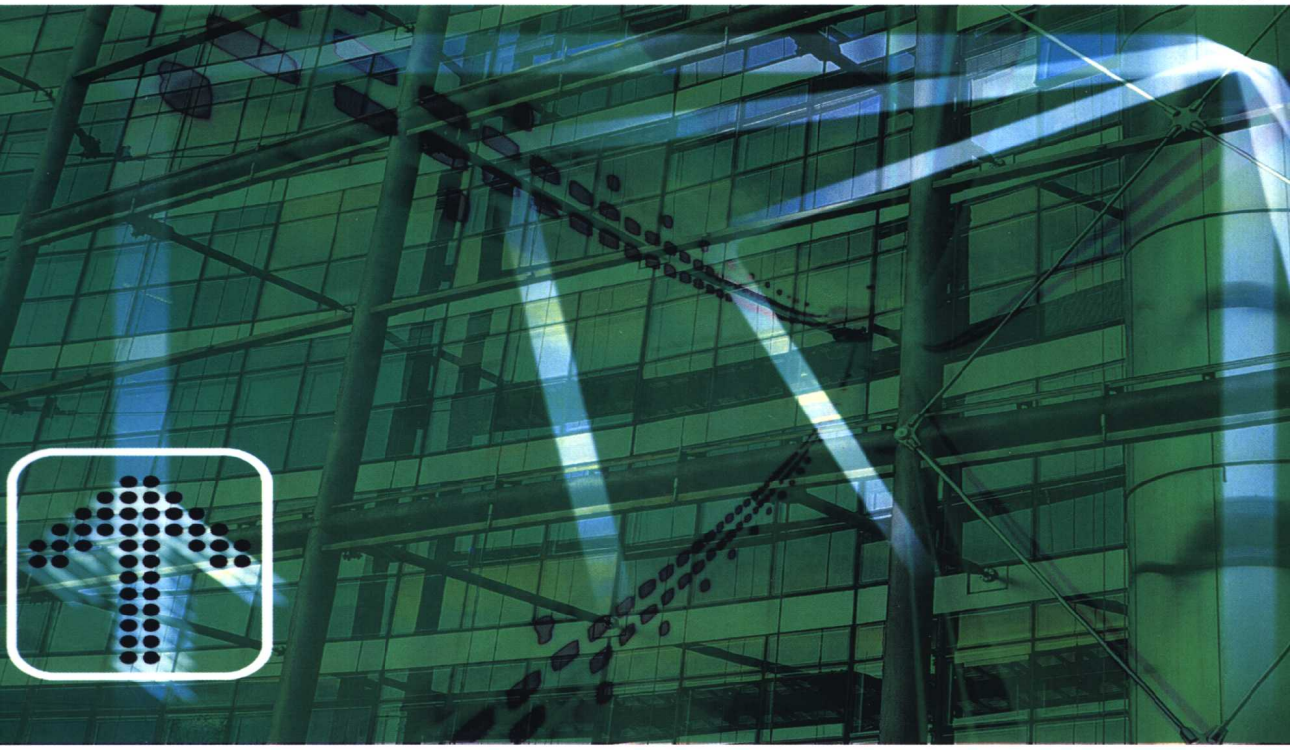
万水软件项目应用与实例开发丛书

COMPANION WEB SITE!

Delphi

深度编程及其项目应用开发

李存斌　汪兵　编著



中国水利水电出版社
www.waterpub.com.cn

万水软件项目应用与实例开发丛书

Delphi 深度编程及其项目应用开发

李存斌　汪　兵　编著

中国水利水电出版社

内 容 提 要

本书是在总结作者多年 Delphi 开发经验的基础上编著而成。全书分为基础篇和应用篇。基础篇结合示例论述了 Delphi 的深度编程技术，其中包括 9 章，分别为：理解 Windows 消息、进程与线程、自定义组件的编写、文件操作、创建 DLL 应用程序、两层数据库应用程序、多层数据库应用程序、Socket 编程、串口编程；应用篇结合物资管理信息系统项目应用开发技术和经验，详细阐述了一般管理信息系统软件通用模块的开发，其中包括 10 章，分别为：物资管理信息系统概述及其总体框架设计、物资管理信息系统后台数据库设计、应用服务器的实现、客户端应用程序的设计、动态连接应用服务器的实现、通用权限管理模块的设计、通用查询组件和报表模块的制作、通用基础数据维护模块的设计、物资管理信息系统业务操作模块的设计、综合查询模块的设计。读者在具有一定 Delphi 知识的基础上，通过本书的学习，可快速提高 Delphi 的编程能力和实际开发水平。

