

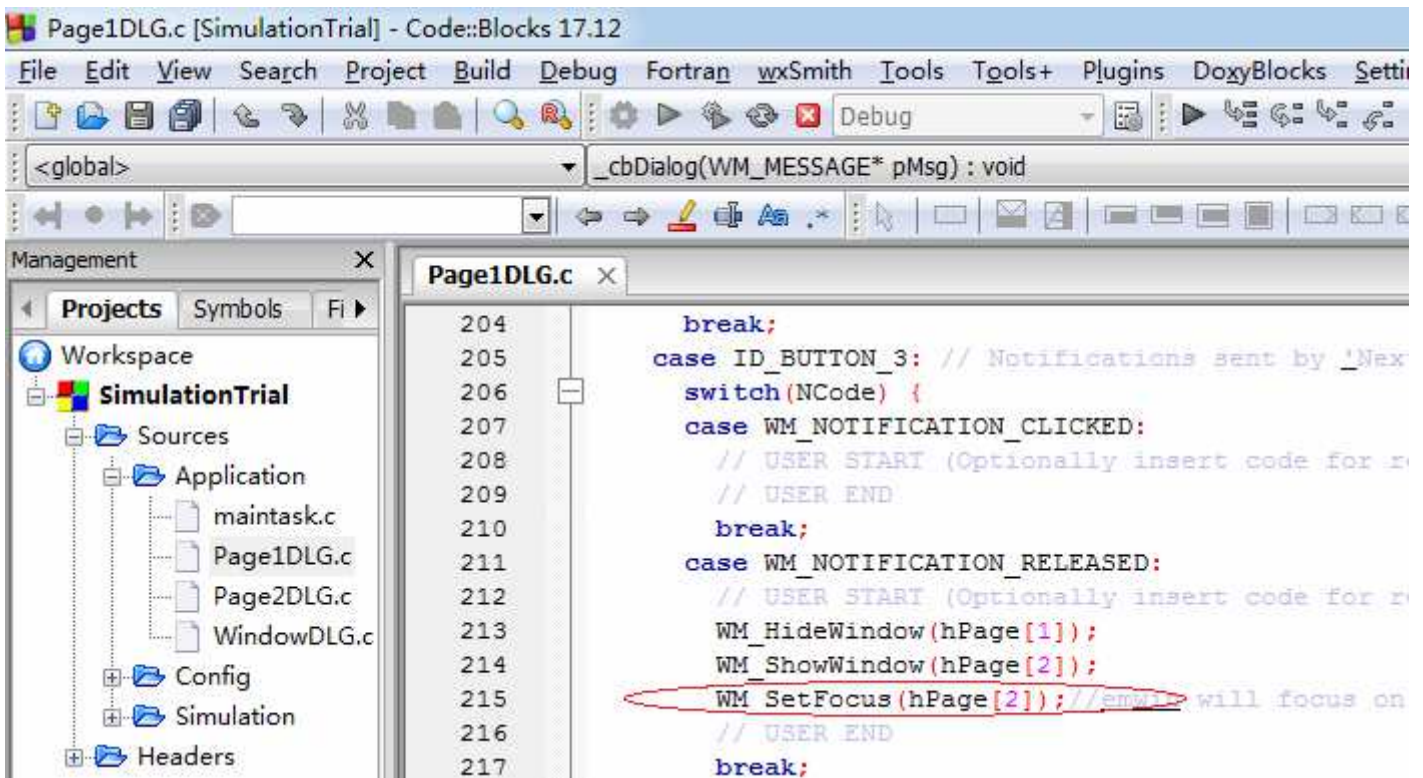
emwin 2-day quick tutorial 004_控件聚焦和用户消息的使用方法

emWin 的控件具有聚焦(Focus)功能,聚焦功能对于操作界面来说有时候是不需要的,但有时候是必不可少的.比如需要用键盘操作界面时,没有控件聚焦就没法决定操作哪个控件了,有聚焦的话可以通过 TAB 键将焦点移动到需要操作的控件,然后执行需要的操作.控件聚焦是 emWin 内部自己处理,我们作为用户是不需要理会的,但有时候我们需要将聚焦指定到哪个特定控件,我们就可以用 WM_SetFocus() 函数实现.

