

# 第七讲 Delphi中的事件和消息

---

吕松茂

lsmtech@163.com

# 内容提要

---

✿ **Windows事件和消息机制**

✿ **Delphi消息处理机制**

✿ **键盘事件**

✿ **鼠标事件**

# 内容提要

---

## Windows事件和消息机制

### ✿ Delphi消息处理机制

### ✿ 键盘事件

### ✿ 鼠标事件

# 一、Windows事件和消息机制

---

## 1.1 Windows事件和事件驱动程序编程

✿ 处理Windows发往应用程序的消息一直是传统Windows编程的关键内容之一，深刻理解Window的消息机制是编写出优秀Windows应用程序的基础。

✿ Delphi提供了大多数常用消息的处理方法，并把Windows消息转化为组件开发者和使用者所能处理的事件。

# 一、Windows事件和消息机制

---

## 1.1 Windows事件和事件驱动程序编程

✿事件是组件的重要组成部分，它连接对应的系统事件和为此而编写的代码（即为事件处理过程）。

✿事件的应用，使开发者不需要改变组件本身就能设置组件的行为。

✿当用户对组件进行某些操作时，如单击组件就会产生一个事件。还有一些事件由系统自动产生，如改变该组件的一个特殊属性值。

# 一、Windows事件和消息机制

---

## 1.2 Windows的消息机制

✿消息是Windows发出的一个通知，它告诉应用程序某个事件发生了。

✿Windows并不通知应用程序应该完成什么工作，而是通知应用程序发生了什么事情，这便是所谓的消息，然后，由截获消息的应用程序自己决定如何去处理这些事件。

# 一、Windows事件和消息机制

---

## 1.2 Windows的消息机制

✿ 一个应用程序一旦启动，Windows便为该应用程序建立一个消息队列（**message queue**），开始且陆续地将与该应用程序有关的消息加入它的队尾。应用程序则要逐个地将其消息队列中的队头消息取出，筛选出有用的消息，再指派给某个窗口过程，由其对消息进行处理和做出响应。从某种意义上讲，Windows的应用程序不过是一个消息处理器。事件就是被消息所驱动的。每个应用程序内部必定有一段循环程序，负责检索其消息队列，这段特殊的程序称为消息循环（**message loop**）。