本书适用于具有初步编程能力的读者，也可作为高校高年级学生毕业设计的指导书。中国水利水电出版社网站（www.waterpub.com.cn）上包括了书中示例和较为完整的物资管理信息系统的源代码文件，为读者的学习提供方便，同时也为相关软件开发人员的实际应用开发提供捷径和参考。

图书在版编目（CIP）数据

Delphi 深度编程及其项目应用开发/李存斌，汪兵编著．－北京：中国水利水电出版社，2002

（万水软件项目应用与实例开发丛书）

ISBN 7-5084-1213-3

Ⅰ.D… Ⅱ.①李…②汪… Ⅲ. DELPHI 语言－程序设计，Ⅳ. TP312

中国版本图书馆 CIP 数据核字（2002）第 072609 号

书　　名	Delphi 深度编程及其项目应用开发
作　　者	李存斌　汪兵　编著
出版、发行	中国水利水电出版社（北京市三里河路 6 号　100044） 网址：www.waterpub.com.cn E-mail：mchannel@public3.bta.net.cn（万水） sale@waterpub.com.cn 电话：（010）68359286（万水）、63202266（总机）、68331835（发行部）
经　　售	全国各地新华书店
排　　版	北京万水电子信息有限公司
印　　刷	北京市天竺颖华印刷厂
规　　格	787×1000 毫米　16 开本　　25.25 印张　　556 千字
版　　次	2002 年 9 月第一版　2002 年 9 月北京第一次印刷
印　　数	0001—5000 册
定　　价	42.00 元

凡购买我社图书，如有缺页、倒页、脱页的，本社发行部负责调换

丛书前言

随着经济全球化、管理现代化的到来，信息化建设已提到议事日程。国家机关、高等院校、参与国际竞争的大企业等企事业单位，为了提高现代化管理水平、为了在激烈的市场竞争中获胜，纷纷不同程度地加强信息化建设，如组建自己的局域网、实施适用本单位或本部门的企业级 MIS（管理信息系统）项目或 ERP（企业资源计划）项目。在网络环境下对企业 MIS 或 ERP 项目的开发，对开发语言和工具提出了更高的要求。目前，关于开发语言和工具的书籍层出不穷，但结合实际应用软件开发项目的书籍不多。

基于此，我们组织长期从事软件项目开发的教师编写了本套“软件项目应用与实例开发丛书”。该套丛书的编撰特点：结合软件项目，注重适用性，可操作性强。丛书中的每本书都按照一个软件项目的开发顺序安排内容，以一个完整实例（如物资管理信息系统）为主线，将知识点融于实例中，实例的编写结合编著者多年的开发经验，如在录入窗体的按钮设计、查询模块的条件表达式输入等方面都有一定的技巧和经验。

中国水利水电出版社网站（www.waterpub.com.cn）上包含书中所有调试通过的源程序，对源程序中难以理解的语句增加了注释说明。主要的源程序以“程序清单 2-1”方式在书中列出，相应源程序的一组文件（如项目文件、窗体文件等）均放在网站上，以方便读者学习调用。

本套丛书目前共有 7 本，内容涵盖了当前流行的可视化软件开发语言和工具，它们是：

- Delphi 深度编程及其项目应用开发
- Visual FoxPro 高级编程及其项目应用开发
- Visual Basic 高级编程及其项目应用开发
- Visual C++高级编程及其项目应用开发
- PowerBuilder 高级编程及其项目应用开发
- ASP 高级编程及其项目应用开发
- Java 高级编程及其项目应用开发

本套丛书适用于具有初步编程能力而急需掌握应用程序设计方法、提高软件项目应用开发水平的读者；也可供高校高年级本科生毕业设计、研究生课题研究以及教师教学时参考。对于从事软件项目开发的工程技术人员具有一定的参考价值。