消息是 emWin 的基本元素,emWin 是通过消息和我们自己的程序交互的,emWin 可以发消息通知我们什么事情发生了,当然我们也可以发送消息给 emWin 告诉它要做什么. emWin 有固定的消息,比如:WM_INIT_DIALOG(创建窗体时通知我们需设置控件哪些初始参数)、WM_PAINT(通知我们窗体刷新绘制完毕)、WM_NOTIFY_PARENT(通知我们有控件被点击)等等...;当然 emWin 还可以让我们自己定义消息,这个是非常有用的,比如你在程序某个地方需要通知某个窗体做什么改变,这个时候你就自己定义一个消息,然后用 WM_SendMessage() 函数向这个窗体发送消息,在这个窗体的回调函数中加入这个消息和需要处理的事情即可.

在例程中,点击“Focus To Edit”之后,本该聚焦到该按键的,但在程序中使用 WM_SetFocus() 函数将焦点指定到了 Edit 控件;点击“Send Message”按键之后,在程序中使用 WM_SendMessage() 函数发送一个 WM_MSGUSER 消息(自己定义的)给自己的父窗口,再在 WM_MSGUSER 消息中改变 Edit 的值(这样做只是举例而已,其实是可以直接改变 Edit 的值).

另外,如果你用到键盘控制界面的话,聚焦就非常重要了!如果一个页面失去聚焦的话,页面将无法响应键盘的操作,程序也无法跑到按键触发消息(WM_KEY)!在例程程序中,WM_HideWindow()和 WM_ShowWindow()之后,页面并不会聚焦到当前页面,加入下面的函数就是为了聚焦到当前页面:



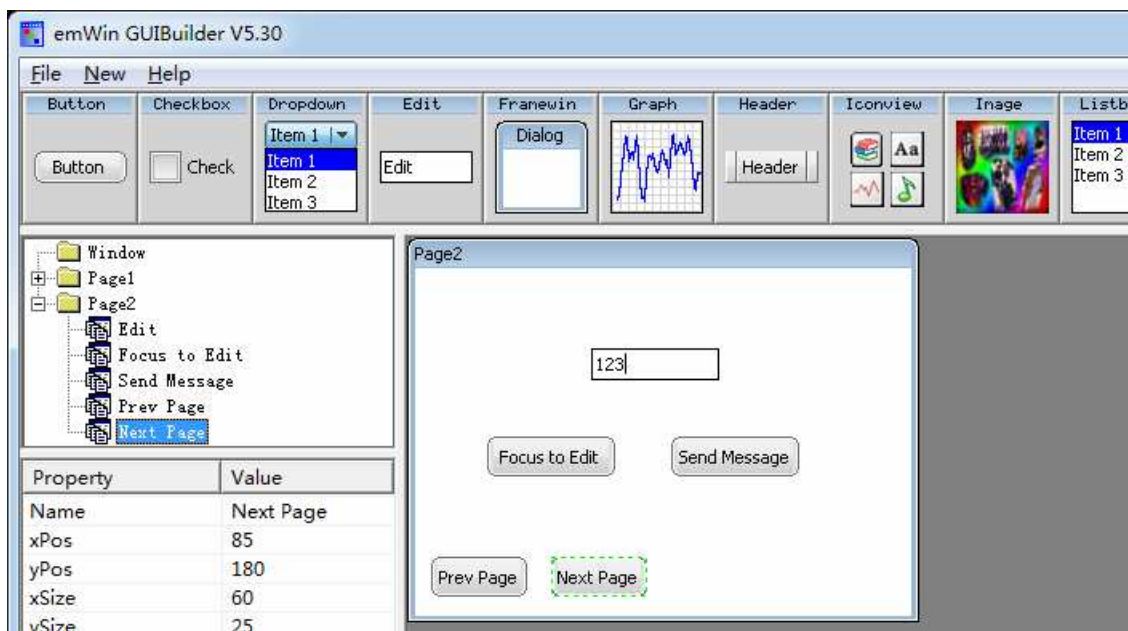
```
Page1DLG.c [SimulationTrial] - Code::Blocks 17.12
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Setti
<global>
_cbDialog(WM_MESSAGE* pMsg) : void
Management
Projects Symbols Fi
Workspace
SimulationTrial
Sources
Application
maintask.c
Page1DLG.c
Page2DLG.c
WindowDLG.c
Config
Simulation
Headers
204
205
206
207
208
209
210
211
212
213
214
215
216
217
break;
case ID_BUTTON_3: // Notifications sent by '_Nex
switch(NCode) {
case WM_NOTIFICATION_CLICKED:
// USER START (Optionally insert code for n
// USER END
break;
case WM_NOTIFICATION_RELEASED:
// USER START (Optionally insert code for n
WM_HideWindow(hPage[1]);
WM_ShowWindow(hPage[2]);
WM_SetFocus(hPage[2]); //emwin will focus on
// USER END
break;
```

```

Page1DLG.c x
204     break;
205     case ID_BUTTON_3: // Notifications sent by 'Next Page'
206     switch(NCode) {
207     case WM_NOTIFICATION_CLICKED:
208         // USER START (Optionally insert code for reacting on notification message)
209         // USER END
210         break;
211     case WM_NOTIFICATION_RELEASED:
212         // USER START (Optionally insert code for reacting on notification message)
213         WM_HideWindow(hPage[1]);
214         WM_ShowWindow(hPage[2]);
215         WM_SetFocus(WM_GetDlgItem(WM_GetClientWindow(hPage[2]), ID_BUTTON_0)); //
216         // USER END
217     break;

