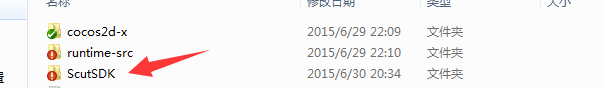
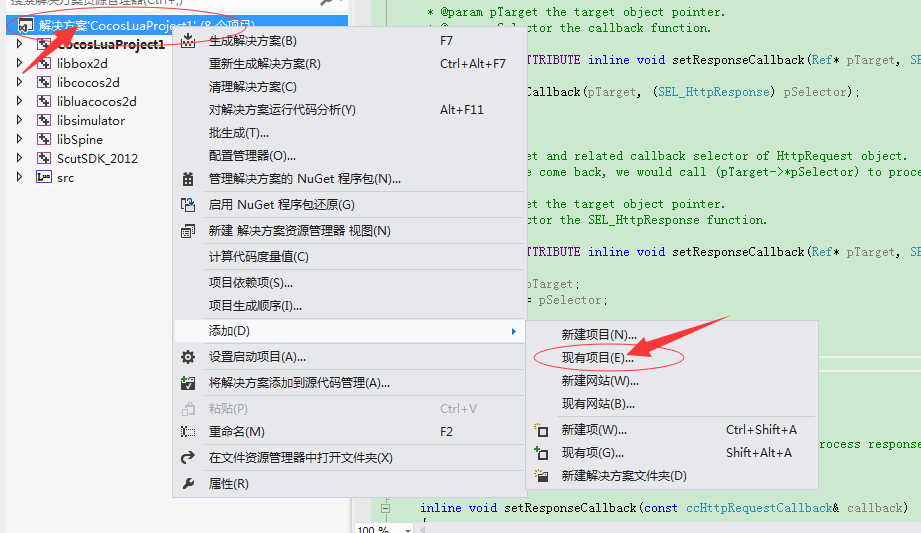
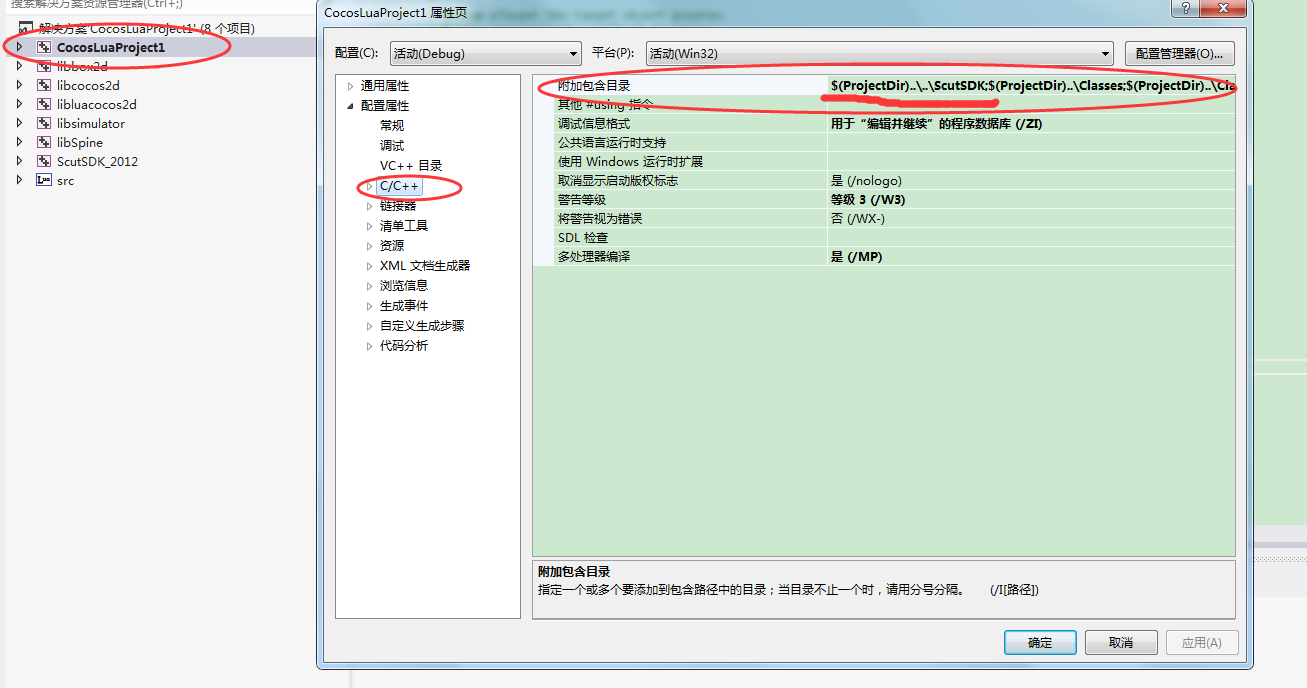