李存斌

2002 年 8 月于北京

前　言

Delphi 是 Inprise 公司近年来推出的优秀开发语言与工具。Delphi 以其友好的集成开发界面、面向对象的可视化开发模式、良好的数据库及多媒体应用支持以及高效的软件开发与程序运行，备受广大软件开发人员的好评。尤其是 2001 年 6 月问世的 Delphi 6 版本，功能更强大、开发效率更高，不仅是网络环境下的优秀前端开发语言和工具，也是服务器端 Web 编程的优秀工具。

本书是 Delphi 的入门培训教材《Delphi 6 程序设计及其应用开发》（李存斌主编，中国水利水电出版社 2002 年 3 月出版）的姊妹篇，适用于有一定 Delphi 知识的用户。通过本书的系统学习，读者可以迅速掌握在短期内开发高质量应用软件的技术，快速提高 Delphi 的编程能力和实际开发水平。

本书是在作者总结多年 Delphi 开发经验的基础上编著而成的。全书共分两篇，第 1 篇为基础篇，结合示例论述了 Delphi 的深度编程技术，其中包括 9 章，分别为：理解 Windows 消息、进程与线程、自定义组件的编写、文件操作、创建 DLL 应用程序、两层数据库应用程序、多层数据库应用程序、Socket 编程、串口编程。第 2 篇为应用篇，结合物资管理信息系统项目应用开发技术和经验，详细阐述了一般管理信息系统软件通用模块的开发过程，其中包括 10 章，分别为：物资管理信息系统概述及其总体框架设计、物资管理信息系统后台数据库设计、应用服务器的实现、客户端应用程序的设计、动态连接应用服务器的实现、通用权限管理模块的设计、通用查询组件和报表模块的制作、通用基础数据维护模块的设计、物资管理信息系统业务操作模块的设计和综合查询模块的设计。

本书可作为高等院校有关教师的教学参考书或高年级学生或研究生的自学用书，也可作为有关软件开发人员的参考书。中国水利水电出版社网站（www.waterpub.com.cn）上包括了书中示例和较为完整的物资管理信息系统的源代码文件，为读者的学习提供方便，同时也为有关人员的实际应用开发提供捷径和参考。

本书由李存斌、汪兵编著，同时参加编写工作的还有陈鹏、李天华、沈党国、黄铁英、王哲、王冬放、郭晓鹏、樊建平、高丽林、陈钢、邓振杰、齐建玲、田惠英、董富贵等同志。在本书的编著出版过程中，得到了中国水利水电出版社计算机编辑室全体同志的大力支持和帮助，在此一并表示衷心的感谢。

由于时间仓促和作者水平所限，书中错误和不妥之处在所难免，敬请读者批评指正。

编者

2002 年 8 月

目　　录

应用篇：物资管理信息系统项目应用开发

基础篇：Delphi 深度编程技术

第 1 章　理解 Windows 消息

消息是 Windows 的重要概念，通过消息 Windows 可以处理用户输入和协调系统应用程序。理解和使用 Windows 消息，可以使程序的编写更为灵活、处理的范围更为广泛、处理的手段更为专业。因此作为一名高级 Windows 程序员，理解 Windows 消息处理机制是非常必要的，并将加深对 Windows 编程的认识。本章将介绍 Delphi 中处理消息、编程者截获消息、处理消息、发送消息以及自定义消息的内容。

本章主要包括以下内容：

- 消息概述
- Windows 消息的工作原理
- Delphi 的 VCL 消息系统处理原理
- 发送消息
- 处理消息
- 消息过滤

1.1　消息概述

消息就像是人类社会中人与人之间的对话和交流，而在计算机系统中消息就是 Windows 发出的一个通知，告诉应用程序某个事情发生了，然后相关的对象就去执行特定的操作。举个常见的例子：当单击鼠标或者按下键盘上的某个键时，Windows 都会发送一个消息给应用程序，通知应用程序发生的事情或者需要执行的操作。

消息本身是作为一个记录传递给应用程序的，记录中包含消息的类型以及其他的信息。这个记录类型叫做 TMsg，它在 Windows 单元中的声明如下：

```
Type
TMsg=packedRecord
    Hwnd:HWND;          //窗口句柄
    Message:UNIT;       //消息常量标识符
    WParam:WPARAM;      //32 位消息的特定附加信息
    LParam:LPARAM;      //32 位信息的特定附加信息
    time:DWORD;         //消息创建的时间
```

```
    pt:Tpoint;          //消息创建时的鼠标位置
end;
```

