

关于捕获VCL没有处理的Windows消息 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/253/2021_2022__E5_85_B3_E4_BA_8E_E6_8D_95_E8_c97_253806.htm 对于C Builder的程序员来说，VCL以其灵活、高效的特点令人喜爱。因为VCL是在Windows API的基础上进行了封装，同时舍弃了一些不常用的功能，所以，VCL在功能上是Windows API的子集。VCL提供了对大多数Windows消息的处理机制，但是对于没有处理的Windows消息呢，在需要是如何捕获呢？C Builder采用了消息映射表机制，通过消息映射表将特定的Windows消息于代码中的函数联系起来，当窗口捕获到消息时就会调用这个函数。C Builder消息映射表定义形式如下：

```
BEGIN_MESSAGE_MAP MESSAGE_HANDLER(,,)
```

```
END_MESSAGE_MAP(ClassName) 其中：
```

BEGIN_MESSAGE_MAP：消息映射定义起始语句

MESSAGE_HANDLER：消息处理定义 END_MESSAGE_MAP

：消息映射定义结束语句 ClassName：是要接受消息的类名

message：是要截获的Windows消息 message handler：消息处理

函数名称 message structure：是用于传递给VCL消息的结构名称，

此结构里包含有处理消息时所需的全部参数，不同的消息

所对应的消息结构是不同的。由于每一个VCL控件（无论是窗体

还是按钮）都可以独立的接收Windows消息，并且进行独立的

响应，所以一定要注意消息定一种的ClassName参数。

现在举例说明消息定义、传递的应用。现在在一个窗

体FormMain，和2个TperformanceGraph控件（不能响应鼠标

事件），现在我要对2个 TperformanceGraph控件定义鼠标单

击事件，对FormMain也重定义鼠标单击事件，过程如下（假定工程文件为Message.bpr，程序文件为main.cpp、main.h）：源代码如下：

```
//----main.h-----  
----- #ifndef mainH #define mainH  
//----- #include  
#include #include #include #include "PERFGRAP.h"  
//----- class  
TFormMain : public TForm { __published: // IDE-managed  
Components //----2个标准TperformanceGraph控件  
TPerformanceGraph *PerformanceGraph1. TPerformanceGraph  
*PerformanceGraph2. TEdit *Edit2. TEdit *Edit1. void __fastcall  
FormCreate(TObject *Sender). private: // User declarations //----自  
定义的消息处理函数，其中MESSAGE可以不写 MESSAGE  
void __fastcall LButtonDown(TMessage amp.message). void  
__fastcall MyWndProc2(TMessage amp.message) { //----如果鼠标  
在窗体内（控件外）被单击，则此事件被激活  
if(GetAsyncKeyState(VK_LBUTTON) {  
Application->MessageBoxA("Form Get Messsge", "Message",  
MB_OK). //----向Edit2发送鼠标消息，则Edit2将产生单击获得  
焦点的现象 SendMessage(FormMain->Edit2->Handle,  
message.Msg, message.WParam, message.LParam). } }  
//-----  
----- void __fastcall TFormMain::FormCreate(TObject *Sender) {  
//----替换2个TperformanceGraph控件的消息处理函数句柄  
OldWndProc1 = PerformanceGraph1->WindowProc.
```

```
OldWndProc2 = PerformanceGraph2->WindowProc.  
PerformanceGraph1->WindowProc = MyWndProc1.  
PerformanceGraph2->WindowProc = MyWndProc2. }
```

```
//-----
```

```
----- void __fastcall TFormMain::MyWndProc1(TMessage  
amp.message) { if (message.Msg == WM_LBUTTONDOWN)  
ShowMessage("PerformanceGraph 2 get Message"). else  
OldWndProc2(message). } 如果在TFormMain::FormCreate ( )  
函数中加入以下两句语句： OldWndProc3 =  
FormMain->WindowProc. FormMain->WindowProc =  
MyWndProc3. 再添加函数： void __fastcall  
TFormMain::MyWndProc3(TMessage &amp;message) { if  
(message.Msg == WM_LBUTTONDOWN) //----如果消息是 “  
鼠标单击消息” ，则显示信息 ShowMessage( “ FormMain Get  
the Message ”). else //----如果消息是其他消息，则交给控件原  
来的处理 OldWndProc1(message). } 则把FormMain的鼠标消息  
响应函数进行了改装，此时窗体就会截获鼠标消息，而2  
个TperformanceGraph控件将不会得到消息。这是因为窗体将  
先于控件得到消息。从上面的例子可以看出： 1、窗体将先  
于控件得到消息； 2、对于响应某一消息的控件，可以  
用SendMessage想起发送消息； 3、对于不响应某一消息的控  
件，可以用重载其WindowProc属性获得你想要的效果。实际  
上，掌握了利用Windows的消息传递机制才是掌握C Builder的  
第二阶段，任重而道远。 100Test 下载频道开通，各类考试题  
目直接下载。详细请访问 www.100test.com
```