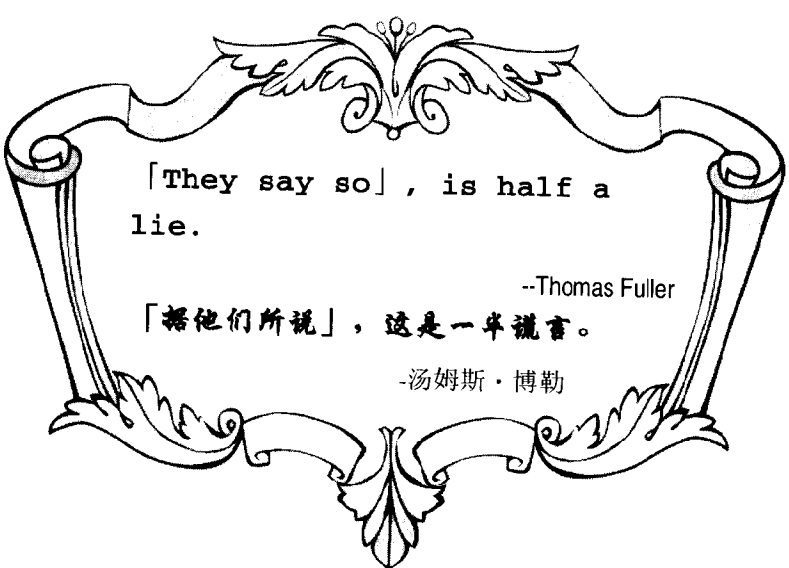


对话框应用程序与控件的应用

- 对话框应用程序的使用
- Button 与 Static Text
- Picture
- List Box、Combo Box 与多页对话框的建立
- Scroll Bar、Slider、Spin、Edit Box 与 Check Box
- Tree Control、List Control 与 Radio Box

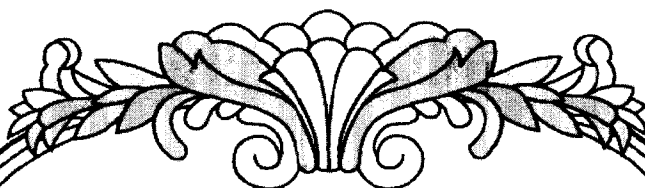


「They say so」, is half a
lie.

--Thomas Fuller

「据他们所说」，这是一半谎言。

-汤姆斯·博勒



本章导读

在这一章里，我们将介绍如何建立对话框应用程序，并且介绍各种控件的使用。而从这章起，我们将借由 AppWizard 协助建立出对话框应用程序的基本程序代码，然后再利用 ClassWizard 建立讯息响应函数，以及 DDX/DDV 机制。相信从这一章里，您一定能体会到 Visual C++ 提供的各项工具，的确是可以加快视窗程序设计的速度与品质。

25-1 对话框应用程序的使用

说实在的，建立一个 Doc/View 架构的应用程序实在是一场规模浩大，且令人望之却步的工程，当然这个复杂的架构也不是没有存在的意义，因为它着实替我们解决了不少数据处理的问题。但是并不是所有的视窗应用程序都必须被建立成具备 Doc/View 架构的应用程序，对于一些并不处理数据储存的应用程序，Doc/View 架构就有点多余。因此，我们只需要将这类应用程序建立成对话框应用程序即可。

一般来说，对话框应用程序相较于 Doc/View 架构的视窗应用程序，通常是比较小的程序，例如：CD 播放程序、Ftp 程序...等。其实对话框应用程序就是一个独立的 Modal 对话框，因为两者不论在架构上，或者是在产生的方式上，都是一样的。两者的差别只不过在于对话框应用程序是独立的一个程序，而对话框仅是某一视窗程序的一部分，仅在被呼叫时，才被产生。

