1. **安装jdk，sdk。Sdk在ftp上就有的下载。安装完毕后在客户端Edit/Preferences，选择External tools，在最下方配置Android的jdk和SDK。**
2. **创建目录branch010，在该目录下创建目录并checkout工程artproject,src,创建export存放打包生成的resource和pkg文件。Export文件夹存放以版本号命名的文件夹。创建Package存放补丁文件。例如0.0.0.1。**

-art

-export

-0.0.0.0

-0.0.0.1

-Package

-src

**注：在src客户端的Assets目录下创建StreamingAssets文件夹。**

1. **修改配置文件，以步骤一的文件夹为例修改配置文件：**

配置文件目录：“../art/Assets/Configs/BuildEnvConfig.xml”。修改如下内容：涉及目录的都要用绝对路径，并且注意路径都是以斜杆结尾的。以你自己本地目录为准，我这里只写后面的关键路径。

isBuildMobileRes = “1”

output = “../branch010/export/”

buildTarget = “Android”

<libs folder=”../branch010/src/GameCode/Lib/”>

<RuntimeResource>../src/RuntimeSource/</RuntimeResource>

<pkgOutoutDir>../Package/</pkgOutoutDir>

<publishStreamAssetDir>../src/RuntimeProjectAssets/StreamingAssets/</publishStreamAssetDir>

1. **打开artproject工程，开始打包Resources和pkg：**

**打pkg:**

在Unity中选择Build/Pkg/03、Build Data

在Unity中选择Build/Pkg/03.5、Build Lua

在Unity中选择Build/Pkg/04、Build Libs

在Unity中选择Build/Pkg/05、Build First Pkg File

到这一步pkg文件已经打包好，执行以下一个步骤将pkg文件复制到src客户端工程中。两者取其一，如果有资源变动则用第一个，如果资源没变动用第二个。

在Unity中选择Build/Pkg/06、CopyToPublishStreamAsset

在Unity中选择Build/Pkg/06.5、CopyOnlyPkgToPublishStreamAsset

**打Resources：**

如果是checkout的是分支，则资源都已经打好了，打好的资源都放在svn上的目录../export/trunk/0.0.0.xx，只要把版本最高的文件夹checkout到export目录即可。再使用：

在Unity中选择Build/Pkg/06、CopyToPublishStreamAsset将资源copy到src客户端工程中。

**如果不需要自己准备资源(即svn上的export目录已经有准备好的0.0.0.x目录)，可以跳过这一步，这里是生成export目录下0.0.0.x文件夹的步骤：**

在Unity中选择Build/Pkg/01、UpVersion，这一步会在export文件夹下创建0.0.0.x目录，会自动生成比原来最高版本还要高版本号的目录。

在Unity中选择Build/Pkg/02、Build Resources，（不要选到Build Update Resources）,这一步操作会执行好几个小时，可以用另一种方式达到目的，找到artproject的Resources/目录，对Resources文件夹下的所有子目录执行：右键-> Build AssetBundle From Selection, 最好是几个目录执行一次，而不是全部一起执行。全部生成完毕，在Unity中选择Build/Pkg/06、CopyToPublishStreamAsset即可把资源copy到

**到这里就将Resources目录和pkg文件复制到了src客户端的StreamingAssets目录下。**

1. **打开src客户端，开始打包**

打包之前首先打开File/Build Setting/Player Settings。查看参数，包名，游戏名之类的。

以下两个配置文件都在../src/Assets/Resources目录下。保证打出来的能够登录还需要修改这两个文件：

配置文件system\_config\_android.xml:

<mode releaseMode=”1”/>

<cfg url=”配置文件注释中有两个可选的路径”/>

//路径239是可以让打出来的包连接到内网的，另一个是连接外网的。连接外网要慎重，不可以随意修改数据和发送言论。如果两个都不可以需要跟项目负责人确认url链接。

配置文件 VersionInfo.xml:

ProgramVersion,App的大版本，这里的版本如果小于服务器端配置的版本会主动下载新的版本,所以这里的版本不能小于服务器端配置的版本号。ResourceVersion，同理不能小于服务器端补丁的版本，一般查看svn export文件夹下最高的版本，不低于这个版本就可以。GameLoaderVersion，如果没有修改过GameLoader这里可以不改，但是如果改过GameLoader则必须保证这里的版本比服务端的高，否则会下载旧版本覆盖。

在Unity中选择 打包看这里/普通打包 或者 打包看这里/普通不升版本打包。优先选择普通不升版本打包。打包之前有弹窗，选择忽略。因为我们不需要高版本工具。

等进度条走完就可以在客户端的RuntimeProject目录下看到生成的包了。

SDK接入

1. 目录所在svn路径，以永恒为例：

<http://192.168.11.241/svn/xgame/qjzr/src/client/sdk/Qjzr>

1. 使用eclipse打开，在MainActivity中添加新增接口方法。一般新增sdk接口，运营方会给你发送新的jar包，将旧版本的jar包从eclipse项目配置路径中去除（旧jar包右键，Build Path, Remove from Build Path,在libs目录下删除不用的jar包），添加新版本的jar包到libs目录下，右键， Build Path, Add To Build Path。这时可以从新的jar包中访问到新的接口。
2. 接口添加完毕，就可以将工程导出成一个jar包供GameCode访问。右键工程，选择export,选择Java/JAR file，点击Next,只选择src，填好路径，接着一路next，finish即可。 导出的jar包名字跟旧的一致就好，以永恒为例命名是qjzr\_unity.jar，该jar包放在src/Assets/Plugins/Android/bin。需要注意一点的是运营方发给你的新jar包要放在src/Assets/Plugins/Android/libs目录下，如果jar包没更新则不需要。
3. 打开GameCode工程，在GameLoader/GameLoader/PlatformSdk/目录下有三个文件，PlatformSdkMgr.cs，IOSSdkMgr.cs，AndroidSdkMgr.cs。PlatformSdkMgr是IOSSdkMgr，AndroidSdkMgr的基类，新增接口需要先在PlatformSdkMgr类中定好接口，子类继承这些接口然后按照平台各自的需求去实现功能。在AndroidSdkMgr中调用第三步导出的jar中添加好的接口，调用过程非常简单，看下其他例子就会明白，带返回值，参数等等都能在AndroidSdkMgr类中找到例子。需要注意的是，再PlatformSdkMgr定义接口需要综合考虑好 接口的参数以及返回值，例如：同一个接口，ios可能有五个参数，而在Android可能不需要参数，这时候需要保证PlatformMgr中定义的接口能满足不同平台调用。最好是接之前向运营方确认好。接好之后重新生成GameLoader。
4. 在GameMain/GameMain/LuaFacade.cs中定义接口暴露给Lua层，该接口使用PlatformSdkMgr调用你在第四步定好的接口，重新生成GameMain，这时候你就能在Lua代码中通过C#层去调用Android接口了。

**Debug:**

连接设备，打开eclipse或者Android Studio，打开DDMS视图，运行游戏，所有的报错都能在DDMS视图中看到。

**常见问题：**

1. **打完包，发现修改都没生效？**

答：查看pkg有没有copy到StreamingAsset目录下；查看src中VersinInfo.xml版本号设置是否低于服务端版本,具体看第一次打开App是否执行更新，下载补丁等操作，或者找到App的内部文件，在Config目录下找到一个VersionInfo.xml文件和src中的VersionInfo.xml文件进行对比。

1. **App登录不上，报客户端过旧提示？**
2. 答：查看artproject，src目录是否更新到最新。跟项目负责人确认system\_config\_android.xml配置的url是否正确。
3. **SDK接完调用没有起作用？**

答：检查接完之后有没有重新生成GameLoader，GameMain。或者版本号是否过低，修改的内容都被覆盖。