

```
1 class ISum:public IUnknown
2 {
3 public:
4     // ISum specific method declarations
5     virtual HRESULT __stdcall SumOfTwoIntegers(int,int,int *)=0;// pure virtual
6 };
7 class ISubtract:public IUnknown
8 {
9 public:
10     // ISubtract specific method declarations
11     virtual HRESULT __stdcall SubtractionOfTwoIntegers(int,int,int *)=0;// pure virtual
12 };
13 // CLSID of SumSubtract Component {DFB1278D-20D3-4388-A83D-B2F2DE4EF59B}
14 const CLSID CLSID_SumSubtract=
15     {0xdfb1278d,0x20d3,0x4388,0xa8,0x3d,0xb2,0xf2,0xde,0x4e,0xf5,0x9b};
16 // IID of ISum Interface
17 const IID IID_ISum=
18     {0x791876b8,0x4bd,0x4202,0x91,0x8d,0xc2,0x66,0x30,0x96,0xfe,0xbf};
19 // IID of ISubtract Interface
20 const IID IID_ISubtract=
21     {0x9f2a8316,0x4eda,0x4113,0xac,0x4c,0x64,0x52,0x24,0x18,0x78,0x47};
```

```
REGEDIT4
[HKEY_CLASSES_ROOT\CLSID\{DFB1278D-20D3-4388-A83D-B2F2DE4EF59B}]
@="MyComDll"
[HKEY_CLASSES_ROOT\CLSID\{DFB1278D-20D3-4388-A83D-
B2F2DE4EF59B}\InprocServer32]
@="E:\\WINNT\\system32\\ClassFactoryDllServerWithRegFile.dll"
```

```

1  #define UNICODE
2  #include<windows.h>
3  #include"ClassFactoryDllServerWithRegFile.h"
4  // global function declarations
5  LRESULT CALLBACK WndProc(HWND,UINT,WPARAM,LPARAM);
6  // global variable declarations
7  ISum *pISum=NULL;
8  ISubtract *pISubtract=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,iSum;
74     TCHAR str[255];
75     // code
76     switch(iMsg)
77     {
78     case WM_CREATE:
79         hr=CoCreateInstance(CLSID_SumSubtract,NULL,CLSCTX_INPROC_SERVER,
80             IID_ISum,(void **)&pISum);
81         if(FAILED(hr))
82         {
83             MessageBox(hwnd,TEXT("ISum Interface Can Not Be Obtained"),TEXT
84                 ("Error"),MB_OK);
85             DestroyWindow(hwnd);
86         }
87         // initialize arguments hardcoded
88         iNum1=55;
89         iNum2=45;
90         // call SumOfTwoIntegers() of ISum to get the sum
91         pISum->SumOfTwoIntegers(iNum1,iNum2,&iSum);
92         // display the result
93         wsprintf(str,TEXT("Sum Of %d And %d = %d"),iNum1,iNum2,iSum);
94         MessageBox(hwnd,str,TEXT("Result"),MB_OK);
95         // call QueryInterface() on ISum,to get ISubtract's pointer
96         hr=pISum->QueryInterface(IID_ISubtract,(void **)&pISubtract);
97         if(FAILED(hr))
98         {
99             MessageBox(hwnd,TEXT("ISubtract Interface Can Not Be Obtained"),TEXT
100                 ("Error"),MB_OK);
101             DestroyWindow(hwnd);
102         }
103         // as ISum is now not needed onwords, release it

```

```
102     pISum->Release();
103     pISum=NULL;// make released interface NULL
104     // again initialize arguments hardcoded
105     iNum1=155;
106     iNum2=145;
107     // again call SumOfTwoIntegers() of ISum to get the new sum
108     pISubtract->SubtractionOfTwoIntegers(iNum1,iNum2,&iSum);
109     // as ISum is now not needed onwords, release it
110     pISubtract->Release();
111     pISubtract=NULL;// make released interface NULL
112     // display the result
113     wsprintf(str,TEXT("Subtraction Of %d And %d = %d"),iNum1,iNum2,iSum);
114     MessageBox(hwnd,str,TEXT("Result"),MB_OK);
115     // exit the application
116     DestroyWindow(hwnd);
117     break;
118 case WM_DESTROY:
119     PostQuitMessage(0);
120     break;
121 }
122 return(DefWindowProc(hwnd,iMsg,wParam,lParam));
123 }
124 void SafeInterfaceRelease(void)
125 {
126     // code
127     if(pISum)
128     {
129         pISum->Release();
130         pISum=NULL;
131     }
132     if(pISubtract)
133     {
134         pISubtract->Release();
135         pISubtract=NULL;
136     }
137 }
138
```



```
1  #define UNICODE
2  #include<windows.h>
3  #include"ClassFactoryDllServerWithRegFile.h"
4  // class declarations
5  class CSumSubtract:public ISum,ISubtract
6  {
7  private:
8      long m_cRef;
9  public:
10     // constructor method declarations
11     CSumSubtract(void);
12     // destructor method declarations
