

Exports 子句; //数据接口函数

Begin //程序执行体

End.

**Application**主要格式:

Program 项目名;

Uses 子句;

Begin //程序执行体

End.

## 一、动态链接库

---

DLL项目文件和一般项目文件主要存在两个方面的区别：

(1) Application项目文件用Program关键字作程序头，而DLL项目文件用Library关键字作程序头，因此编译器会根据不同的关键字来生成不同的可执行文件。用Program关键字生成的是.exe文件，而Library关键字生成的是.dll文件。

(2) 第二个区别是DLL提供接口都是Exports关键字来实现的。在DLL项目文件中，将我们想要输出的函数或过程，列在Exports子句中，就可以实现输出了。

## 一、动态链接库

### 1.3 编写DLL函数

```
uses
  SysUtils,
  Classes;

function DllAdd(i : integer) : integer; stdCall;
begin
  result := i + 3;
end;

procedure DllMessage(strMessage : string); stdCall;
begin
  ShowMessage('DLL函数提供的消息: ' + strMessage);
end;

Exports
  DllAdd name 'DllAdd',
  DllMessage name 'DllMessage';

{$R *.res}

begin
end;
```

2012-4-5

第十七讲: Windows高级编程

12/26

## 一、动态链接库

---

### 1.4 DLL文件的静态调用

DLL文件的调用一般有两种方法，即静态调用方法和动态调用方法，静态调用方式，就是在本单元的Interface部分用External指示字列出要从DLL中调用的过程；动态调用方式是通过调用Windows API中的LoadLibrary函数、GetProcAddress函数和FreeLibrary函数来实现DLL文件的动态调用。

## 一、动态链接库

---

```
implementation

const
  MyDll='PMyDll.dll';

{Import routine from PMyDll.dll}
function DllAdd(i : integer) : integer; stdCall; external MyDll;
//将数字转换成大写
function CapitalNumber(N1:String):string; stdCall; external MyDll;
//将金额转换成大写
function CapitalMoney(N2:String):String; stdCall; external MyDll;
procedure DllMessage(strMessage : string); stdCall; external MyDll;

($R *.dfm)
```

## 内容提要

---

⚙️ 动态链接库

👉 自定义组件

⚙️ OLE

## 二、自定义组件

---

### 2.1 创建组件的步骤

- ✿ 确定一个祖先类。
- ✿ 创建一个组件单元。
- ✿ 在新组件中添加属性、方法和事件。
- ✿ 测试该组件。
- ✿ 在Delphi中注册该组件。
- ✿ 为该组件建立帮助文件



## 二、自定义组件

---

### 2.2 VCL中的组件基类

- ❁TObject : 所有组件的父类。
- ❁TComponent 这是非可视组件的起点，它的特点是在设计时能够以流的方式在IDE上存取。
- ❁TGraphicControl 创建一个不需要窗口句柄、但要在屏幕上显示的组件时，用这个类作为祖先类。
- ❁TWinControl 所有需要窗口句柄的组件，都应以此类为基类。该类提供了Windows组件的一般属性和事件。
- ❁TCustomControl 该类是从TWinControl继承下来的。它具有Canvas属性和Paint()方法，能够控制组件的外观。也用于需要句柄的组件

## 二、自定义组件

### 2.3 创建组件

通过 Delphi 菜单的【Component】|【New Component】创建组件。



## 二、自定义组件

### 组件单元文件

```
unit LSHEdit;

interface

uses
  SysUtils, Classes, Controls, StdCtrls;

type
  TLSHEdit = class(TEdit)
  private
    { Private declarations }
  protected
    { Protected declarations }
  public
    { Public declarations }
  published
    { Published declarations }
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('LSH', [TLSHEdit]);
end;

end.
```

2012-4-5

第十七讲：Windows高级编程

19/26

## 二、自定义组件

### 编写组件相关事件

```
type
  TLSHEdit3 = class(TEdit)
  private
    { Private declarations }
  protected
    { Protected declarations }
  public
    constructor Create(AOwner: TComponent); override;
  published
    { Published declarations }
  end;

procedure Register;

implementation

procedure Register;
begin
  RegisterComponents('LSH3', [TLSHEdit3]);
end;

constructor TLSHEdit3.Create(AOwner: TComponent); (这个在单元的implementation 那
begin
  [Inherited Create(AOwner); (必须在那: 必须用? )
  Color := clYellow; (设置新的值)
end;

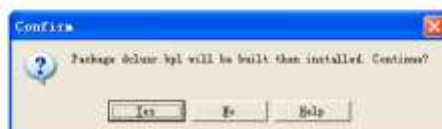
end.
```

2012-4-5

6

## 二、自定义组件

### 2.4 安装组件



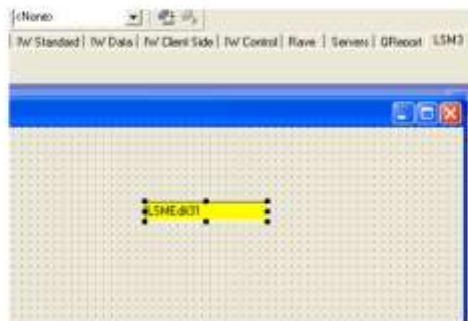
2012-4-5

第十七讲： Windows高级编程

21/26

## 二、自定义组件

### 2.5 测试组件



2012-4-5

第十七讲：Windows高级编程

22/26

## 内容提要

---

⚙️ 动态链接库

⚙️ 自定义组件

🗨️ OLE

## 三、OLE

---

### 3.1 OLE介绍

❁对象链接和嵌入(Object Linking and Embedding)是一组服务功能，它提供了一种用源于不同应用程序的信息创建复合文档的强有力方法。对象可以是几乎所有的信息类型，如文字、位图、矢量图形，甚至于声音注解和录像剪辑等。



## 三、OLE

---

### 3.2 OLE的使用

✿例：通过OLE调用Excel将数据导出

## 总结

---

- ⚙ 动态链接库
- ⚙ 自定义组件
- ⚙ OLE

谢谢大家