25-2 Button 与 Static Text

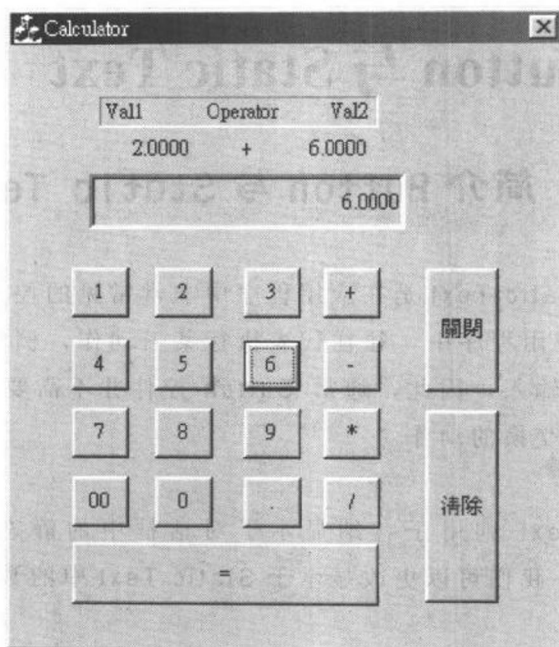
25-2-1 简介 Button 与 Static Text 控件

Button 与 Static Text 是在应用程序中非常常见的控件。一般来说，Button 控件在应用程序中，往往代表执行某一动作，例如：完成了某一对话框中数据的输入。因此，通常 Button 控件并不需要运用 DDX/DDV 机制，进行数据交换的动作。

而 Static Text 则用于容纳显示于对话框中的静态文字，而通过 DDX/DDV 机制，我们可以更改显示于 Static Text 中的数据。

25-2-2 小计算器程序范例

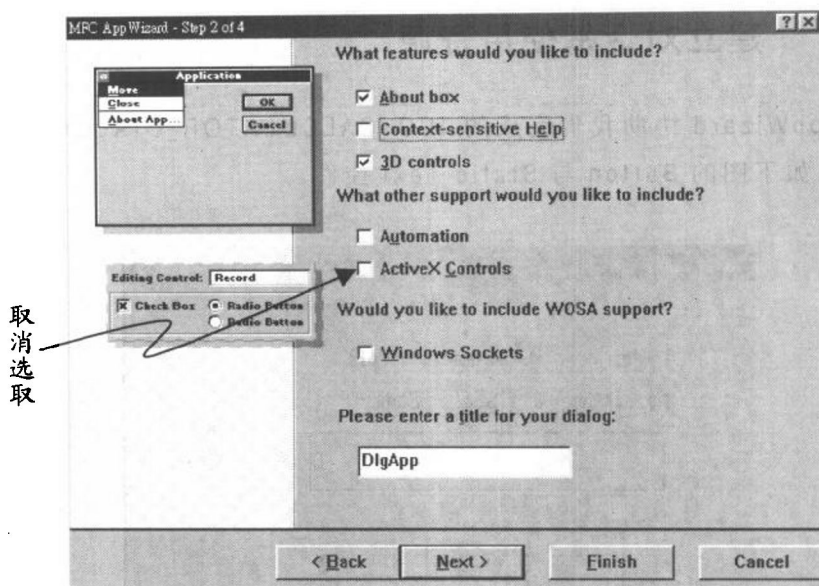
在这一节里，我们将以撰写一个简单的小计算器程序，来告诉您如何使用 Button 与 Static Text 这两个控件。下图是 Calculator 程序范例的执行结果。



