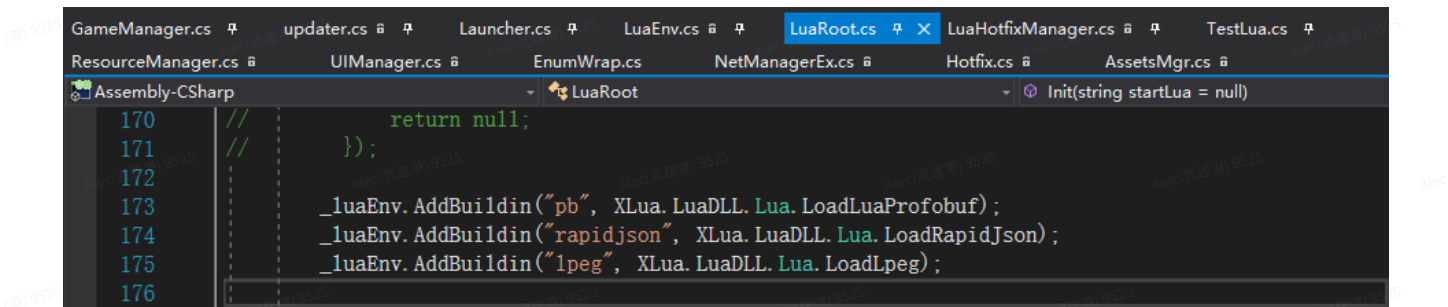


# 《放置大陆》XLua ByteCode的编译和加载

下载XLua源码和几个扩展库：protobuf，rapidjson，lpeg，将这几个库的编译配置添加到CMakeLists.txt当中，在编译动态链接库的时候会把这几个库也编译进去；

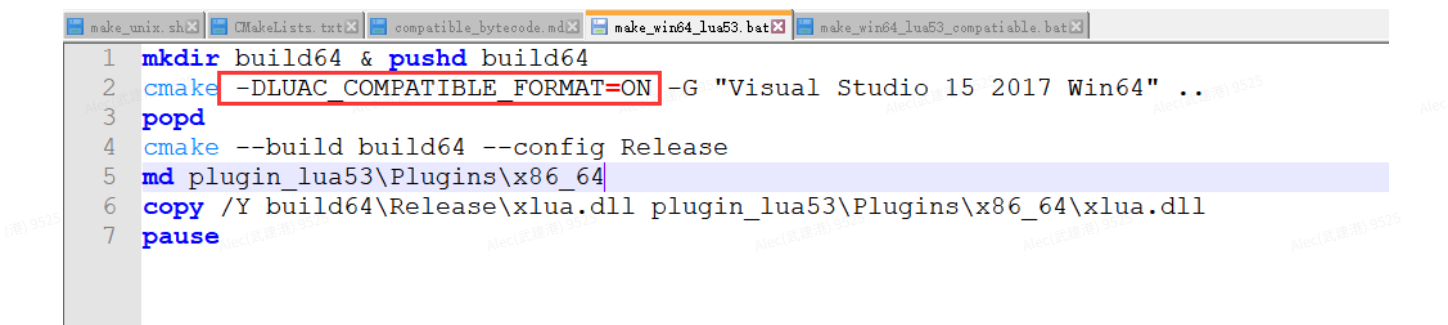
XLua源码+扩展库：[https://github.com/chexiongsheng/build\\_xlua\\_with\\_libs.git](https://github.com/chexiongsheng/build_xlua_with_libs.git)

