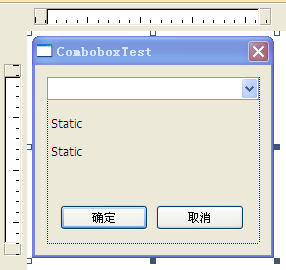
在学习ComboBoxEx控件时遇到一个非常奇怪的问题，我写了一个小程序来说明这种情况：

1、我添加一个ComboBoxEx和两个静态文本控件：



添加为控件添加变量，分别为：

CComboBoxEx m\_ComboBox;

CString m\_str1;

CString m\_str2;

2、在OnInitDialog()函数中，为ComboBoxEx控件添加三个项：

COMBOBOXEXITEM cbi;

CString strTemp[3];

strTemp[0]=\_T("Tom");

strTemp[1]=\_T("LiLei");

strTemp[2]=\_T("Lucy");

cbi.mask=CBEIF\_TEXT;

for (int i=0;i<3;i++)

{

cbi.iItem=i;

cbi.pszText=(LPTSTR)(LPCTSTR)strTemp[i];

cbi.cchTextMax=256;

m\_ComboBox.InsertItem(&cbi);

}

3、在控件的OnCbnSelchangeComboboxex1()函数中添加以下代码，当选择ComboBoxEx上的项时，会激发这个函数：

void CComboboxTestDlg::OnCbnSelchangeComboboxex1()

{

CString str;

//CString str("Dummy\_String");

COMBOBOXEXITEM cbi;

cbi.pszText=(LPTSTR)(LPCTSTR)str;

cbi.iItem=-1;

cbi.mask=CBEIF\_TEXT;

cbi.cchTextMax=256;

m\_ComboBox.GetItem(&cbi);

//m\_str1=str;

UpdateData(FALSE);

}

注意，这里我并没有为两个静态文本控件赋任何值。

4、问题出现了：运行的效果如下：



当我选择ComboBox中的任意一项的时候，两个静态文本框的值都随着改变了，可是我没有写任何一句改变字符串str1和str2的代码呀？难道str1和str2他们和str是同一个内存地址？？？

5、如果我把代码中注释部分打开，改写代码如下：

void CComboboxTestDlg::OnCbnSelchangeComboboxex1()

{

CString str("Dummy\_String");

COMBOBOXEXITEM cbi;

cbi.pszText=(LPTSTR)(LPCTSTR)str;

cbi.iItem=-1;

cbi.mask=CBEIF\_TEXT;

cbi.cchTextMax=256;

m\_ComboBox.GetItem(&cbi);

m\_str1=str;

UpdateData(FALSE);

}

这样，就可以得到正确的结果，当我任选ComboBoxEx中任何一项的时候，只有静态文本框1的值随之改变了，因为我在代码中有m\_str1=str这句，而静态文本框2的值没有改变。

**VC驿站SYC的解答：**

你这个是很明显的缓冲区溢出了。。。将类内部的成员：  
CString m\_str1;  
CString m\_str2;  
的内存给覆盖了，所以才导致这两个字符串的值就是下拉框里面的值，具体是谁给覆盖的呢？  
就是这个函数中：OnCbnSelchangeComboboxex1()  
CString str;  
COMBOBOXEXITEM cbi;  
cbi.pszText=(LPTSTR)(LPCTSTR)str;  
这样写代码的时候，str内部是个空串，也就是说 CString 不会给他分配空间，然后呢，你又调用：  
cbi.pszText=(LPTSTR)(LPCTSTR)str;  
这句话将 str 对象内部的指针赋值给 cbi.pszText 成员了，于是当调用 m\_ComboBox.GetItem(&cbi); 这句话的时候，就会将获取过来的字符串拷贝进 cbi.pszText 指针所指的空间中，于是就引起缓冲区溢出了。  
  
当你这么定义 str 对象时：CString str("Dummy\_String"); 默认初始化的时候就给 str 申请了空间，所以就不会出错了！但是也不安全，当下拉框里面的字符串的长度超过 "Dummy\_String" 的长度时，一样会引起溢出。所以，如果非要用 CString 的话，正确的做法是这样：

CString str;

COMBOBOXEXITEM cbi;

cbi.pszText=str.GetBufferSetLength(256);

cbi.iItem=-1;

cbi.mask=CBEIF\_TEXT;

cbi.cchTextMax=256;

m\_ComboBox.GetItem(&cbi);

str.ReleaseBuffer();

**问：**可不可以这样理解，m\_str1、m\_str2、str在没有为它们初始化的时候，它们的buffer都指向同一地址？然后str的地址又赋给了cbi.pszText，所以拷贝字符到cbi.pszText的时候，m\_str1,m\_str2,str的值都改变了。如果为str初始化一个值，就为它的buffer分配了内存，地址就与m\_str1,m\_str2不一样，是不是这样理解？

**答：**这个说的不恰当，不一定是指向同一个地址，有可能而已！恰好被覆盖了