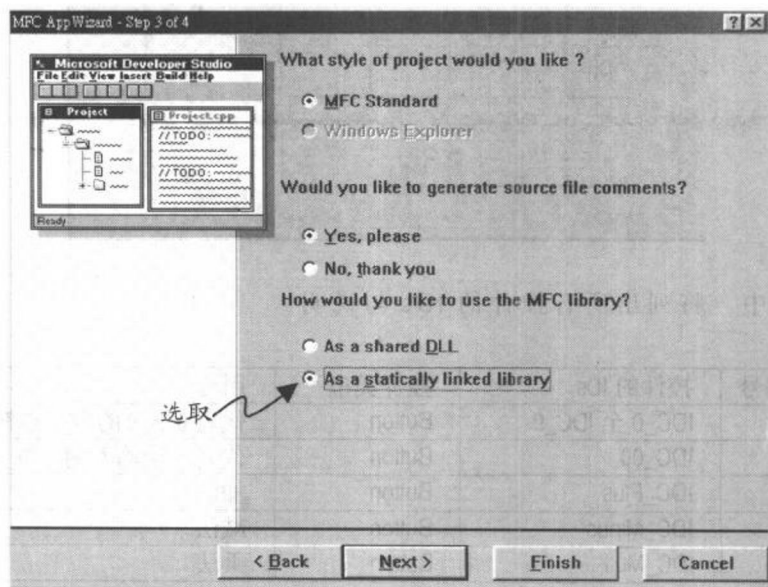
下面是建立 Calculator 程序范例的过程。

第一步 利用 AppWizard 建立对话框应用程序

请利用 AppWizard 建立一个名为 Calculator 的工程。在建立过程的 Step 2 里取消 ActiveX Controls 项的选取。

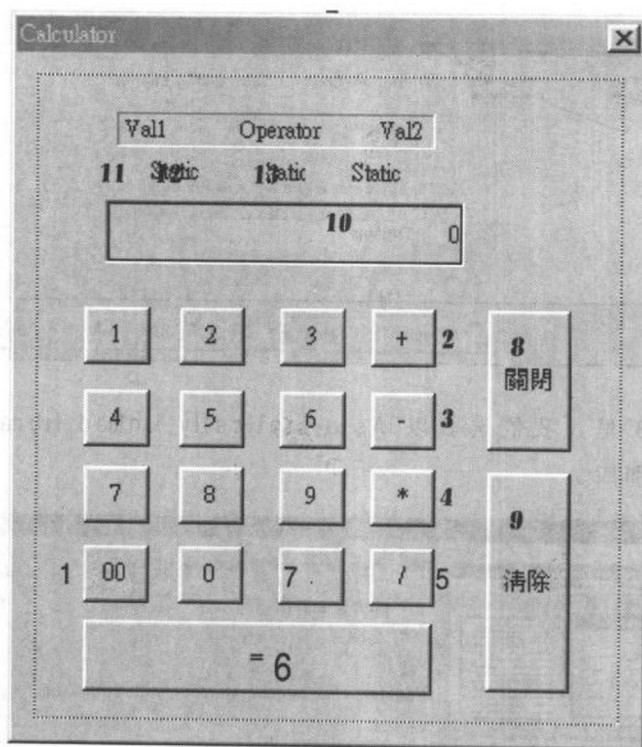


在 Step 3 里，我们选取以 As a statically linked library 的方式使用 MFC 的类别库。



第二步 建立对话框使用资源

在 AppWizard 协助我们完成的 IDD_CALCULATOR_DIALOG 对话框中，建立如下图所示的 Button 与 Static Text 控件。



下表中，将列出所有控件的 IDs 与说明。

编号	控件的 IDs	控件类型	用途
	IDC_0 至 IDC_9	Button	代表 0 至 9 的数字按钮
1	IDC_00	Button	代表一次输入两个 0
2	IDC_Plus	Button	加法
3	IDC_Minus	Button	减法
4	IDC_Mult	Button	乘法