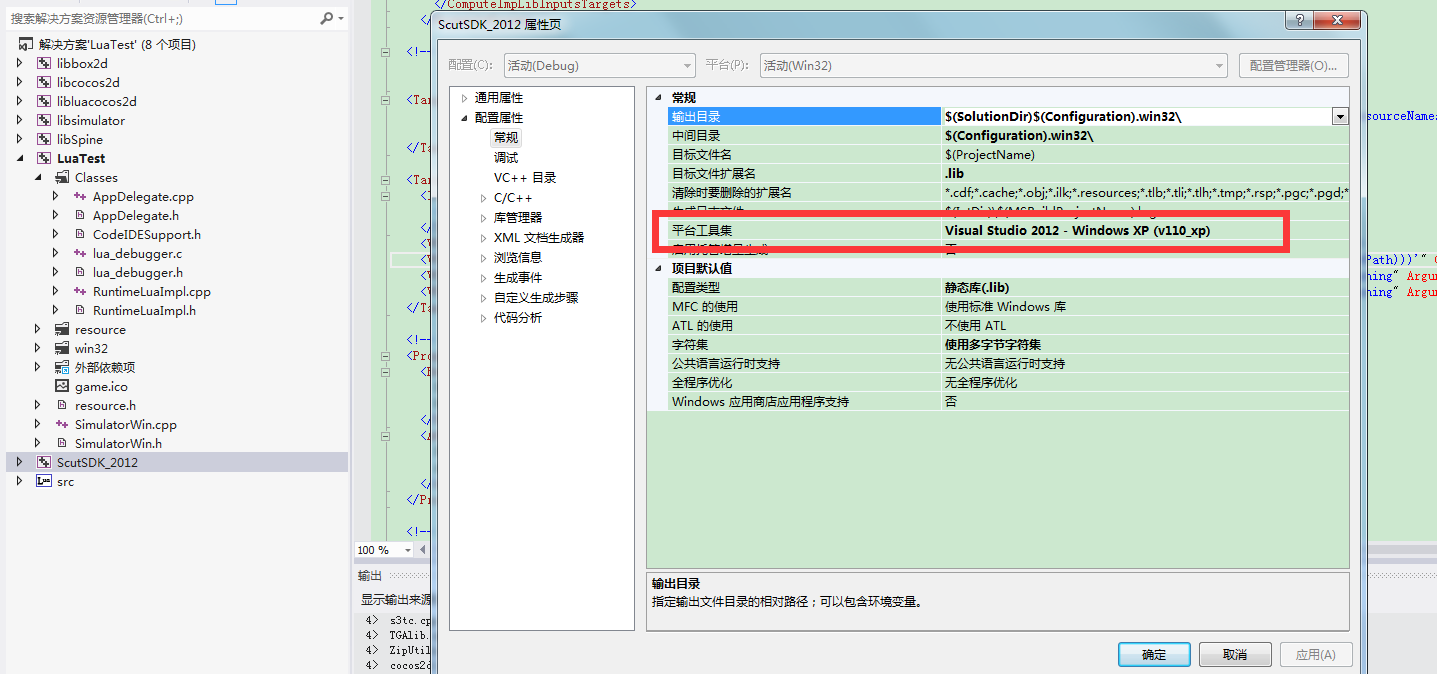
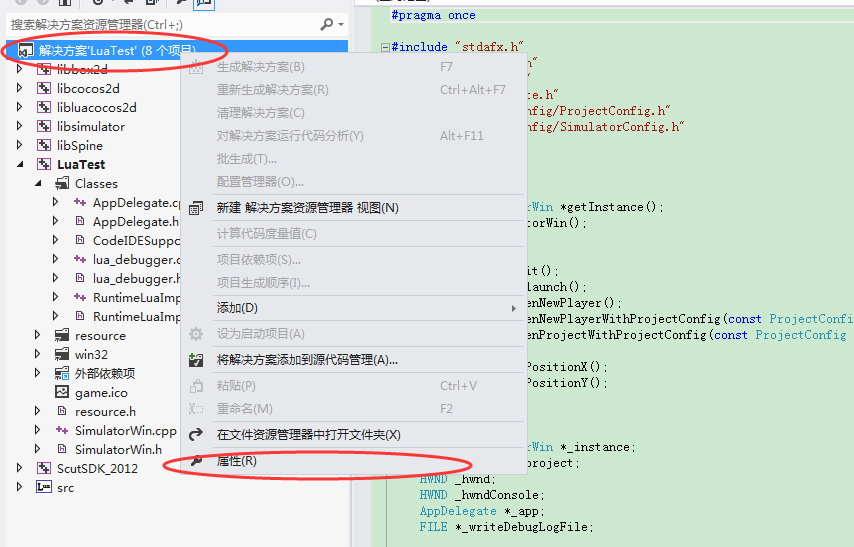
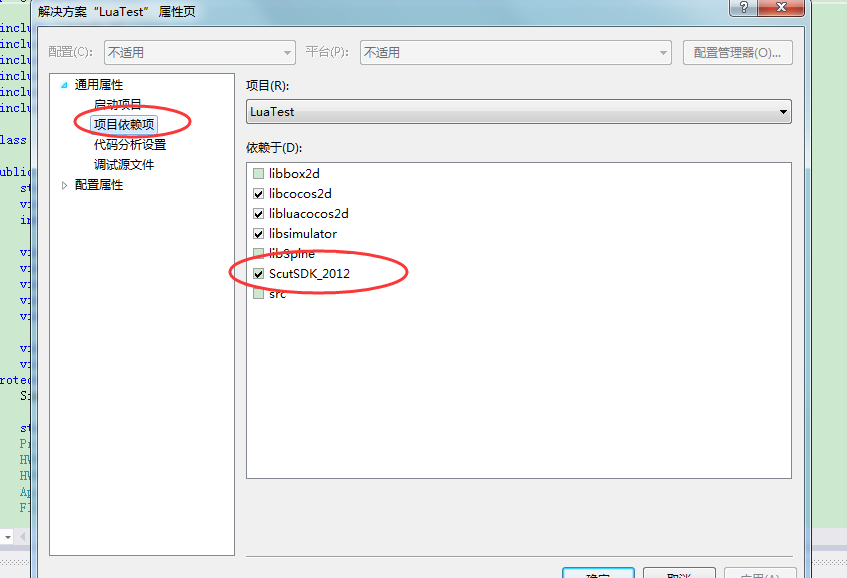
1. 拷贝scutsdk文件夹到frameworks目录
2. 打开工程项目，添加scut项目到我们项目中
3. 添加完成之后，在我们项目中添加scutext的引用目录$(ProjectDir)..\..\ScutSDK;