这几个库的加载逻辑：

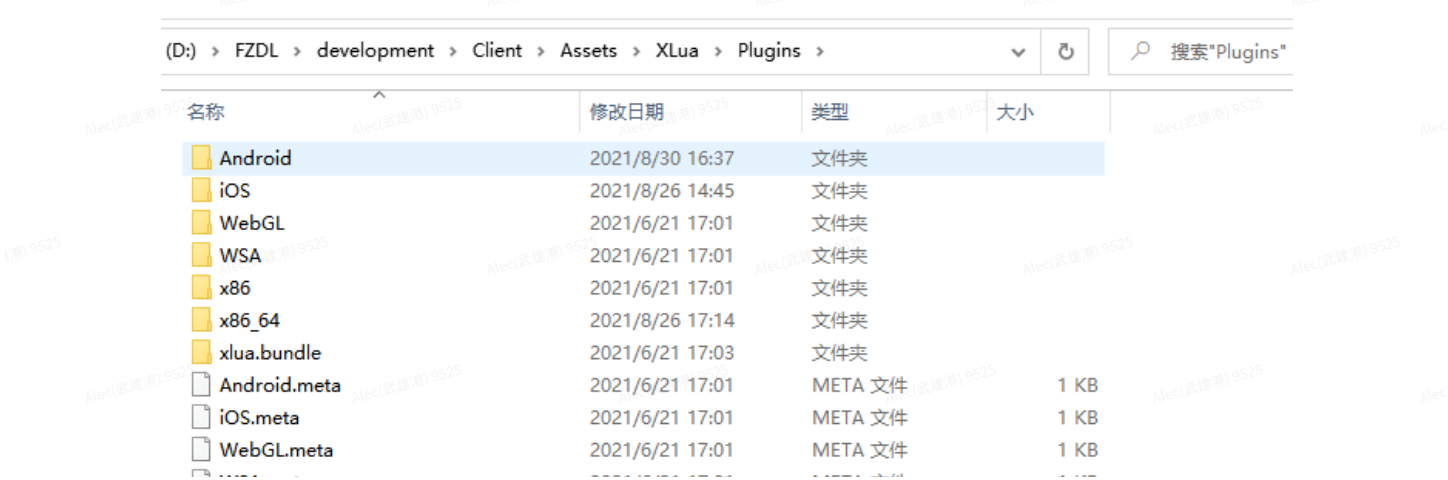


编译的时候增加一个选项，只有加了才支持bytecode加载：-DLUAC\_COMPATIBLE\_FORMAT=ON

XLua在lua的源码上做了一些优化，不用区分32位bytecode和64位bytecode。可以编译一份字节码，跨平台使用



配置编译环境，编译出win，mac，android，ios这四个平台的动态链接库



WSA.meta	2021/6/21 17:01	META 文件	1 KB
x86.meta	2021/6/21 17:01	META 文件	1 KB
x86_64.meta	2021/6/21 17:01	META 文件	1 KB
tolua.bundle.meta	2021/6/21 17:03	META 文件	2 KB

## tolua.exe

使用tolua将lua文件编译为bytecode，先编译需要的tolua工具，linux和mac平台使用make\_unix.sh

temp > bb > build_lua_with_libs_git > build_lua_with_libs_git > build > tolua					▼	🔄	🔍 搜索"tolua"
名称	修改日期	类型	大小				
CMakeLists.txt	2020/12/23 18:45	TXT 文件	4 KB				
make_unix.sh	2020/12/23 18:45	Shell Script	1 KB				
make_win32.bat	2020/12/23 18:45	Windows 批处理...	1 KB				
make_win64.bat	2020/12/23 18:45	Windows 批处理...	1 KB				

Windows下tolua的使用：

tolua.exe -o a.out a.lua

其中a.out是输出文件名，a.lua是输入的lua脚本

批处理（目录下的所有.lua文件都会被编译为bytecode文件）：

```
for /r %%v in (*.lua) do tolua -o %%v %%v
```