13     ~CSumSubtract(void);
14     // IUnknown specific method declarations (inherited)
15     HRESULT __stdcall QueryInterface(REFIID,void **);
16     ULONG __stdcall AddRef(void);
17     ULONG __stdcall Release(void);
18     // ISum specific method declarations (inherited)
19     HRESULT __stdcall SumOfTwoIntegers(int,int,int *);
20     // ISubtract specific method declarations (inherited)
21     HRESULT __stdcall SubtractionOfTwoIntegers(int,int,int *);
22 };
23 class CSumSubtractClassFactory:public IClassFactory
24 {
25 private:
26     long m_cRef;
27 public:
28     // constructor method declarations
29     CSumSubtractClassFactory(void);
30     // destructor method declarations
31     ~CSumSubtractClassFactory(void);
32     // IUnknown specific method declarations (inherited)
33     HRESULT __stdcall QueryInterface(REFIID,void **);
34     ULONG __stdcall AddRef(void);
35     ULONG __stdcall Release(void);
36     // IClassFactory specific method declarations (inherited)
37     HRESULT __stdcall CreateInstance(IUnknown *,REFIID,void **);
38     HRESULT __stdcall LockServer(BOOL);
39 };
40 // global variable declarations
41 long g1NumberOfActiveComponents=0;// number of active components
42 long g1NumberOfServerLocks=0;// number of locks on this dll
43 // DllMain
44 BOOL WINAPI DllMain(HINSTANCE hDll,DWORD dwReason,LPVOID Reserved)
45 {
46     // code
47     switch(dwReason)
48     {
49     case DLL_PROCESS_ATTACH:
50         break;
51     case DLL_PROCESS_DETACH:
52         break;
```

```

53     }
54     return(TRUE);
55 }
56 // Implementation Of CSumSubtract's Constructor Method
57 CSumSubtract::CSumSubtract(void)
58 {
59     // code
60     m_cRef=1;// hardcoded initialization to anticipate possible failure of
        QueryInterface()
61     InterlockedIncrement(&g1NumberOfActiveComponents);// increment global counter
62 }
63 // Implementation Of CSumSubtract's Destructor Method
64 CSumSubtract::~CSumSubtract(void)
65 {
66     // code
67     InterlockedDecrement(&g1NumberOfActiveComponents);// decrement global counter
68 }
69 // Implementation Of CSumSubtract's IUnknown's Methods
70 HRESULT CSumSubtract::QueryInterface(REFIID riid,void **ppv)
71 {
72     // code
73     if(riid==IID_IUnknown)
74         *ppv=static_cast<ISum *>(this);
75     else if(riid==IID_ISum)
76         *ppv=static_cast<ISum *>(this);
77     else if(riid==IID_ISubtract)
78         *ppv=static_cast<ISubtract *>(this);
79     else
80     {
81         *ppv=NULL;
82         return(E_NOINTERFACE);
83     }
84     reinterpret_cast<IUnknown *>(*ppv)->AddRef();
85     return(S_OK);
86 }
87 ULONG CSumSubtract::AddRef(void)
88 {
89     // code
90     InterlockedIncrement(&m_cRef);
91     return(m_cRef);
92 }
93 ULONG CSumSubtract::Release(void)
94 {
95     // code
96     InterlockedDecrement(&m_cRef);
97     if(m_cRef==0)
98     {
99         delete(this);
100         return(0);
101     }
102     return(m_cRef);
103 }

```

```
104 // Implementation Of ISum's Methods
105 HRESULT CSumSubtract::SumOfTwoIntegers(int num1,int num2,int *pSum)
106 {
107     // code
108     *pSum=num1+num2;
109     return(S_OK);
110 }
111 // Implementation Of ISubtract's Methods
112 HRESULT CSumSubtract::SubtractionOfTwoIntegers(int num1,int num2,int *pSubtract)
113 {
114     // code
115     *pSubtract=num1-num2;
116     return(S_OK);
117 }
118 // Implementation Of CSumSubtractClassFactory's Constructor Method
119 CSumSubtractClassFactory::CSumSubtractClassFactory(void)
120 {
121     // code
122     m_cRef=1;// hardcoded initialization to anticipate possible failure of
123     QueryInterface()
124 }
125 // Implementation Of CSumSubtractClassFactory's Destructor Method
126 CSumSubtractClassFactory::~CSumSubtractClassFactory(void)
127 {
128     // code
129 }
130 // Implementation Of CSumSubtractClassFactory's IClassFactory's IUnknown's
131 // Methods
132 HRESULT CSumSubtractClassFactory::QueryInterface(REFIID riid,void **ppv)
133 {