# 一、Windows事件和消息机制

---

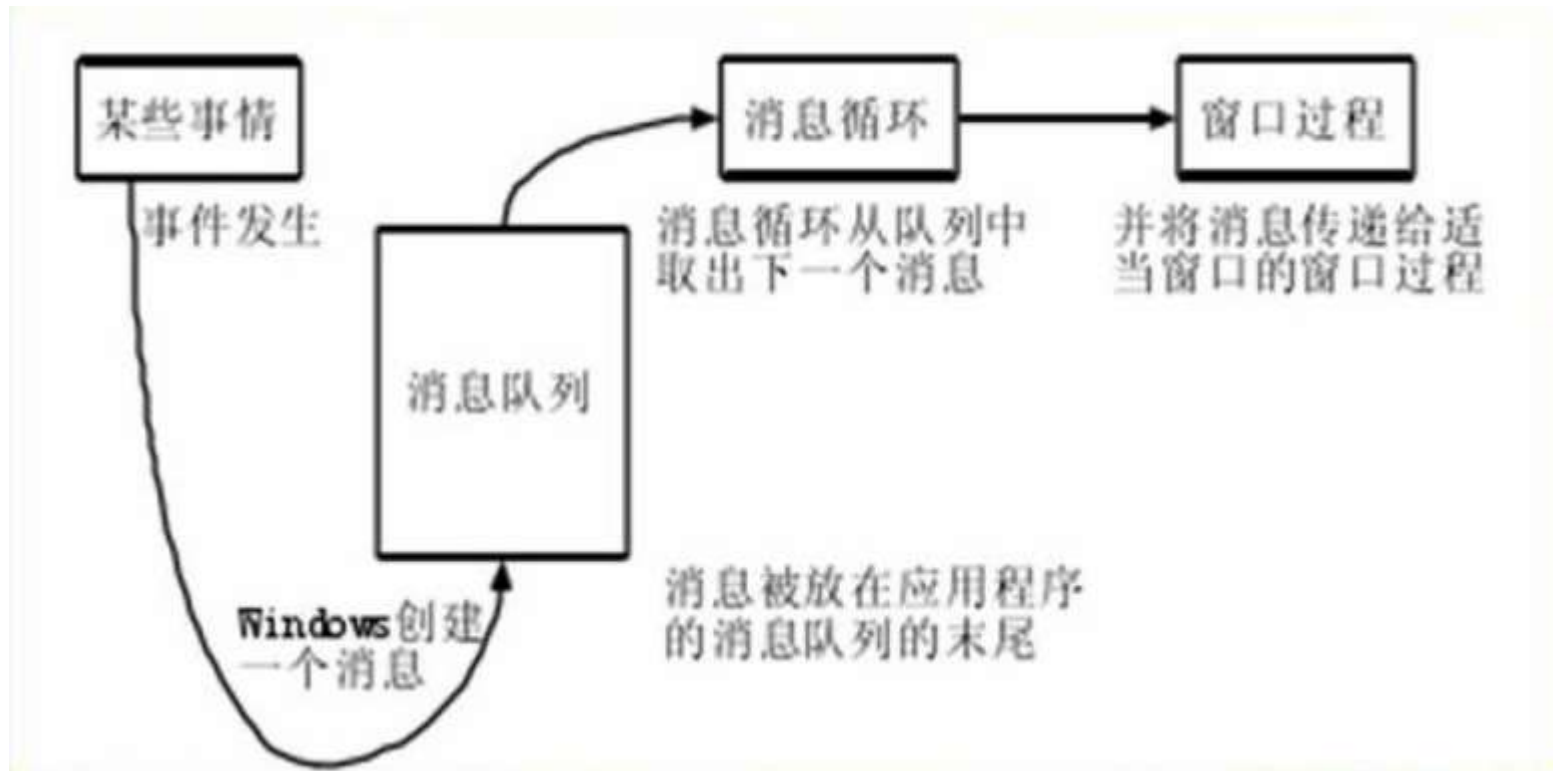
## Windows消息系统的组成

- ✿消息队列：Windows为每个应用程序维护一个消息队列。应用程序从消息队列中获取消息，然后分派给某个窗口。
- ✿消息循环：应用程序循环地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，循环进行。
- ✿窗口过程：每个窗口都有一个窗口过程来接收传递给它的消息，它的任务就是获取消息然后响应它。窗口过程是一个回调函数，处理了一个消息后，它通常要返回一个值给Windows。



# 一、Windows事件和消息机制

## Windows应用的消息循环过程



# 一、Windows事件和消息机制

---

消息的数据结构在Windows中被定义为如下的记录类型：

```
TMessage = record  
    Msg: Cardinal;  
    WParam: Word;  
    LParam: Longint;  
    Result: Longint;  
end;
```

# 一、Windows事件和消息机制

---

消息定义为如下的TMsg记录类型：

TMsg = packed record

    hwnd: HWND;

    message: UINT;

    wParam: WPARAM;

    lParam: LPARAM;

    time: DWORD;

    pt: TPoint;

end;