```

因为 ID_BUTTON_0 控件是在 Framewin 的 Client 窗口(有疑问请看:教程 002), 而 hPage[1] 是 Framewin 主窗口的句柄, 因此要获取 ID_BUTTON_0 控件的句柄应该使用 API 函数 WM_GetClientWindow(). 理解 Framewin 的 Client 区概念是非常必要的!



小技巧: 放置 xxxx 控件之后不要用鼠标移动, 用上下左右键移动更容易对齐(步进是 5).

备注: GUIBuilder 生成的 c 文件, 用户代码最好加在“USER START”和“USER END”之间, 其他地方除了数字以外, 不要做任何修改, 否则 GUIBuilder 将无法再次打开此 c 文件; 另外, GUIBuilder 再次打开编辑并保存时, “USER START”和“USER END”之间的内容将不会被更改; 还有 c 文件不要在 GUIBuilder 打开状态直接去修改 c 代码, 否则点 GUIBuilder 保存之后修改的内容将会丢失。



在 emWin 的说明书描述中, 消息实际上是一个结构体, 并且以 pMsg 指针的形式传递给回调函数:

```
static void _cbDialog(WM_MESSAGE * pMsg)
{
    ....
}
```

Elements of structure WM_MESSAGE

Data type	Element	Description
int	MsgId	Type of message. See list of messages below.
WM_HWIN	hWin	Destination window.
WM_HWIN	hWinSrc	Source window.
union	Data	Data union. See elements below.
Elements of Data		
void *	p	Data pointer.
int	v	Data value.

1) MsgId 表示是什么消息, 比如 emWin 的固定消息: WM_INIT_DIALOG (创建窗体时通知我们需要设置控件哪些初始参数)、WM_PAINT (通知我们窗体刷新绘制完毕)、WM_NOTIFY_PARENT (通知我们有控件被点击) 等等..., 当然还可以是我们自己起名的自定义消息:

```
message.MsgId = WM_MSGUSER;
WM_SendMessage(hWin, &message);
```

2) hWin 是目标窗口句柄, 即要发送消息到哪个窗口:

```
WM_SendMessage(hWin, &message);
```

需要注意的是要区分好哪个才是目标窗口,比如是要发送给 hFramewin 的主窗口还是发送给 hFramewin 的 Client 窗口:

```
WM_SendMessage(hFramewin, &message);
```

```
WM_SendMessage(WM_GetClientWindow(hFramewin), &message);
```

但是对于 pMsg->hWin, 不管 static void _cbDialog(WM_MESSAGE * pMsg) 是 Framewin 主窗口的回调函数, 还是 Client 窗口的回调函数, pMsg->hWin 表示的都是 Framewin 主窗口的句柄 (觉得奇怪也没办法, emWin 就是这样设计的).

3) hWinSrc 是源窗口/控件句柄, 即消息是由谁发出的, 在回调函数中有下面一句程序, 用来获取发出消息的那个控件的 ID:

```
Id = WM_GetId(pMsg->hWinSrc);
```

比如当你按下一个 Button 控件, 在 emWin 底层内部, 会发送一个消息到这个按键的父窗口, 那么 pMsg->hWinSrc 自然就是这个 Button 的句柄.

4) 在你要向某个窗口发送消息的时候, 还想附带个数据怎么办? 这时候就可以用到 Data.v 或 Data.p 元素, Data.v 可以储存一个数, Data.p 可以指向一个数据群.

比如在 TIMER 例程中有这样一句话: WM_RestartTimer(pMsg->Data.v, 100);, 在 emWin 底层内部, emWin 把 TIMER 的句柄存到 Data.v 了, 既然这个消息是 TIMER 发出的, 那么该 TIMER 句柄为什么不是 pMsg->hWinSrc? 可能是因为 TIMER 不是一个控件或窗口, 所以把句柄存到 Data.v 传过去.