# 打卡 写入版本信息

[202208281805](GIF/202208281805.gif)

# 打卡 写入AB包信息

[202208281811](GIF/202208281811.gif)

复制到外面[202208281919](GIF/202208281919.gif)

# 打卡 采用MD5，更新AB的变化

01 打一次包

02 修改标记要打包的某个资源

03 再打一次包

04 点击AB热更

05 可以看到更新AB的变化

[202209141635](GIF/202209141635.gif)

# 打卡 生成热更包配置表Patch

[202209141835](GIF/202209141835.gif)

# 服务器文件部署--------------------------

# Apache 下载

<https://httpd.apache.org/download.cgi#apache24>

[202209141859](GIF/202209141859.gif)

# 打卡 Apache 配置软件位置

[202209141908](GIF/202209141908.gif)

# bug 安装后不存在服务 服务开启失败

以下加颜色有用

查所有（管理员权限下cmd）

netstat -ano

查PID

tasklist |findstr 4

01 打开（或管理者运行）httpd.exe，窗口闪退

02 netstat -ano有端口占用情况，不知道是谁



03 改端口，再次打开依然闪退

D:\ProgramFilesTrim\Apache\Apache24\conf\httpd.conf

Listen 8081

...

ServerName localhost:8081

04 管理员权限cmd到bin目录，httpd.exe -k install

没用，这是个安装命令

05 管理员权限cmd到bin目录，

D:\ProgramFilesTrim\Apache\Apache24\bin\httpd.exe -k install -n “Apache2.4”

D:\ProgramFilesTrim\Apache\Apache24\bin\httpd.exe -k uninstall "Apache2.4"这个是卸载

双引号里面是什么，服务名也就是什么。但是服务启动不了

[202209142043](GIF/202209142043.PNG)

06 安装别的版本，走到生成服务时。mmp，80没解决，又来443，这么喜欢抢端口



07 杀死PID4，不能杀死

taskkill /PID PID /F

09 442(我的被占用了)全改443（我的这个没被占用）端口apache\conf\extra\httpd-ssl.conf

## 01 以上总结

01

改端口，再次打开依然闪退

D:\ProgramFilesTrim\Apache\Apache24\conf\httpd.conf

Listen 8081（我的这个没被占用）

...

ServerName localhost:8081

02

442(我的这个被占用了)全改443（我的这个没被占用，后面发现也被占用）端口apache\conf\extra\httpd-ssl.conf

03（这个关乎有没有服务）

管理员权限cmd到盘符（ProgramFilesTrim是不确定支不支持空格）

D:\ProgramFilesTrim\Apache\Apache24\bin\httpd.exe -k install -n “Apache2.4”

D:\ProgramFilesTrim\Apache\Apache24\bin\httpd.exe -k uninstall "Apache2.4" 这个是卸载



99 补充

（管理员权限下cmd）

查所有，netstat -ano

查PID ，tasklist |findstr 4

99 补充 后面出现了端口占用VSVN占用了443，结束掉了该进程（这玩意每次自启）（后来发现VSVN还会抢我Apache的端口，但也可能是时间久Apache关了）

99 补充，窗口又闪退，但是不影响，后台有进程



# 打卡 下载安装到打开apache首页

[202209142322](GIF/202209142322.gif)

# 了解 将AB变化移动的位置

D:\ProgramFilesTrim\Apache\Apache24\htdocs\AssetBundle

# 打卡 服务器配置表

复制 D:\Data\Projects\Unity\Ocean\_RealFram\_20220710\_2018.2.10f1\RealFram\_20220710\_2018.2.10f1\Hot\StandaloneWindows64\Patch.xml

到D:\ProgramFilesTrim\Apache\Apache24\htdocs\ServereInfo.xml

修改相关内容

[202209151420](GIF/202209151420.xml)

# bug AB总包名字不一样

打包后会自动生成StreamingAssets文件，但Ocean的叫StandaloneWindows64

所以把内部打包的位置加上平台名，导到外面，有趣的是，发现原来的StreamingAssets自动替换成StandaloneWindows64

[202209151413](GIF/202209151413.gif)

