

# 在 VC++ 中文本框控件如何通过回车键转移焦点

the Shift of Focus Between Editbox using ENTER in VC++

蔡伟丰 朱彦华 郑少慧 赵益建 陈泽铭 (广州铁路中心医院电脑室 510080)

**摘要:**本文介绍了 Visual C++ 6.0 在对话框的多个文本框之间,通过回车键转移焦点的几种方法,并提出了一个改进方法。

**关键词:**Visual C++ 6.0 文本框 回车键 焦点

## 1 引言

在使用 VC++ 进行各种编程时,往往要在对话框中设置多个文本框,用于用户输入各种信息,但是用户在输入数据时,默认方式下必须用 Tab 键在文本框之间转移焦点,如果用户输入回车键将可能使对话框退出,对于大多数用户是非常不方便。本文介绍了输入回车键在各个文本框中转移焦点的几种方法。

## 2 技术难点

在对话框中,如果用户在一个单行文本框中按回车键,Windows 将会自动寻找对话框的默认按钮,并执行默认按钮的代码,或者未找到默认按钮时,则会自动执行 OnOK() 函数,使对话框退出。对话框及文本框的 WM\_KEYDOWN 或 WM\_CHAR 消息都无法捕捉到回车键。因此如果能捕捉到回车键消息后向对话框发送转移焦点的 WM\_NEXTDLGCTL 消息,则可以解决此问题。

## 3 几种常见的处理方法

### 3.1 利用对话框的 WM\_COMMAND 及映射消息

当对话框或它的某一个控件具有输入焦点时,如果按回车键,Windows 将会发送 WM\_COMMAND 消息,并且将默认的按钮 ID 值传递给该消息的参数 wParam,如果没有默认按钮,将传递 IDOK 值,因此通过截取 WM\_COMMAND 的消息并判断 wParam 的值转移焦点。转移焦点的方法如下:首先获取当前焦点控件,并且判断是否为文本框控件,如果是则发送 WM\_NEXTDLGCTL 给对话框,使对话框将焦点移至下一个控件。示例代码如下:

```
case WM_COMMAND: //截取消息
    if(wParam == IDOFDEFAULTBUTTON || wParam == IDOK) // IDOFDEFAULTBUTTON 为默认按钮的 ID 值,若无默认按钮,则 wParam 值为 IDOK
    {
        CWnd * pWnd = GetFocus(); //获得当前焦点控
```

```
件
        BOOL bRetVal = TestIsEditWnd (pWnd -> m_hWnd); //判断是否为文本控件
        if(bRetVal) {
            PostMessage(hDlg, WM_NEXTDLGCTL, 0, 0L); //发送消息将焦点转移至下一控件
            return TRUE;
        }
        else {
            EndDialog(hDlg, TRUE); //结束对话框
            return FALSE;
        }
    }
    break;
```

### 3.2 子类化或超类化文本控件

当一个文本控件被子类化或超类化后,所有的键盘输入将被发送至自定义子类或超类函数中,而不管该对话框是否有默认按钮,通过在该函数中捕捉 WM\_KEYDOWN(或 WM\_CHAR)消息,加入相应代码即可。示例如下:

```
long FAR PASCAL MySubClassProc (HWND hWnd, WORD wMessage, WORD wParam, LONG lParam) {
    switch (wMessage)
    {
        case WM_CHAR: /
        if((wParam == VK_RETURN) || (wParam == VK_TAB)) /当获得回车键时,直接返回 0
            return 0;
        else
            return (CallWindowProc (lpOldProc, hWnd, wMessage, wParam, lParam)); //调用默认处理过程
        case WM_KEYDOWN:
            if ((wParam == VK_RETURN) || (wParam == VK_TAB)) { //捕捉到按回车键或 TAB 键时
```

```

PostMessage ( ghDlg, WM_NEXTDLGCTL, 0,
0L); // 发送消息给对话框,使焦点转移至下一控件
return FALSE; // 返回
}
return ( CallWindowProc ( lpOldProc, hWnd, wMes-
sage, wParam, lParam)); // 执行默认的消息
break ;
}

```

### 3.3 创建新的对话框成员函数

这种方法允许按回车键在各个文本框中来回转移焦点,即当前焦点为最后一个文本框时,当按回车键时将使焦点转移至第一个文本框控件。

实现步骤如下:

(1) 将对话框的默认 ID 值 IDOK 改为另一值,如 ID\_MYOK 并将该默认按钮改为非默认按钮;

(2) 为对话框的该按钮创建一个新的对话框成员函数,这个成员函数是 ID\_MYOK 控件的 BN\_CLICKED 消息;

(3) 在第二步创建的新成员函数中加入正常的处理代码;

(4) 在对话框中加入如下函数原型:

```

protected:
virtual void OnOK();

```

(5) 为 OnOK 函数加入转移焦点的代码,例如:

```

void CMyDialog::OnOK()
{
CWnd * pWndCtrl = GetFocus(); // 获得当前焦点控件
CWnd * pWndCtrlNext = pWndCtrl;
UINT controlId = pWndCtrl ->GetDlgCtrlID(); // 获
得焦点控件的 ID 值
switch (controlID) {
case IDC_EDIT1:
pwndCtrlNext = GetDlgItem(IDC_EDIT2); // 通
过 ID 值的判断获得下一焦点控件
break;
case IDC_EDIT2:
pwndCtrlNext = GetDlgItem(IDC_EDIT3);
break;
case IDC_EDIT3:
pwndCtrlNext = GetDlgItem(IDC_EDIT4);
break;
case IDOK: // 当焦点为默认按钮时执行对话框的 On-
OK 函数
CDialog::OnOK();
break;
}
}

```

```

default:
break;
}
pwndCtrlNext ->SetFocus(); // 为下一焦点控件设置焦
点
}

```

## 4 改进方法

当按回车键时,将执行默认按钮的函数,因此通过在默认按钮的函数加入相应的代码便可达到通过回车键转移焦点,另外还可以对某一文本框输入完成后,使按回车时执行特定功能,如查找、列表更新等。示例如下:

首先为每一个文本框控件增加文本框类型的数据成员,下一步是编辑默认按钮处理函数

```

void CmyDialog::OnMyOK() // 默认按钮的处理函数
{
CWnd * pWnd = GetFocus(); // 获取当前焦点控件
if ( pWnd -> IsKindOf ( RUNTIME _ CLASS
(CEdit))) // 判断是否为文本框控件
{
if ( pWnd -> m_hWnd == m_cEditRequery. m_
hWnd)
{
DoRequery(); // 当在查询文本框按回车键时将执行
查询功能,而不执行转移焦点
}
else
::PostMessage ( this -> m_hWnd, WM_NEXTDL-
GCTL, 0, 0L); // 将焦点转移至下一控件
}
else
CDialog::OnOK();
}
}

```

## 5 总结

本文通过介绍三种常规方法及一种改进方法,说明了如何在各个文本框之间的焦点转移通过回车键实现,最后对于文本框输入完成后按回车执行某种特定功能的方法。本示例在 Windows 2000 Visual C++ 6.0 上运行通过。

### 参考文献

- 1 微软, MSDN
- 2 雷神, [VC++ 技术内幕] 学习笔记(10), 易会网。