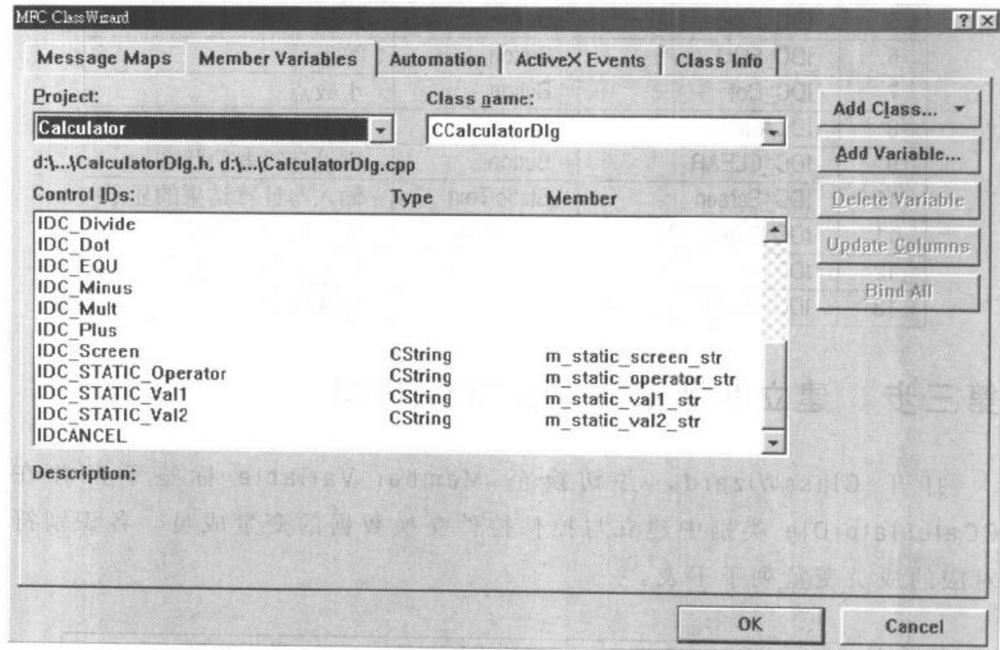
5	IDC_Divide	Button	除法
6	IDC_EQU	Button	等于
7	IDC_Dot	Button	小数点
8	IDC_CANCEL	Button	关闭计算器
9	IDC_CLEAR	Button	重设计算器的状态
10	IDC_Screen	Static Text	输入与计算结果的显示
11	IDC_STATIC_Val1	Static Text	输入的第一个数字
12	IDC_STATIC_Val2	Static Text	输入的第二个数字
13	IDC_STATIC_Operator	Static Text	欲执行的计算

第三步 建立对话框的 DDX/DDV 机制

打开 ClassWizard，并切换至 Member Variable 标签下，并在 CCalculatorDlg 类别中建立与控件交换数据的变量成员。各识别符对应的成员变量列于下表。

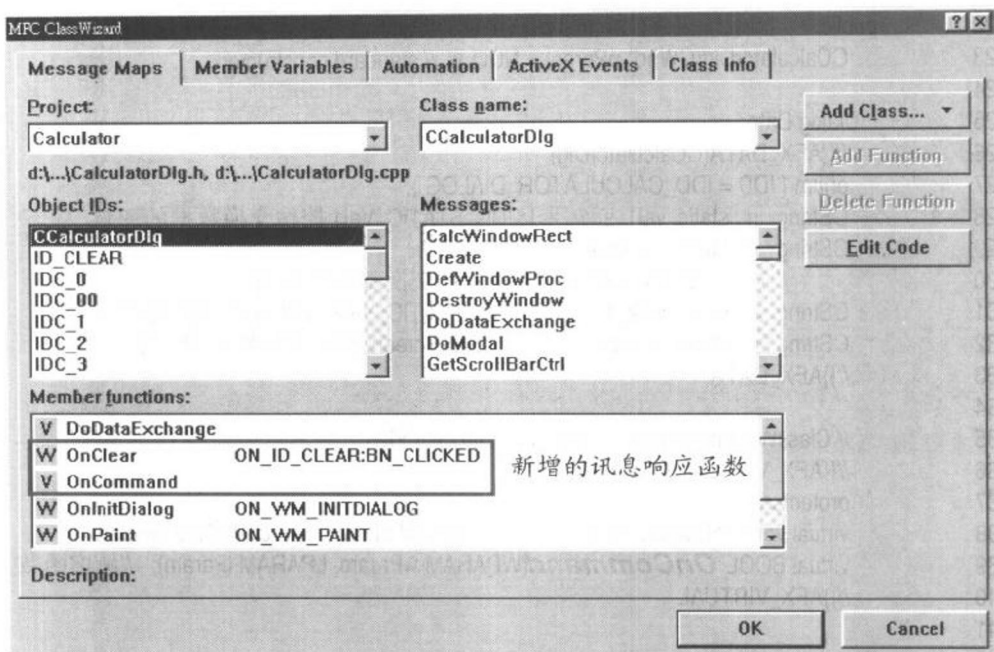
控件的 IDs	对应的成员变量	数据类型
IDC_Screen	m_static_screen_str	CString
IDC_STATIC_OPERATOR	m_static_operator_str	CString
IDC_STATIC_Val1	m_static_val1_str	CString
IDC_STATIC_Val2	m_static_val2_str	CString