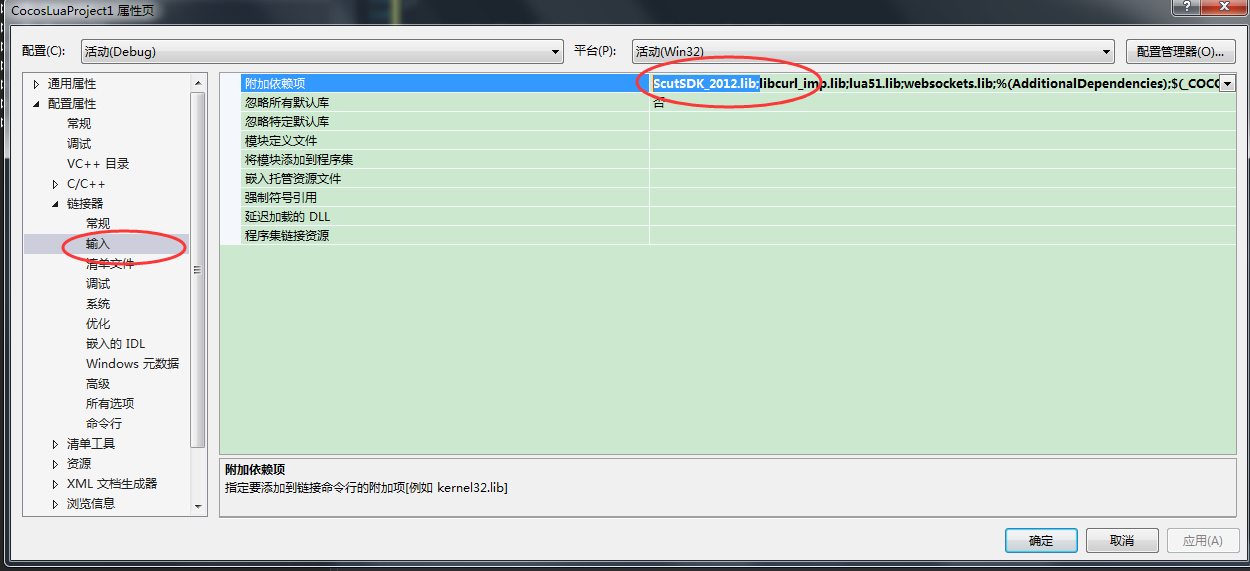
4. 修改scut项目的平台工具集和我们项目的平台工具集一致

5．设置依赖关系



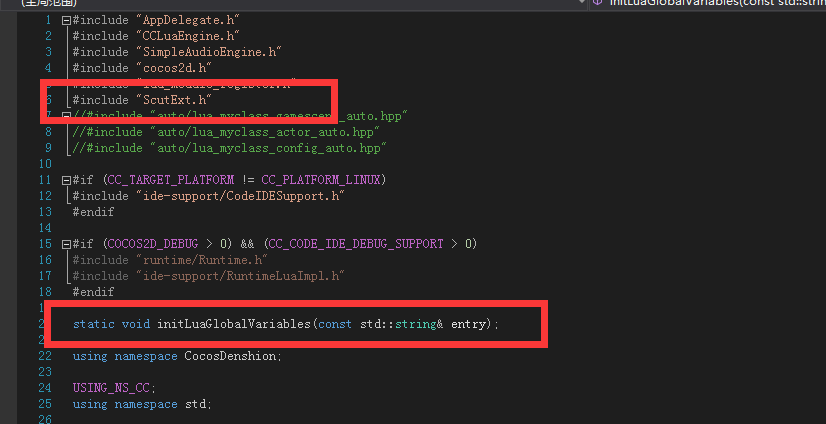


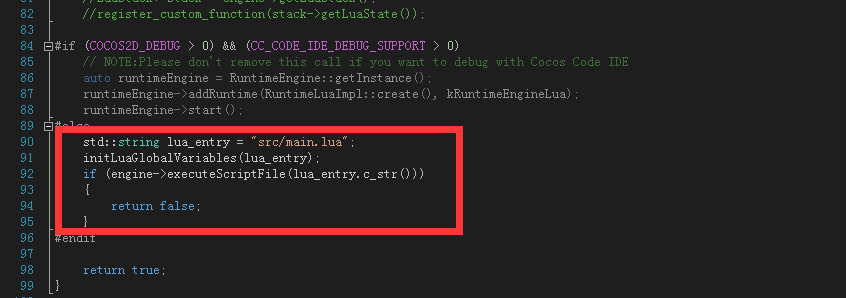
6.添加依赖库，编译scut项目之后会生成一个静态库ScutSDK\_2012.lib在Debug.win32下，引入这个ScutSDK\_2012.lib;



到这一步之后基本配置就搞定了，我们接下来添加代码调用

6.修改appdelegate.cpp







7.NetHelper.lua

--ZyReader.recordBegin()：结构体中，开始解析二进制流

--ZyReader.recordEnd()：结构体闭合，结束解析二进制流

--ZyReader.getBYTE()：解析unsigned char（byte，1字节）方法。

--ZyReader.getWORD()：解析unsigned short（WORD，2字节）方法。

--ZyReader.getDWORD()：解析unsigned int（DWORD，4字节）方法。

--ZyReader.getFloat()：解析float方法。

--ZyReader.getDouble()：解析double方法。

--ZyReader.readString()：解析String方法。

--ZyReader.readInt64()：解析Int64方法。

--ZyReader.getInt()：解析Int32方法。

--ZyReader.readErrorMsg()：解析错误消息。该错误消息需要与服务端约定。当网络请求发生异常，或者业务逻辑出现错误时候，打印的消息。

--ZyReader.getResult()：获取当前网络请求状态，服务端返回错误码，该错误码需要与客户端约定。

--ZyReader.getActionID ()：获取当前请求的ActionID。

**ZyRequestParam** = {param = {}}

**function** **ZyRequestParam**:getParamData(nTag)

**return** **ZyRequestParam**.param[nTag]

**end**

**ZyRequestTag** = 0

**ZyReader** = **ScutDataLogic**.CNetReader:getInstance()

**ZyWriter** = **ScutDataLogic**.CNetWriter:getInstance()

**eScutNetSuccess** = 0

**eScutNetError** = 10000

**function** **ZyReader**.readString()

**local** nLen = **ZyReader**:getInt()

**local** strRet = **nil**

**if** nLen ~= 0

**then**

**local** str = **ScutDataLogic**.CLuaString:new("")

**ZyReader**:getString(str, nLen)

strRet = **string**.format("%s", str:getCString())

str:delete()