将所有的.lua文件后缀修改为.lua.bytes，打进AssetBundle包内，以TextAsset的形式读取加载

## PowerShell

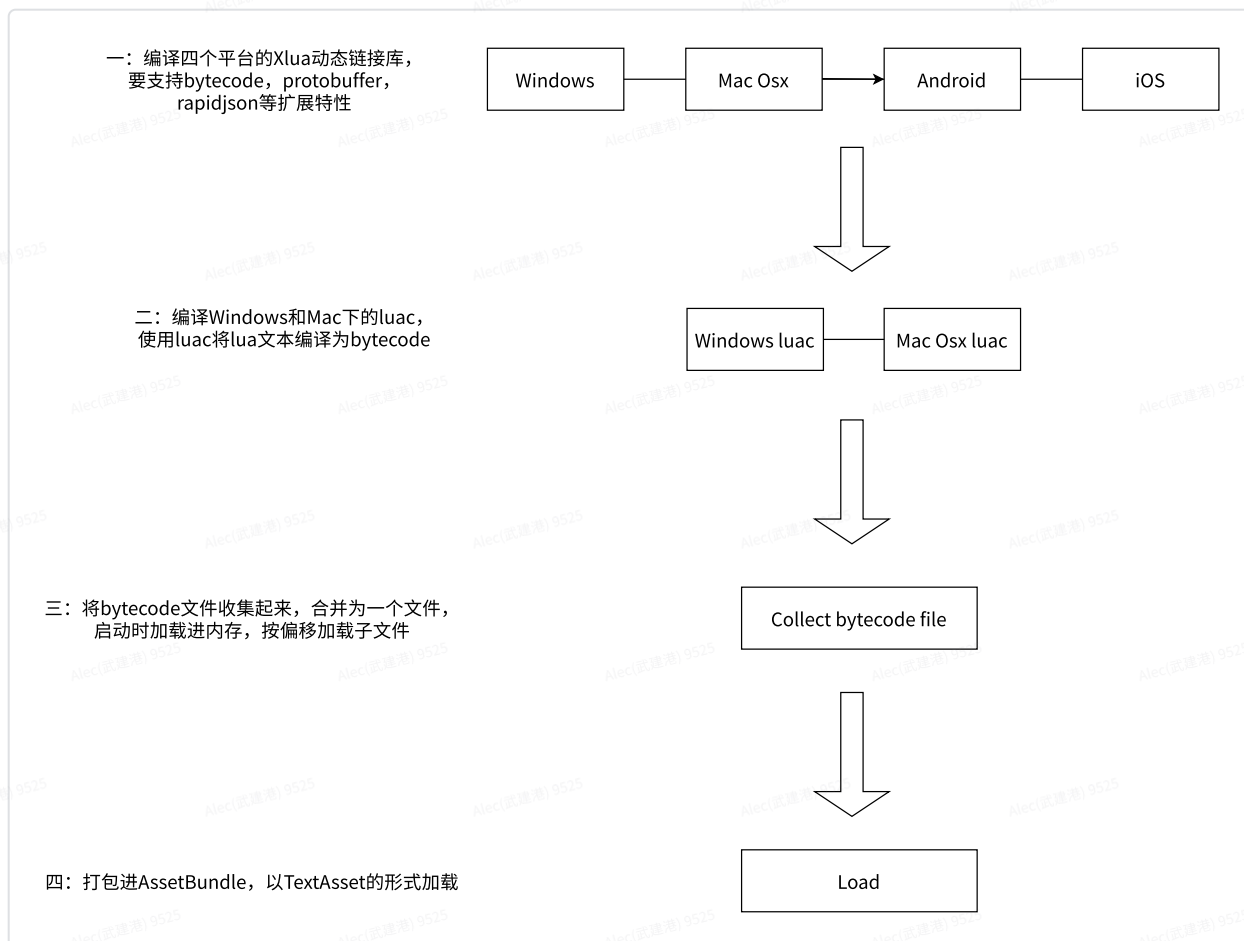
```
1 @set LocalPath=%~dp0
2 echo "this is make.bat"
3 md XLuaByteCode
4
5 for /r %%v in (*.lua) do %LocalPath%luac.exe -o %%v.bytes %%v
6 pause
```

## Mac下luac的使用

## Shell

```
1  #!/bin/bash
2  SHELL_FOLDER=$(cd "$(dirname "$0")";pwd)
3  echo $SHELL_FOLDER
4  cd $SHELL_FOLDER
5  mkdir -p XLuaByteCode222
6  foreachd(){
7      for file in $1/*
8      do
9          # 如果是目录就继续遍历
10             if [ -d $file ]
11             then
12                 echo $file"是目录"
13                 foreachd $file
14             elif [ -f $file ]
15             then
16                 echo $file
17                 if echo "$file" | grep -q -E '\.lua$'
18                 then
19                     echo "true"
20                     $SHELL_FOLDER/luac -o $file.bytes $file
21                 else
22                     echo "false"
23                 fi
24             fi
25         done
26     }
27     chmod +x $SHELL_FOLDER/luac
28     # 执行，如果有参数就遍历指定的目录，否则遍历当前目录
29     if [[ "x$1" == 'x' ]]
30     then
31         foreachd "."
32     else
33         foreachd "$1"
34     fi
```

流程图;



性能测试：

压缩包：2.45M（部分比较大的表格文件未进压缩包）

696个字节码文件，612个小于20kb的压缩在一起启动时加载进内存，剩下的大的单独加载。去除Debug信息，高通855上加载时间是8ms

XLua源码已经上传到了Perforce仓库，如果后续有源码层的改动，可在此工程基础上进行