完成 CCalculatorDlg 类别成员变量建立后的 Member Variables 标签如下所示。



第四步 建立讯息响应函数

将 ClassWizard 切换到 Message Maps 标签下，然后新增响应控件讯息的 OnCommand 函数，以及响应 IDC_CLEAR 按钮 BN_CLICKED 讯息的 OnClear 函数。



结束上述步骤并完成程序代码的建立后，下面是 `CControlDlg` 类别的完整程序代码，较为重要的部分，我们将以灰色底纹标明。

◆ Calculator 程序范例 - `CCalculatorDlg` 类别的标头文件

```

1      // CalculatorDlg.h: header file
2      //
3...13     ..... //由 AppWizard 产生与 CListPage 类别无关的程序代码
14     enum Operator{Divide, Mult, Minus, Plus};//用于记录计算方式的自定类型变量
15
16     //////////////////////////////////////
17     // CCalculatorDlg dialog
18
19     class CCalculatorDlg: public CDialog
20     {
21     // Construction

```

```

22     public:
23         CCalculatorDlg(CWnd* pParent = NULL); // standard constructor
24
25     // Dialog Data
26     //{AFX_DATA(CCalculatorDlg)
27     enum { IDD = IDD_CALCULATOR_DIALOG };
28     CString m_static_val1_str; //用于 IDC_STATIC_Val1 控件交换数据的变量
29     CString m_static_operator_str;
30         //用于 IDC_STATIC_Operator 控件交换数据的变量
31     CString m_static_val2_str; //用于 IDC_STATIC_Val2 控件交换数据的变量
32     CString m_static_screen_str; //用于 IDC_Screen 控件交换数据的变量
33     //}AFX_DATA
34
35     // ClassWizard generated virtual function overrides
36     //{AFX_VIRTUAL(CCalculatorDlg)
37     protected:
38     virtual void DoDataExchange(CDataExchange* pDX);    // DDX/DDV support
39     virtual BOOL OnCommand(WPARAM wParam, LPARAM lParam); //新增函数
40     //}AFX_VIRTUAL
41
42     // Implementation
43     protected:
44     HICON m_hIcon;
45
46     // Generated message map functions
47     //{AFX_MSG(CCalculatorDlg)
48     virtual BOOL OnInitDialog ();
49     afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
50     afx_msg void OnPaint ();
51     afx_msg HCURSOR OnQueryDragIcon ();
52     afx_msg void OnClear (); //新增函数
53     //}AFX_MSG
54     DECLARE_MESSAGE_MAP ()
55     private:
56     void EqualTo () ;//执行运算的函数
57     Operator m_operator;    //自定类型变量，记录欲执行的运算
58     void InpNum(int num);    //读取输入数字的函数
59     int m_after_dot;    //用于记录小数点后位数的变量
60     BOOL m_dot;    //是否处于输入小数的状态
61     double m_val1, m_val2;    //储存运算数字的变量
62     };

```

63...