# 了解 下载的流程

## 工程的Hot文件夹

D:\Data\Projects\Unity\Ocean\_RealFram\_20220710\_2018.2.10f1\RealFram\_20220710\_2018.2.10f1\Hot\StandaloneWindows64



第一次全部打包，第二次修改某些文件（我选了image），再次打包出现的。反映打包的变化

StandaloneWindows64是文件夹的名字

**Patch.xml（自动）**

<?xml version="1.0" encoding="utf-8"?>

<Pathces xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" Version="1">

<Files Name="image" Url="http://127.0.0.1/AssetBundle/0.1/1/image" Platform="StandaloneWindows64" Md5="25925f17e67aaa08aa80dff09a1e87d5" Size="5.42773438" />

<Files Name="StandaloneWindows64" Url="http://127.0.0.1/AssetBundle/0.1/1/StandaloneWindows64" Platform="StandaloneWindows64" Md5="bd254a0eb7f6ea9ada10e8c9d03d556a" Size="1.75878906" />

</Pathces>

## Apache

D:\ProgramFilesTrim\Apache\Apache24\htdocs

index.html自带的

其它都要自己创建（根据Hot文件夹的内容，复制过来）



D:\ProgramFilesTrim\Apache\Apache24\htdocs\AssetBundle\0.1\1



Apache服务器弄好后，可以访问（8081是D:\ProgramFilesTrim\Apache\Apache24\conf\httpd.conf照片呢个设置的，之前有说）



**ServerInfo.xml（手动复制Patch修改）**

<?xml version="1.0" encoding="utf-8"?>

<ServerInfo>

<GameVersion Version="0.1">

<Pathces Version="1" Des="测试热更">

<Files Name="image" Url="http://127.0.0.1/AssetBundle/0.1/1/image" Platform="StandaloneWindows64" Md5="25925f17e67aaa08aa80dff09a1e87d5" Size="5.42773438" />

<Files Name="StandaloneWindows64" Url="http://127.0.0.1/AssetBundle/0.1/1/StandaloneWindows64" Platform="StandaloneWindows64" Md5="bd254a0eb7f6ea9ada10e8c9d03d556a" Size="1.75878906" />

</Pathces>

</GameVersion>

</ServerInfo>

## PC模拟下载的地方

C:\Users\lenovo\AppData\LocalLow\DefaultCompany\RealFrame（RealFrame是PlayerSettings设置的项目名）

主要验证ServerInfo.xml有没有从Apache中更新过来

在HotPatchMgr.cs中，我用第一句得不到，用第二种的得到了（在工程中报的错误是xml转ServerInfo，因为xml为空）

//private string m\_hotCfg = "http://127.0.0.1/ServerInfo.xml"; //热更配置表

private string m\_hotCfg = "http://localhost:8081/ServerInfo.xml";



## 建议 做时用快捷方式



# bug Editor文件夹外不能使用

打包时，第一句报错，第二句可以

就是打包时，会检查项目编译，发现不在Editor的文件使用了EditorUserBuildSettings.activeBuildTarget.ToString()，会报错、

暂时是打包之前去设置一个字符串

EditorUserBuildSettings.activeBuildTarget.ToString()

StandaloneWindows64

# bug 校验码错误，重新下载4次仍失败

7e,b3是代码中MD5的最后两位数字

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | new | old | local | server |  |
| Apache |  |  |  |  | b3 |
| PC下载 |  |  | 5a | b3 |  |
| 代码 | 7e | b3 |  |  |  |

|  |  |
| --- | --- |
| m\_DownLoadMD5Dic | downLoadAssetLst |
| b3 |  |
| Url "http://127.0.0.1/AssetBundle/0.1/1/image" | downLoad.SaveFilePath = "C:/Users/lenovo/AppData/LocalLow/DefaultCompany/RealFrame\_Test/DownLoad/image" |

加黄有问题，

我改成

void AddDownLoadList(Patch patch)

{

patch.Url = patch.Url.Replace("http://127.0.0.1","http://localhost:8081");//http://127.0.0.1==http://localhost