134     // code
135     if(riid==IID_IUnknown)
136         *ppv=static_cast<IClassFactory *>(this);
137     else if(riid==IID_IClassFactory)
138         *ppv=static_cast<IClassFactory *>(this);
139     else
140     {
141         *ppv=NULL;
142         return(E_NOINTERFACE);
143     }
144     reinterpret_cast<IUnknown *>(*ppv)->AddRef();
145     return(S_OK);
146 }
147 ULONG CSumSubtractClassFactory::AddRef(void)
148 {
149     // code
150     InterlockedIncrement(&m_cRef);
151     return(m_cRef);
152 }
153 ULONG CSumSubtractClassFactory::Release(void)
154 {
155     // code
```



```

154     InterlockedDecrement(&m_cRef);
155     if(m_cRef==0)
156     {
157         delete(this);
158         return(0);
159     }
160     return(m_cRef);
161 }
162 // Implementation Of CSumSubtractClassFactory's IClassFactory's Methods
163 HRESULT CSumSubtractClassFactory::CreateInstance(IUnknown *pUnkOuter,REFIID riid,void **ppv)
164 {
165     // variable declarations
166     CSumSubtract *pCSumSubtract=NULL;
167     HRESULT hr;
168     // code
169     if(pUnkOuter!=NULL)
170         return(CLASS_E_NOAGGREGATION);
171     // create the instance of component i.e. of CSumSubtract class
172     pCSumSubtract=new CSumSubtract;
173     if(pCSumSubtract==NULL)
174         return(E_OUTOFMEMORY);
175     // get the requested interface
176     hr=pCSumSubtract->QueryInterface(riid,ppv);
177     pCSumSubtract->Release();// anticipate possible failure of QueryInterface()
178     return(hr);
179 }
180 HRESULT CSumSubtractClassFactory::LockServer(BOOL fLock)
181 {
182     // code
183     if(fLock)
184         InterlockedIncrement(&glNumberOfServerLocks);
185     else
186         InterlockedDecrement(&glNumberOfServerLocks);
187     return(S_OK);
188 }
189 // Implementation Of Exported Functions From This Dll
190 HRESULT __stdcall DllGetClassObject(REFCLSID rclsid,REFIID riid,void **ppv)
191 {
192     // variable declaraions
193     CSumSubtractClassFactory *pCSumSubtractClassFactory=NULL;
194     HRESULT hr;
195     // code
196     if(rclsid!=CLSID_SumSubtract)
197         return(CLASS_E_CLASSNOTAVAILABLE);
198     // create class factory
199     pCSumSubtractClassFactory=new CSumSubtractClassFactory;
200     if(pCSumSubtractClassFactory==NULL)
201         return(E_OUTOFMEMORY);
202     hr=pCSumSubtractClassFactory->QueryInterface(riid,ppv);
203     pCSumSubtractClassFactory->Release();// anticipate possible failure of
    QueryInterface()

```

```
204     return(hr);
205 }
206 HRESULT __stdcall DllCanUnloadNow(void)
207 {
208     // code
209     if((glNumberOfActiveComponents==0) && (glNumberOfServerLocks==0))
210         return(S_OK);
211     else
212         return(S_FALSE);
213 }
214
```

```
...DllServerWithRegFile\ClassFactoryDllServerWithRegFile.def 1
1 LIBRARY ClassFactoryDllServerWithRegFile
2 EXPORTS
3     DllGetClassObject    @100 PRIVATE
4     DllCanUnloadNow      @101 PRIVATE
5
```

```
1 class ISum:public IUnknown
2 {
3 public:
4     // ISum specific method declarations
5     virtual HRESULT __stdcall SumOfTwoIntegers(int,int,int *)=0;// pure virtual
6 };
7 class ISubtract:public IUnknown
8 {
9 public:
10     // ISubtract specific method declarations
11     virtual HRESULT __stdcall SubtractionOfTwoIntegers(int,int,int *)=0;// pure virtual
12 };
13 // CLSID of SumSubtract Component {DFB1278D-20D3-4388-A83D-B2F2DE4EF59B}
14 const CLSID CLSID_SumSubtract=
15     {0xdfb1278d,0x20d3,0x4388,0xa8,0x3d,0xb2,0xf2,0xde,0x4e,0xf5,0x9b};
16 // IID of ISum Interface
17 const IID IID_ISum=
18     {0x791876b8,0x4bd,0x4202,0x91,0x8d,0xc2,0x66,0x30,0x96,0xfe,0xbf};
19 // IID of ISubtract Interface
20 const IID IID_ISubtract=
21     {0x9f2a8316,0x4eda,0x4113,0xac,0x4c,0x64,0x52,0x24,0x18,0x78,0x47};
```

```
REGEDIT4
[HKEY_CLASSES_ROOT\CLSID\{DFB1278D-20D3-4388-A83D-B2F2DE4EF59B}]
@="MyComDll"
[HKEY_CLASSES_ROOT\CLSID\{DFB1278D-20D3-4388-A83D-
B2F2DE4EF59B}\InprocServer32]
@="E:\\WINNT\\system32\\ClassFactoryDllServerWithRegFile.dll"
```