在 Win32 中预定义的一些消息常量往往是以 WM 开头，以代表某一特定的消息，如 WM_Activate 代表窗体激活消息，WM_Close 代表窗体关闭消息。最初的时候，Windows 程序员必须详细地了解每个消息中参数的意义，后来微软公司对消息的参数作了命名，使得每条消息的参数变得容易理解。例如按下键盘上某键的消息被定义为如下记录形式：

```
Type
TWMKeyDown = record
     Msg:Cardinal;
     CharCode:Word;
     KeyData:Longint;
     Result:Longint;
end;
```

由以上定义可见，虽然每个参数都有具体的命名，但是消息的结构仍然是记录，占用 12 个字节。Delphi 的 Message 单元中定义了所有的 Windows 消息，有兴趣的读者可以打开 Message 单元研究一下。

1.2 Windows 消息工作机理

开发基于 Windows 平台的应用程序时，了解 Windows 消息系统的组成是非常必要的。Windows 的消息系统由以下 3 部分组成：

- 消息队列：Windows 能够为所有的应用程序维护一个消息队列，应用程序必须从消息队列中获取消息，然后分派给某个窗口。
- 消息循环：通过这个循环机制，应用程序从消息队列中检索消息，再把它分派给适当的窗口，然后继续从消息队列中检索下一条消息，再分派给适当的窗口，依次进行。
- 窗口过程：每个窗口都有一个窗口过程，以接收 Windows 传递给窗口的消息，窗口过程的任务就是获取消息并且响应它。窗口过程是一个回调函数，处理完一个消息后，通常要给 Windows 一个返回值。

从消息的产生到消息被一个窗口响应，这其中要经历 5 个步骤：

（1）系统中发生了某个事件。

（2）Windows 把这个事件翻译成消息，然后把它放到消息队列中。

（3）应用程序从消息队列中接收这个消息，并把它存放在 TMsg 记录中。

（4）应用程序把消息传递给一个适当的窗口过程。

（5）窗口过程响应这个消息并进行处理。

步骤 3 和步骤 4 构成了应用程序的消息循环。由于消息循环能够使应用程序响应外部事件，因此消息循环往往就是 Windows 应用程序的核心。消息循环的任务是从消息队列中检索消息并且把消息传递给适当的窗口。如果消息队列中没有消息，Windows 就允许其他应用程

序处理它们的消息。图 1-1 显示了上述 5 个步骤。

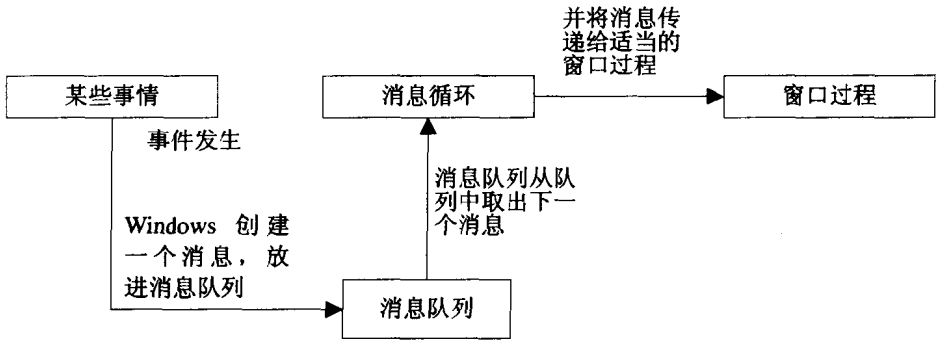


图 1-1　Windows 消息系统

1.3　Delphi 的 VCL 消息系统处理原理

Delphi 中的每一个 VCL 组件（如 TButton，TEdit 等）都有内在的消息处理机制，在建立一个窗体或者加入一个组件时，VCL 就已经注册了一个消息接受例程 MainWndProc，这个例程是每个窗体和组件所固有的。Delphi 内部使用这个例程接收来自各方面的消息并把它们发送给适当的处理方法，如果没有特定的处理方法，则调用缺省的消息处理句柄。例程 MainWndProc 是定义在 TWinControl 类中的一个静态方法，不能被重载。它不直接处理消息，而是交由 WndProc 方法处理并提供异常保护。如果有异常发生，则调用 Application.HandleException 方法处理异常。MainWndProc 方法的声明如下：

```
Procedure MainWndProc(var Message: TMessage);
```