**end**

**return** strRet

**end**

**function** **ZyReader**:readInt64()

**return** **ScutDataLogic**.CInt64:new\_local(**ZyReader**:getCInt64())

**end**

**function** **ZyReader**.readErrorMsg()

**return** **string**.format("%s", **ZyReader**:getErrMsg():getCString())

**end**

--参数pScutScene 为请求接口发起的scene对象，此对象为CCScene类型。

--参数lpData为用户附加数据，该数据将被直接传回。

**function** **ZyExecRequest**(pScutScene, lpData, addressPath)

**ZyRequestTag** = **ZyRequestTag** + 1

**if** lpData **then**

**table**.insert(**ZyRequestParam**.param, **ZyRequestTag**, lpData)

**end**

**if** addressPath == **nil** **then**

**ScutDataLogic**.CDataRequest:Instance():AsyncExecRequest(pScutScene, **ZyWriter**:generatePostData(), **ZyRequestTag**, **nil**)

**else**

**local** lenth=**string**.len(**ZyWriter**:generatePostData())

**ScutDataLogic**.CDataRequest:Instance():AsyncExecTcpRequest(pScutScene,addressPath, **ZyRequestTag**, **nil**, **ZyWriter**:generatePostData(), lenth)

**ScutDataLogic**.CNetWriter:resetData()

**end**

**ScutDataLogic**.CNetWriter:resetData()

**End**

8. ScutManager.lua

-------------------------------------------------------------------

--文件：ScutManager.lua

--作者：wangsheng

--日期：2015年6月30日

--描述：服务器接口管理器

-------------------------------------------------------------------

**import**(".NetHelper")

**function** **OnHandleData**(pScene, nTag, nNetRet, lpData, lpExternal)

**ScutManager**:execCallback(nTag, nNetRet, lpData,lpExternal)

**end**

**ScutManager** = **class**("ScutManager")

**local** loadingCache = {}--loading缓存

**local** callbackCache = {}--回调函数

**local** callTag = 0--调用的id递增

**ScutManager**.url = "192.168.2.98:9001"--服务器地址

--初始化

**function** **ScutManager**:init()

**ScutDataLogic**.CNetWriter:setUrl(**ScutManager**.url)

**cc**.Director:getInstance():getScheduler():scheduleScriptFunc(**function**()

**ScutDataLogic**.CDataRequest:Instance():PeekLUAData()

**end**, 0, **false**)

**end**

--执行网络回调函数。

**function** **ScutManager**:execCallback(nTag, nNetState, lpData,lpExternal)

**local** actionID = **ZyReader**:getActionID()

**local** lpExternalData = **lpExternalData** **or** 0

**local** userData = **ZyRequestParam**:getParamData(lpData)

**if** 2 == nNetState **then**

**local** reader = **ScutDataLogic**.CDataRequest:Instance()

**local** bValue = reader:LuaHandlePushDataWithInt(lpData)

**if** callbackCache[nTag] **then**

callbackCache[nTag]()

callbackCache[nTag]=**nil**

**end**

self:netDecodeEnd(nTag)

**end**

**end**

--调用接口

--callBack回调函数

--url地址

--lpData为用户附加数据，该数据将被直接传回。

**function** **ScutManager**:doHandler(callBack,url,lpData)

**ZyExecRequest**(**cc**.Director:getInstance():getRunningScene(),lpData,url **or** **ScutManager**.url)

**if** callBack **then**

callbackCache[**ZyRequestTag**] = callBack

**end**

--添加遮罩

-- local loading = ClassManager.createInstance("LoadingLayer")

-- loadingCache[ZyRequestTag] = loading

-- UIManager.push(loading)

**end**

--执行完毕

**function** **ScutManager**:netDecodeEnd(nTag)

--调用完毕，如有loading 移除loading

-- local loading = loadingCache.nTag

-- if loading then

-- UIManager.pop(loading:getId())

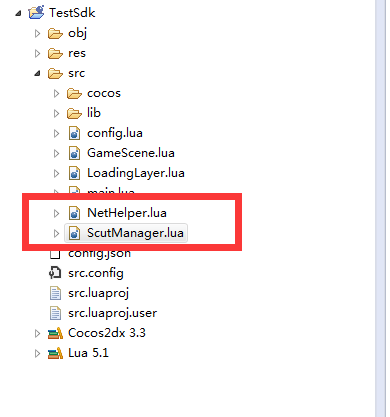
-- loadingCache.nTag = nil

-- end

**end**

**ScutManager**:init()

9.这两个类放同一个目录下



Lua代码和scut修改过的项目在附件中

10.使用方法