# 内容提要

---

✿ **Windows**事件和消息机制

 **Delphi**消息处理机制

✿ 键盘事件

✿ 鼠标事件

## 二、Delphi消息处理机制

---

### 1.1 Delphi的消息处理

✿所有Delphi提供的对象都有内建的消息处理机制。其基本思想是：对象接收某些类型的消息，并发送出去，继而调用一组与该消息相关的消息处理方法。

✿若调用消息的特定方法不存在，则Delphi调用缺省处理。

## 二、Delphi消息处理机制

---

### 1.1 Delphi的消息处理

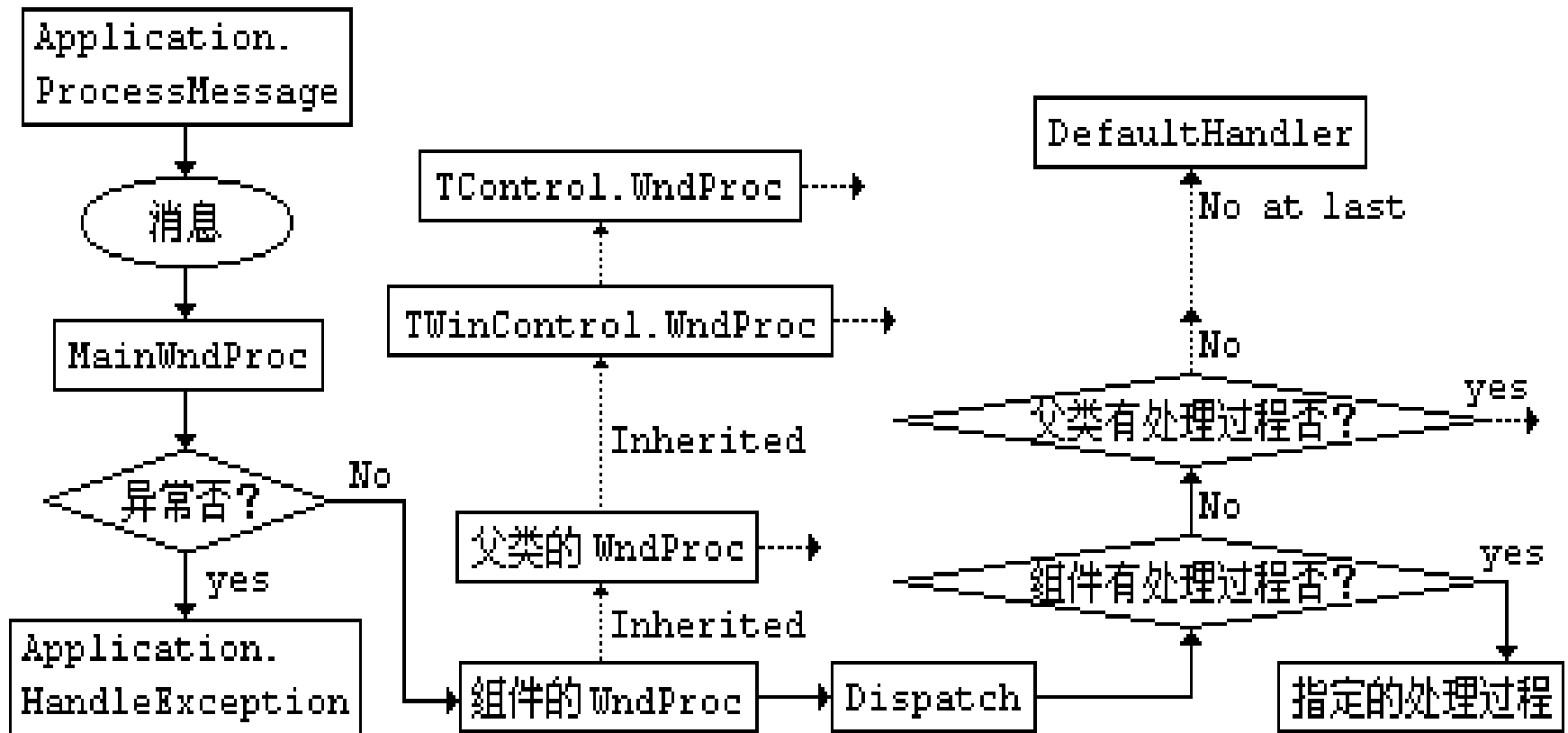
✿当某个事件满足触发条件时，就会发送特定的消息到Windows系统消息队列。

✿Delphi为应用程序中每种类型的组件注册了名为WinWndProc的方法作为窗口过程。

✿窗口过程同时包含了异常处理模块、完成从Windows到名为WndProc的虚方法的消息记录传送、并调用应用程序对象的HandleException方法处理异常。