WndProc是在TControl类中定义的一个虚拟方法，这意味着可以覆盖它，提供自定义的消息处理例程。在此例程中，可以对自己关心的消息进行截获、处理或扔掉它。对于自己不处理的消息，最好调用继承让父类去处理，这是非常好的编程习惯。WndProc调用Dispatch方法进行消息的分配，WndProc方法的声明如下：

```
Procedure WndProc (var Message:TMessage);virtual;
```

Dispatch 方法是在 TObject 根类中定义的，其声明如下：

```
Procedure TObject.Dispatch (var Message);
```

传递给 Dispatch 方法的消息参数必须是一个记录类型，这个记录中的第一个点必须是一个 Cardinal 类型的域（field），它包含了要分配的消息的消息号码。例如：

```
type
   TMessage = Record
     Msg:Cardinal;
     Wparam:word;
     Lparam:longint;
     Result:longint;
```

```
End;
```

而 Dispatch 方法将根据消息号码调用组件的最后继承类中处理此消息的句柄方法。如果此组件和它的祖先类中都没有对应此消息的处理句柄，Dispatch 方法便会调用 DefaultHandler 方法。DefaultHandler 方法是定义于 TObject 中的虚拟方法，其声明如下：

```
Procedure DefaultHandler (var Message);virtual;
```

TObject 类中的 DefaultHandler 方法将只实现简单的返回而不对消息进行任何处理。我们可以通过对此虚拟方法的重载，在子类中实现对消息的缺省处理。对于 VCL 中的组件而言，其 DefaultHandler 方法会启动 Windows API 函数 DefWindowProc 对消息进行处理。其处理流程可概括如下：如果父类的 WndProc 被覆盖的话，消息同样要被筛选一次。不管父类 WndProc 是否被覆盖，在继承中调用祖先类的 WndProc，这样一直追溯到 TControl 类的 WndProc。在 TControl.WndProc 中，它调用最终祖先 TObject 的 Dispatch 派发消息。假如 MainWndProc 收到一条消息，沿着继承的路线调用 WndProc，如果一路上的 WndProc 都未处理该消息，那么调用到 TObject.Dispatch 寻找一个例程来处理这条消息。

TObject.Dispacth 派发消息时也是按照继承链逆流而上的。首先查找本类中用 Message 定义的消息处理方法，如果覆盖的话，则调用它来处理，没有覆盖就继续查找父类。首先从 Message 定义方法查起，然后再看看 DefaultHandler 中是否覆盖，这样一直找到 TObject 的 DefaultHandler。在 TObject.DefaultHandler 中，什么消息都不处理，只是走到了消息处理的头。

在这个漫长的筛选和派发过程中，有两个中间类值得注意，它们是 TWindowControl 和 TControl 类。TControl 继承了 TObject，而 TWinControl 又继承了 TControl。一般的组件都是从这两个类继承来的。这两个类都对 WndProc 和 DefaultHandler 作了覆盖。

其消息处理流程如图 1-2 所示。

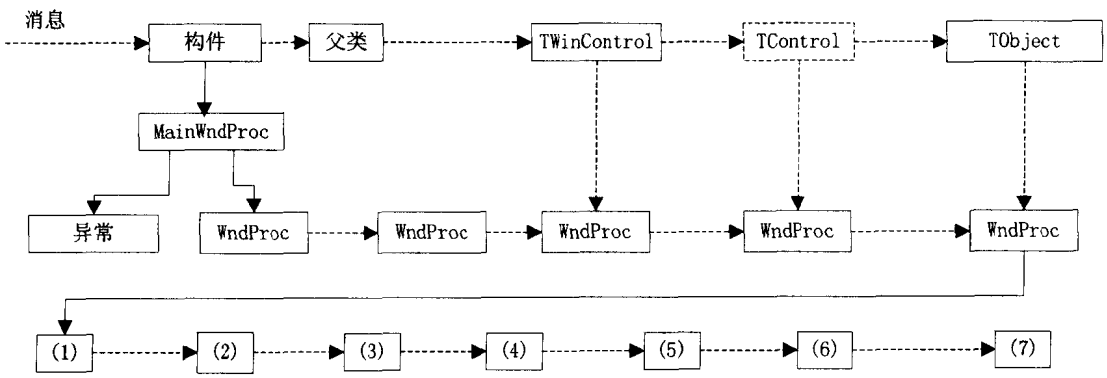


图 1-2 Delphi 消息处理流程图

处理消息的虚拟方法如下：

TControl.WndProc 定义了鼠标的基本消息，如单击和拖动。其他消息调用 TObject.Dispatch。

TWinControl.WndProc 覆盖了 TControl.WinProc，定义对聚焦、鼠标键盘消息的响应。TControl.DefaultHandler 处理 Windows 管理组件的文本串的消息。

