

高手解答：关于RICHEDIT的两个问题 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\\_\\_E9\\_AB\\_98\\_E6\\_89\\_8B\\_E8\\_A7\\_A3\\_E7\\_c97\\_253819.htm](https://www.100test.com/kao_ti2020/253/2021_2022__E9_AB_98_E6_89_8B_E8_A7_A3_E7_c97_253819.htm)

问题1:RichEditCtrl在用DDX进行数据交换的时候会发生数据丢失问题?为什么!当我们在拖了一个控件到程序里后，通常的做发是Ctrl W，用类向导给控件关联一个变量，然后依靠DDX/DDV进行数据交换，如果我们用同样的方法来给RICHEDIT关联一个CString类型的变量就会存在一个问题，就是如果我们的数据大于了64K，数据就会丢失。通过查MSDN发现

，WM\_GETTEXT消息并没有设计在RICHEDIT的数据大于64K的时候怎样处理。而类向导生成的代码是用DDX\_Text来交换控件和CString变量的数据。恰好，DDX\_Text函数是调用GetWindowText函数，而这个函数又会发出WM\_GETTEXT消息到控件来返回控件里的数据。WM\_GETTEXT消息不能接受超过64K的数据，因此导致了RICHEDIT在数据交换的时候发生了丢失。为了解决这个问题，我们要用

到DDX\_RichText函数。添加下面两个函数到工程以下是引用

```
片段： DWORD CALLBACK ES2MemCallBack(DWORD_PTR  
dwCookie,LPBYTE pbBuff, LONG cb, LONG *pcb) {  
LPTSTRamp. value) { extern void AFXAPI  
AfxSetWindowText(HWND hWndCtrl, LPCTSTR lpszNew).  
HWND hWndCtrl = pDX->PrepareEditCtrl(nIDC). if  
(pDX->m_bSaveAndValidate) { int nLen =  
::GetWindowTextLength(hWndCtrl). LPTSTR lpszStrFill =  
value.GetBufferSetLength(nLen). EDITSTREAM es = {
```

(DWORD\_PTR) amp.es). value.ReleaseBuffer(). } else {  
AfxSetWindowText(hWndCtrl, value). } } 之后我们还需要修改工程的.clw文件，用文本方式打开.clw文件。参考里面类的格式加下面两行代码：以下是引用片段： ExtraDDXCount=1  
ExtraDDX1=7..TextOver64KB.CString..RichText.Retrieves text in excess of 64KB from RichEdit controls 如果没有采用上面的步骤，我们就需要手动修改代码，把所有的DDX\_Text改为DDX\_RichText。同时要把他们移到类向导控制代码的外面。也就是移出：以下是引用片段：//{{AFX\_DATA\_INIT(...)  
//}}AFX\_DATA\_INIT ////{{AFX\_DATA\_MAP(...)  
//}}AFX\_DATA\_MAP reference: Q280447 BUG: Text from a Rich Edit Control Is Truncated During Dialog Data Exchange (DDX) 问题2：当我们用类向导给richedit添加了EN\_SETFOCUS, EN\_KILLFOCUS的函数后却不能响应，我发现这个响应函数根本就没有被调用。即使是一个MessageBox()函数也不会调用。原来是默认的消息映射添加错误了。正确的消息映射和响应应该是：以下是引用片段：  
ON\_EN\_SETFOCUS(IDC\_RICHEDIT1,OnSetfocusRichedit1)  
ON\_EN\_KILLFOCUS(IDC\_RICHEDIT1,OnKillfocusRichedit1)  
响应函数形式为：以下是引用片段：afx\_msg void  
OnSetfocusRichedit1().afx\_msg void OnKillfocusRichedit1(). 但是如果我们用类向导来直接添加，生成的代码却是：以下是引用片段：ON\_NOTIFY(EN\_SETFOCUS, IDC\_RICHEDIT1,  
OnSetfocusRichedit1) ON\_NOTIFY(EN\_KILLFOCUS,  
IDC\_RICHEDIT1, OnKillfocusRichedit1) 我们需要自己手动改为上面的形式。还有一个问题就是RichEditCtrl有时候不会出

现在类向导的控件ID列表里。这就需要我们自己添加DDX/DDV函数。自己动手啦!^\_^以下是引用片段： --

```
sampledlg.h -- class CSampleDlg : public CDialog { public:
CSampleDlg(CWnd* pParent = NULL). // Dialog Data
//{{AFX_DATA(CSampleDlg) enum { IDD =
IDD_SAMPLE_DIALOG }. CString m_edit. // Added by
ClassWizard for an edit control //}}AFX_DATA // Manually add
member variables for the rich edit control CRichEditCtrl
m_richEditCtrl. .... sampledlg.cpp -- ..... void
CSampleDlg::DoDataExchange(CDataExchange* pDX) {
CDialog::DoDataExchange(pDX).
//{{AFX_DATA_MAP(CSampleDlg) DDX_Text(pDX,
IDC_EDIT, m_edit). DDV_MaxChars(pDX, m_edit, 10).
//}}AFX_DATA_MAP // Manually add DDX_Control, DDX_Text
and DDV_MaxChars for the // rich edit control
DDX_Control(pDX, IDC_RICHEDIT1, m_richEditCtrl).
DDX_Text(pDX, IDC_RICHEDIT1, m_richedit).
DDV_MaxChars(pDX, m_richedit, 10). } 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
```