## 二、Delphi消息处理机制

### 1.1 Delphi的消息处理



## 二、Delphi消息处理机制

---

### 1.2 发送消息

下列两个Windows API函数均可实现消息的发送。

✿function SendMessage(hWnd:HWND; Msg:UINT;  
wParam:WPARAM; lParam:LPARAM)

:LRESULT; stdcall;

✿function PostMessage(hWnd:HWND; Msg:UINT;  
wParam:WPARAM; lParam:LPARAM)

: BOOL; stdcall;



# 一、Delphi消息处理机制

---

## 1.4 响应消息

方法1. 覆盖组件继承的虚拟方法WndProc

```
procedure TNewComponent.WndProc(var message:TMessage);  
begin  
    if message.Msg = 需要过滤的消息 then  
    begin  
        // 此处编写处理过滤消息代码  
    exit;  
    end;  
    inherited WndProc(message);  
    dispatch(message);  
end;
```

# 一、Delphi消息处理机制

---

## 1.4 响应消息

方法2. 覆盖组件继承的虚拟方法

DefaultHandler

在继承类中对DefaultHandler过程进行覆盖，同样可以实现类似于前一种方法的消息过滤。

## 二、Delphi消息处理机制

---

### 1.3 响应消息

#### 方法3. 自定义消息和消息处理方法

Windows预定义了1024个编码从\$0000到\$03FF的消息常量标识符，留出从\$0400到\$7FFF的编码空间用于用户自定义消息。其中的\$0400已定义为WM\_USER。因此，自定义消息常量标识符的形式如下：

const

WM\_UserMsgName = WM\_USER + N;      // 其中，  
0<=N<=31743

## 二、Delphi消息处理机制

---

### 1.3 响应消息

#### 方法4. TApplication的OnMessage事件

每当应用程序从消息队列中检索到一个消息时，便触发Application的OnMessage事件。由此，在消息被派送到对其响应和处理的目标对象之前，有机会处理该消息。

# 内容提要

---

✿ Windows事件和消息机制

✿ Delphi消息处理机制

 键盘事件

✿ 鼠标事件

## 三、键盘事件

### 3.1 关于键盘

在计算机发展过程中，一开始是就使用键盘作为输入方式，在**DOS**环境中，大多只有判断键盘是否单击，以及单击的是哪个键，在**Windows**之中，可以判断键盘单击的事件有三种如下图所示：



## 三、键盘事件

---

### 3.1 关于键盘

在这些事件中，OnKeyDown和OnKeyUp都会传入用户单击的**Key**值（**word**值），可以利用这些值，来判断用户按了哪些键，而这些值代表了Windows中的**Virtual Key Code**。而OnKeyPress所返回的是一个**Char**值，代表一个**ASCII**字符。

**ASCII**字符和**Virtual Key Code**是不相同的，因为**Virtual Key Code**中有代表**ASCII**的字符，但**ASCII**字符不包含全部的**Virtual Key Code**，因为**Virtual Key Code**中内含了很多功能键。

# 三、键盘事件

---

## 3.2 键盘的常用事件

### 1. OnKeyDown

当按下键盘上的任一个键时，就会触发此事件。如字母键、数字键、功能键（F1~F12）、Ctrl键、Shift键或Alt键等，都将产生一个OnKeyDown事件。

### 2. OnKeyPress

当用户单击ASCII字符的键盘时，就是说当按下键盘上的一个字符键，如字母键、数字键等会产生一个OnKeyPress事件，但是单独按下功能键（F1~F12）、Ctrl键、Shift键或Alt键等，不会产生OnKeyPress事件。



## 三、键盘事件

---

### 3.2 键盘的常用事件

#### 3. OnKeyUp

当按下键盘上的任一个键后松开时，都会产生一个OnKeyUp事件。对于功能键（F1~F12）、Ctrl键、Shift键或Alt键等，也会产生一个OnKeyUp事件。

## 三、键盘事件

---

### 3.2 键盘的常用事件

#### 4. 检测功能键

在组件的OnKeyDown、OnKeyUp、OnMouseDown和OnMouseUp等事件的处理过程中，有一个TShiftState类型的变量Shift，TShiftState类型定义如下：