后来想想8081是当做服务器的，不能直接用

得要开启能访问http://127.0.0.1的服务

**管理者cmd sc config http start= demand & net start http**





# 打卡 从Apache下载到本地

01 打一次全包（之前打过就不用了）

02 热更编辑器打一次（之前打过就不用了）

03 修改某个资源，再打一次包

04 热更编辑器再打一次。得到（圈圈是固定的）

（在D:\Data\Projects\Unity\Ocean\_RealFram\_20220710\_2018.2.10f1\RealFram\_20220710\_2018.2.10f1\Hot\StandaloneWindows64）



05 将第一第二个复制到

D:\ProgramFilesTrim\Apache\Apache24\htdocs\AssetBundle\0.1\1

06 将Patch修改D:\ProgramFilesTrim\Apache\Apache24\htdocs\ServerInfo.xml

内容看以前的“了解 下载的流程”

07 进行测试（要复现这个场景，需要重复03-06，或者删除"C:/Users/lenovo/AppData/LocalLow/DefaultCompany/RealFrame\_Test/DownLoad/”文件夹）

[202209202025](GIF/202209202025.gif)

# 了解 80与8081；localhost与127.0.0.1

http://127.0.0.1/ServerInfo.xml（默认是80端口，所以等于http://127.0.0.1:80/ServerInfo.xml）

但是Apache改成8081后，这里也要改成

<http://127.0.0.1:8081/ServerInfo.xml>

或者localhost:8081/ServerInfo.xml

具体位置是生成热更包Editor和热更Manager

# 打卡 从Apache下载到本地

[202209271750](GIF/202209271750.gif)

# 打卡 热更包

可以看到image和64这两个包的修改时间是58分（其他的是50分）

[202209271800](GIF/202209271800.gif)

当我把C:\Users\lenovo\AppData\LocalLow\DefaultCompany\RealFrame\_Test\Download中用到的包删除，会报错。也就是AB包内部会根据打包时的目录形成一个局部目录结构。注意的事，不要因为AB包中的路径与unity的资源路径一样，就以为加载AB时用的是unity中的资源。

[202209282218](GIF/202209282218.gif)

# bug 下载到本地后

本地的，和assetbundleconfig认定的untiy中打包的路径是不一样的，那么crc也是不同的，就拿到不到resItem

因为尝试强制设置一个crc是可以拿到resItem的

[202209272105](GIF/202209272105.PNG)

以上的影响就是：实际的打包的m\_Path要统一（Ocean可能后面会统一）

## 了解

043-下载测试 10：59（括号内是给一个关键字命名）

Ocean没有引用PC模拟下载的地方C:\Users\lenovo\AppData\LocalLow\DefaultCompany\RealFrame\_Test\Download（Local）

而是复制了外部的AB包D:\Data\Projects\Unity\Ocean\_RealFram\_20220710\_2018.2.10f1\RealFram\_20220710\_2018.2.10f1\AssetBundle（Outter）

到内部D:\Data\Projects\Unity\Ocean\_RealFram\_20220710\_2018.2.10f1\RealFram\_20220710\_2018.2.10f1\Assets\RealFrame\StreamingAssets\StandaloneWindows64（Inner）

# 打卡 44 AB包加载 热更回退

**热更回退的代码逻辑**

GameStart Open（HotfixWnd）//打开热更面板

HotfixWnd(Hotfix())//热更检查

AssetBundleMgr//中有路径判断是加载热更的还是本地的

**操作流程**

全打AB包一次，作为本地资源，暂时复制一份，演示时放到StreamingAsset中作为本地ab加载.

修改图片，再打AB包，再打热更AB包，部署到Apache

Ocean的是，热更的脸是花的，干掉就加载不花的

我的是，老的是全红的，新的（热更）的有一杠黑。从黝黑到全红，演示了热更回退

[202209292338](GIF/202209292338.gif)

# 打卡 AES加密（写死是某个文件）

[202209301005](GIF/202209301005.gif)

# 了解 FileStream

FileStream类型对象是引用类型，并且作为参数传参时，可以不用ref returm out来获取变化

# 打卡 AES解密（写死是某个文件）

[202209301417](GIF/202209301417.gif)

# bug System.Security.Cryptography.CryptographicException: Padding is invalid and cannot be removed.

fs.Seek(0, SeekOrigin.Begin); //读完 seek回去，保持原始状态

上面的，放1的后面就报错，2的前面就正常

[202209301419](GIF/202209301419.PNG)

