After you built CustomCommand DLL do the following:

1. Open command prompt and create .il file “ildasm CustomCommand.dll /OUT:CustomCommand.il”.
2. Mark initApp and uninitApp methods as export. Find

.method public hidebysig static void initApp() cil managed

{

// Code size

And add “.export [1] as CustomInit” so you get

.method public hidebysig static void initApp() cil managed

{

.export [1] as CustomInit

// Code size

Do the same for uninit so you get

.method public hidebysig static void uninitApp() cil managed

{

.export [2] as CustomUnInit

// Code size

1. Save .il file and build DLL again “ILasm CustomCommand.il /DLL /out:CustomCommand.dll”.
2. Now DLL is ready to be used in a C++ application.

To call a custom C# command from your C++ application add the following:

typedef void (\*InitFn)();

// Load newly created library

HINSTANCE hinstLib = LoadLibrary(L"CustomCommand.dll");

// get init function

InitFn func = 0;

func = (InitFn)GetProcAddress(hinstLib, "CustomInit");

// call init function – i.e. add new custom command to command stack

func();

// get command context

OdDbCommandContextPtr pCmdCtx = GetCommandContext();

// get command stack

OdEdCommandStackPtr pCommands = ::odedRegCmds();

// execute custom command

pCommands->executeCommand("custom", pCmdCtx);