Type

```
TShiftState=setof(ssShift,ssAlt,ssCtrl,ssLeft,ssRight,ssMiddle,ssDouble);
```

根据Shift的值就可以判断当键盘上的键按下时Shift、Alt和Ctrl键的状态，或者按下鼠标左键、中键时的状态或者是否双击了按键。当然，如果有OnMouseDown事件发生了，而又不是按下左键和中键，则按下的一定是右键。

# 内容提要

---

✿ **Windows事件和消息机制**

✿ **Delphi消息处理机制**

✿ **键盘事件**

 **鼠标事件**

## 四、鼠标事件

---

常用鼠标的事件有以下几种：

### 1. 常用鼠标事件

(1)OnClick：当用户单击鼠标任何一个键时，就会触发此事件。

(2)OnMouseDown：当用户单击鼠标时，就会触发此事件。

(3)OnMouseMove：当用户单击鼠标在对象上移动时，就会触发此事件，但停止就不触发了。

## 四、鼠标事件

---

(4)OnMouseUp: 当鼠标的某个按键按下，然后松开后会产生一个此事件。

在这些事件中，OnMouseDown和OnMouseUp都会触发事件，但是在用户单击时，可能会移动鼠标位置，使得两者被触发的对象不同的，但OnClick和OnMouseDown是会触发在同一个对象上的。

当用户在对象A按一下时，A会同时触发  
OnMouseDown OnClick OnMouseUp OnMouseMove  
当用户在对象A单击但在对象B放开时，A会触发  
OnMouseDown OnMouseMove OnMouseUp

## 四、鼠标事件

---

### B会触发OnMouseMove

因此如果要触发OnClick事件，就必须要在一个对象按一下才行，否则只会有OnMouseDown、OnMouseMove、OnMouseUp这些事件。

### 2. 拖放事件

- (1)OnDragDrop: 在拖曳事件开始时会触发此事件。
- (2)OnDragOver: 当拖曳对象跨过一个组件时会触发此事件。
- (3)OnEndDrag: 当拖曳事件结束后会产生触发此事件。

## 四、鼠标事件

---

具体过程如下：

### (1) 拖曳操作开始

大多数的组件具有**DragMode**属性，表示开始拖曳操作的方式。**DragMode**属性的缺省值为**dmManual**，也就是要在被拖动组件的**OnMouseDown**事件的处理过程中调用**BeginDrag**过程才开始拖曳操作。如果将**DragMode**属性设置为**dmAutomation**，则鼠标左键在被拖动组件上按下后就自动开始拖曳操作。

### (2) 接受拖曳操作

当拖动一个组件经过第二个组件的时候，第

## 四、鼠标事件

---

二个组件会产生一个OnDragOver事件。在该事件的处理过程中有一个布尔类型的参数，该参数的设置直接影响是否产生OnDragDrop事件。

一般情况下，在OnDragOver事件的处理过程中，根据参数Source判断拖曳操作的源。如果是可以接受的源，则将Accept参数设置为True；否则，将其设置为False。

### (3)处理拖曳操作

在第二个组件OnDragDrop事件处理过程中，根据拖曳操作的源做一些相应的处理。

### (4)拖曳操作结束



## 四、鼠标事件

---

拖曳操作完成后释放鼠标左键，会在第一个组件中产生一个**OnEndDrag**事件，可以根据参数**Target**的数值进行相应的处理。如果参数**Target**的值为**nil**，则表示拖曳操作没有被接受；如果**Target**的值不为**nil**，则**Target**的值就是接受拖曳操作的组件。

### 3. 滚轮事件

(1) **OnMouseWheel**: 当用户单击滚轮按钮时，就会触发此事件。

(2) **OnMouseWheelDown**: 当用户用滚轮按钮向下转动时，就会触发。

(3) **OnMouseWheelUp**: 当用户滚轮按钮向上转动时，就会触发。

# 总结

---

✿ **Windows事件和消息机制**

✿ **Delphi消息处理机制**

✿ **键盘事件**

✿ **鼠标事件**

谢谢大家