# 打卡 AB包的加密解密（AES）

[202209301523](GIF/202209301523.gif)

# 打卡 删除外部AB路径中的.manifest

[202209302008](GIF/202209302008.gif)

然后加密，为后面加载解密AB做准备。

我把操作分开写，因为我打好了包，不想重跑一遍，所以增量用MenuItem

# 了解 安卓下不能FileStream 读取AES加密的AB

所以要解压

IOS。PC则不用加压

# 了解 C# 类中属性

编辑器面板不显示

# 打卡 WebRequest下载加密包到本地的Origin

手动//把加密的ab复制到内部StreamingAssets（节省上传服务器再下载）

HotPatchMgr.Unpack2PersistentDataPath//解压（加密的ab）到Application.persistentDataPath + "/Origin";

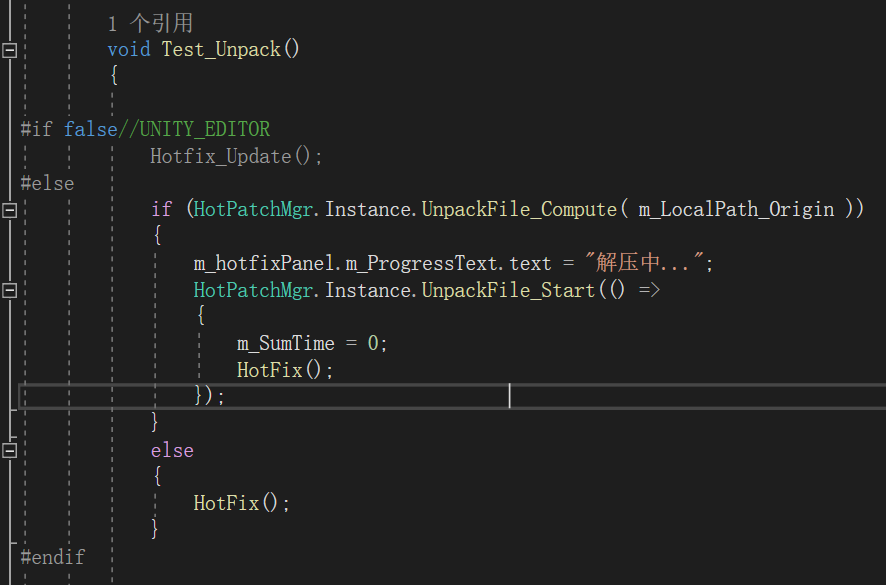
[202210020018](GIF/202210020018.gif)

后面报的是Stream中不允许使用FileStream的错误，后面再处理

# 打卡 解压热更及加载测试

## 01

下载后的包是不是解密的（测试是加密的，跟从Apache下载到DownLoad是一样的原理。这里是用WebRequest从StreamingAssets下载加密包到本地的Origin）。注意HotfixWnd中，走下面的代码去解压

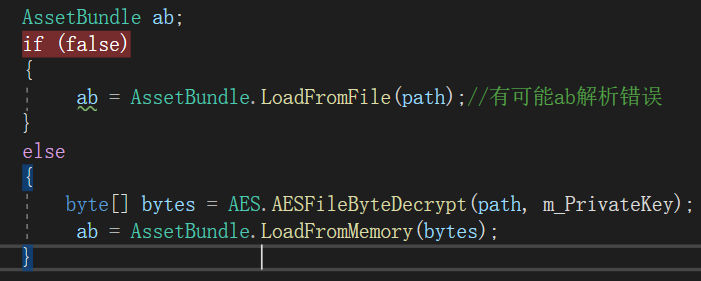


## 02

使用资源（图片，预制体）时，ab包的加载。

看到ABMgr中，参照abCfg的加载有AssetBundle.LoadFromFile和AssetBundle.LoadFromMemory

aABMgr.LoadABItem中也修改。



## 03

加载到了，判断是不是解压的，看其中的脚本图标。

但我加载的是图片，致命的是前期的Mono脚本挂预制体报错的问题没解决。

得回去让它能挂载脚本（也不知道为什么挂在脚本没报错了）

[202210022039](GIF/202210022039.gif)