

```

1  #define UNICODE
2  #include<windows.h>
3  #include<process.h>
4  #include"AutomationServerWithRegFile.h"
5  // global function declarations
6  LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
7  // global variable declarations
8  ISum *pISum=NULL;
9  // WinMain
10 int WINAPI WinMain(HINSTANCE hInstance,HINSTANCE hPrevInstance,
11                    LPSTR lpCmdLine,int nCmdShow)
12 {
13     // variable declarations
14     WNDCLASSEX wndclass;
15     HWND hwnd;
16     MSG msg;
17     TCHAR AppName[]=TEXT("ComClient");
18     HRESULT hr;
19     // code
20     // COM Initialization
21     hr=CoInitialize(NULL);
22     if(FAILED(hr))
23     {
24         MessageBox(NULL,TEXT("COM Library Can Not Be Initialized.\nProgram Will
25             Now Exit."),TEXT("Program Error"),MB_OK);
26         exit(0);
27     }
28     // WNDCLASSEX initialization
29     wndclass.cbSize=sizeof(wndclass);
30     wndclass.style=CS_HREDRAW|CS_VREDRAW;
31     wndclass.cbClsExtra=0;
32     wndclass.cbWndExtra=0;
33     wndclass.lpfnWndProc=WndProc;
34     wndclass.hIcon=LoadIcon(NULL,IDI_APPLICATION);
35     wndclass.hCursor=LoadCursor(NULL,IDC_ARROW);
36     wndclass.hbrBackground=(HBRUSH)GetStockObject(WHITE_BRUSH);
37     wndclass.hInstance=hInstance;
38     wndclass.lpszClassName=AppName;
39     wndclass.lpszMenuName=NULL;
40     wndclass.hIconSm=LoadIcon(NULL,IDI_APPLICATION);
41     // register window class
42     RegisterClassEx(&wndclass);
43     // create window
44     hwnd=CreateWindow(AppName,
45                     TEXT("Client Of COM Dll Server"),
46                     WS_OVERLAPPEDWINDOW,
47                     CW_USEDEFAULT,
48                     CW_USEDEFAULT,
49                     CW_USEDEFAULT,
50                     CW_USEDEFAULT,
51                     NULL,

```

```

52         hInstance,
53         NULL);
54     ShowWindow(hwnd,nCmdShow);
55     UpdateWindow(hwnd);
56     // message loop
57     while(GetMessage(&msg,NULL,0,0))
58     {
59         TranslateMessage(&msg);
60         DispatchMessage(&msg);
61     }
62     // COM Un-initialization
63     CoUninitialize();
64     return((int)msg.wParam);
65 }
66 // Window Procedure
67 LRESULT CALLBACK WndProc(HWND hwnd,UINT iMsg,WPARAM wParam,LPARAM lParam)
68 {
69     // function declarations
70     void SafeInterfaceRelease(void);
71     // variable declarations
72     HRESULT hr;
73     int iNum1,iNum2;
74     // code
75     switch(iMsg)
76     {
77     case WM_CREATE:
78         hr=CoCreateInstance(CLSID_SumAutomation,NULL,CLSCTX_LOCAL_SERVER,
79                             IID_ISum,(void **)&pISum);
80         if(FAILED(hr))
81         {
82             MessageBox(hwnd,TEXT("ISum Interface Can Not Be Obtained"),TEXT
83                         ("Error"),MB_OK|MB_TOPMOST);
84             DestroyWindow(hwnd);
85         }
86         // initialize arguments hardcoded
87         iNum1=155;
88         iNum2=145;
89         // call SumOfTwoIntegers() of ISum to get the sum
90         pISum->SumOfTwoIntegers(iNum1,iNum2);
91         pISum->Release();
92         pISum=NULL;// make released interface NULL
93         // exit the application
94         DestroyWindow(hwnd);
95         break;
96     case WM_DESTROY:
97         SafeInterfaceRelease();
98         PostQuitMessage(0);
99         break;
100     }
101     return(DefWindowProc(hwnd,iMsg,wParam,lParam));
102 }
103 void SafeInterfaceRelease(void)

```

```
103 {  
104     // code  
105     if(pISum)  
106     {  
107         pISum->Release();  
108         pISum=NULL;  
109     }  
110 }  
111
```