TWinControl.DefaultHandler 通过调用 API 函数 CallWindowProc 来处理各种消息。

1.4　发送消息

在 Windows 中主要通过 3 种方式发送消息：调用 Perform()；调用 SendMessage()；调用 PostMessage()。

1.4.1　Perform()

VCL 的 Perform()适用于所有 TControl 派生对象。只要知道 Form 或组件的实例，Perform()可以发送消息给任何一个 Form 或组件。

Perform 函数在 TControl 类中的声明如下：

```
Function TControl.Perform(Msg:Cardinal;Wparam,Lparam:Longint);Longint;
```

1.4.2　SendMessage()和 PostMessage()

SendMessage()函数和 PostMessage()函数是 Windows 中两个发送消息的 API 函数，SendMessage()直接把一个消息发送给窗口程序并等消息处理完之后返回。PostMessage()只是把消息发送给消息队列，然后立即返回，并不需要知道消息是否处理完了。

SendMessage()函数的声明如下：

```
Function SendMessage(hWnd:HWND;Msg:UNIT;Wparam:WPARAM;Lparam:LPARAM):LRESULT;
stdcall;
```

PostMessage()函数的声明如下：

```
Function PostMessage(hWnd:HWND;Msg:UNIT;Wparam: WPARAM;Lparam: LPARAM):
LRESULT;stdcall;
```

SendMessage 函数和 PostMessage 函数的参数是一致的。参数 hWnd 指接收消息的窗口句柄；参数 Msg 指消息标识符；参数 Wparam 指 32 位的特定附加信息；参数 Lparam 指 32 位的特定附加信息。

1.4.3　消息的发送

下面将分三种情况讲述消息的发送。

1. *在应用程序内发送消息*

一个应用程序发送消息给自己是很容易的，只需要调用上面我们所讲的 Perform()、SendMessage()和 PostMessage()函数，并且使消息常量的值为 WM_USER+100 到$7FFF 即可。例如：

```
Const WM_MyMessage=WM_USER+200;
Begin
```

```
  SomeForm.perform(WM_MyMessage,0,0);
  //或者
  SendMessage(someForm.handle,WM_MyMessage,0,0);
  //或者
  PostMessage(someForm.handle,WM_MyMessage,0,0);
End;
```

然后声明和定义一个普通的消息处理过程处理 WM_MyMessage 消息：

```
Tform1=class(TForm)
.
.
private
procedure doMyMessage(var msg:TMessage);message WM_Mymessage;
end;

procedure Tform1.doMyMessage(var Msg:TMessage);
begin
//作想要的处理
end;
```

2. 在应用程序之间发送消息

如果需要在两个或多个应用程序之间发送消息，那么最好调用 RegisterWindowMessage() 函数，这个函数能够确保每个应用程序使用一致的消息序号。

RegisterWindowMessage()函数需要传递一个以 null 结束的字符串，并返回一个范围从 $C000 到$FFFFF 的新的消息常量。这意味着在要发送消息的应用程序之间，每个应用程序都必须传递相同的字符串给 RegisterWindowMessage()函数，而 Windows 也会返回相同的消息常量，这可以确保给定的字符串返回整个系统中惟一的消息常量，这样就可以放心地向所有窗口发送消息，此函数在第二章的进程间通讯中用到了，读者可以做参考。

3. 广播消息

TWinControl 的派生对象可以调用 Broadcast()向它的子组件广播一个消息，当需要向一组组件发送相同的消息时，便可以使用这种技术。例如，Panel1 可以给它的所有子组件发送一个叫 UM_FOO 的自定义消息，代码如下：

```
var
M:TMessage;
begin
  With M do
  begin
    Msg:=UM_FOO;
    wParam:=0;
    lParam:=0;
    Result:=0;
  end;
  panel1.Broadcast(M);
end;
```