◆ Calculator 程序范例 - CCalculatorDlg 类别的实做文件

```

1      // CalculatorDlg.cpp : implementation file
2      //
3
4      #include "stdafx.h"
5      #include "Calculator.h"
6      #include "CalculatorDlg.h"
7      #include <math.h> //定义 pow 函数的标头文件, 此行以手动方式增加
8...59      ..... // CAboutDlg 类别的定义与 CCalculatorDlg 类别无关
60      //////////////////////////////////////
61      // CCalculatorDlg dialog
62      //建构符
63      CCalculatorDlg::CCalculatorDlg(CWnd* pParent /*=NULL*/)
64          : CDialog(CCalculatorDlg::IDD, pParent)
65      {
66          //起始值设定
67          m_val1 = m_val2 = m_after_dot = 0;
68          m_dot = FALSE;
69          m_operator = Plus; //预设执行的运算为加法
70
71          //{AFX_DATA_INIT(CCalculatorDlg)
72          m_static_val1_str = _T("");
73          m_static_operator_str = _T("");
74          m_static_val2_str = _T("");
75          m_static_screen_str = _T("");
76          //}AFX_DATA_INIT
77          // Note that LoadIcon does not require a subsequent DestroyIcon in Win32
78          m_hIcon = AfxGetApp() ->LoadIcon(IDR_MAINFRAME);
79      }
80
81      void CCalculatorDlg::DoDataExchange(CDataExchange* pDX)
82      {
83          CDialog::DoDataExchange(pDX);
84          //数据交换机制
85          //{AFX_DATA_MAP(CCalculatorDlg)

```

```

86 DDX_Text(pDX, IDC_STATIC_Val1, m_static_val1_str);
87 DDX_Text(pDX, IDC_STATIC_Operator, m_static_operator_str);
88 DDX_Text(pDX, IDC_STATIC_Val2, m_static_val2_str);
89 DDX_Text(pDX, IDC_Screen, m_static_screen_str);
90 //}}AFX_DATA_MAP
91 }
92
93 BEGIN_MESSAGE_MAP(CCalculatorDlg, CDialog)
94 //{{AFX_MSG_MAP(CCalculatorDlg)
95 ON_WM_SYSCOMMAND ()
96 ON_WM_PAINT ()
97 ON_WM_QUERYDRAGICON ()
98 ON_BN_CLICKED(IDC_CLEAR, OnClear) //新增的响应讯息
99 //}}AFX_MSG_MAP
100 END_MESSAGE_MAP ()
101
102...184 ..... //由 AppWizard 建立的响应函数, 您可略过
185 //响应各按钮被按下后, 所发出的讯息
186 BOOL CCalculatorDlg::OnCommand(WPARAM wParam, LPARAM lParam)
187 {
188 // TODO: Add your specialized code here and/or call the base class
189 switch(wParam) //利用传入的 wParam 参数, 判断发出讯息的按钮
190 { //将按钮代表之数字传入 InpNum 函数计算使用者欲输入的数字
191 case IDC_0:
192 InpNum(0);
193 break;
194 case IDC_1:
195 InpNum(1);
196 break;
197 case IDC_2:
198 InpNum(2);
199 break;
200 case IDC_3:
201 InpNum(3);
202 break;
203 case IDC_4:
204 InpNum(4);
205 break;
206 case IDC_5:
207 InpNum(5);
208
209 break;
209 case IDC_6:

```

```

210     InpNum(6);
211     break;
212     case IDC_7:
213         InpNum(7);
214         break;
215     case IDC_8:
216         InpNum(8);
217         break;
218     case IDC_9:
219         InpNum(9);
220         break;
221     case IDC_00: //一次输入两个零
222         if(m_dot == FALSE) //判断是否处于输入小数的状态
223             m_val2 = m_val2 * 100; //已输入的数字乘以 100
224         else
225             m_after_dot -= 2; //设定将输入数字的小数点位数
226
227         m_static_screen_str.Format("%10.4f", m_val2); //重新显示已输入数字
228         break;
229     case IDC_Dot:
230         m_dot = TRUE; //设定为将输入小数的状态
231         m_static_screen_str.Format("%10.4f", m_val2); //重新显示已输入数字
232         break;
233     case IDC_Divide:
234         EqualTo (); //计算已输入的算式
235         m_val2 = m_after_dot = 0; //重设 m_val2 与 m_after_dot
236         m_dot = FALSE; //设定为非小数输入状态
237         m_operator = Divide; //设定运算方式为除法
238         m_static_operator_str.Format("/");
239         //更新 IDC_STATIC_OPERATOR 显示的运算符号
240         break;
241     case IDC_Plus:
242         EqualTo (); //计算已输入的算式
243         m_val2 = m_after_dot = 0; //重设 m_val2 与 m_after_dot
244         m_dot = FALSE; //设定为非小数输入状态
245         m_operator = Plus; //设定运算方式为加法
246         m_static_operator_str.Format("+");
247         //更新 IDC_STATIC_OPERATOR 显示的运算符号
248         break;
249     case IDC_Minus:

```

```

250         EqualTo ( ) ; //计算已输入的算式
251         m_val2 = m_after_dot = 0;    //重置 m_val2 与 m_after_dot
252         m_dot = FALSE;    //设定为非小数输入状态
253         m_operator = Minus; //设定运算方式为减法
254         m_static_operator_str.Format("-");
255         //更新 IDC_STATIC_OPERATOR 显示的运算符号
256         break;
257     case IDC_Mult:
258         EqualTo ( ) ; //计算已输入的算式
259         m_val2 = m_after_dot = 0;    //重置 m_val2 与 m_after_dot
260         m_dot = FALSE;    //设定为非小数输入状态
261         m_operator = Mult; //设定运算方式为乘法
262         m_static_operator_str.Format("*");
263         //更新 IDC_STATIC_OPERATOR 显示的运算符号
264         break;
265     case IDC_EQU:
266         EqualTo ( ) ; //计算已输入的算式
267         m_val2 = 0;
268         m_static_operator_str = "";
269         break;
270     }
271
272     m_static_val1_str.Format("%10.4f", m_val1);
273     //更新 IDC_STATIC_Val1 标签的显示数字
274     m_static_val2_str.Format("%10.4f", m_val2);
275     //更新 IDC_STATIC_Val2 标签的显示数字
276     UpdateData(FALSE);    //将数据更新至对话框
277
278     return CDialog::OnCommand(wParam, lParam);
279 }
280
281 //计算使用者输入的数字
282 void CCalculatorDlg::InpNum(int num)
283 {
284     if(m_dot == TRUE && m_after_dot > -5)
285     { //判断是否为小数输入状态, 且输入的小数不可为小数点 4 位以后
286         m_val2 += num * pow(10, --m_after_dot);    //计算输入的数字
287     }
288     else if(m_dot == FALSE) //判断是否为非小数输入状态
289         m_val2 = m_val2 * 10 + num; //计算输入的数字

```

```

290
291     m_static_screen_str.Format("%10.4f", m_val2);
292     //更新 IDC_Screen 标签所显示的数据
293     UpdateData(FALSE);    //将数据更新至对话框
294 }
295 //执行计算动作
296 void CCalculatorDlg::EqualTo ( )
297 {
298     switch(m_operator)//判断欲执行的运算
299     {
300     case Divide://除法运算
301         m_val1 /= m_val2;
302         break;
303     case Minus://减法运算
304         m_val1 -= m_val2;
305         break;
306     case Mult: //乘法运算
307         m_val1 *= m_val2;
308         break;
309     case Plus: //加法运算
310         m_val1 += m_val2;
311         break;
312     }
313     m_static_screen_str.Format("%10.4f", m_val1);
314     //更新 IDC_Screen 标签所显示的数据
315 }
316 //重新起始设定程序的数据与状态
317 void CCalculatorDlg::OnClear ( )
318 {
319     // TODO: Add your control notification handler code here
320     //重新起始设定所有数据与状态
321     m_val1 = m_val2 = m_after_dot = 0;
322     m_operator = Plus;
323     m_static_val1_str.Format("%d",0);
324     m_static_val2_str.Format("%d",0);
325     m_static_screen_str.Format("%d",0);
326     m_static_operator_str.Format("");
327     m_dot = FALSE;
328     UpdateData(FALSE);    //将数据更新至对话框
329 }

```