**外服维护说明：**

1. **新服serverkey对应的服务器ip在./shell/master/info/server.list中配置（注意使用内网ip，同时配置lualib/public/serverinfo.lua，并修改serverinfo.xlsx导表。）**
2. **外服服务器都会对应有一个分支，正式外服是release。外网测试服维护前一天晚上跑middle，之后跑release。**
3. **留意：外网测试服和正式外服都通过h7d-oc跳过去。外网测试服周三晚上用middle，周四早上改为用release代码，和外服保持一致。周三晚上合并到middle，其余时间在线更新维护合并到release。之所以有middle是解决周三晚上合完代码修复外服紧急bug用。仿真服直接通过开发服同步及登陆，无需经过同步服。**
4. **每周二晚上，从trunk合并代码到branches/h7d-middle，直接本地meld对比两个分支来合并就行（合并前记得先update，并清理，保持两个目录纯净，只合server目录，doc目录由策划合并，其中导表的init.lua文件中USE\_TMP\_VERSION项在h7d-release和h7d-middle中必须为false，务必配合检查。编译产物不用合；Version.lua版本号+1，client\_update\_res未false;清空client\_update\_code文件；middle的导表gamedata/client需要清空，没有就算了）**
5. **登陆开发服，进入middle目录，关闭仿真服，先后执行.**

**./shell/master/gs\_close.sh h7demu**

**./shell/master/cs\_close.sh h7demu**

**./shell/master/bs\_close.sh h7demu**

**关闭gs、cs、bs。**

1. **执行./shell/update.sh，make clean并make。**
2. **执行./shell/master/sync\_server.sh h7demu，将代码同步到仿真服。**
3. **执行**

**./shell/master/bs\_run.sh h7demu**

**./shell/master/cs\_run.sh h7demu**

**./shell/master/gs\_run.sh h7demu**

**启动bs、cs、gs。然后执行./shell/master/open\_gate.sh h7demu 3开放登陆。**

1. **每周三晚上，将middle代码同步到外测服，重启外测服。**
2. **ssh h7d-oc登陆同步服，进入middle目录下，先后执行**

**./shell/master/gs\_close.sh h7doutertest**

**~~./shell/master/cs\_close.sh h7doutertest~~**

**~~./shell/master/bs\_close.sh h7doutertest~~**

**关闭gs、~~cs、bs~~。注意外网测试服也属于正式服集群，不能关闭CS，BS**

1. **切回开发服，进入middle目录（开发服对应每个分支都有一个目录，进入对应目录）。**
2. **执行./shell/master/sync\_control.sh h7d，将代码同步到同步服。**
3. **ssh h7d-oc登陆同步服，进入middle目录，执行**

**./shell/master/sync\_server.sh h7doutertest**

**同步代码到外网测试服（集群在shell/master/list/目录下定义）。**

1. **看同步脚本输出，是否都同步成功。登陆h7doutertest（从 h7-oc登录 ssh h7d-outertest）进入h7dpro\_gs90001目录查看version.out看版本是否为刚传过来的版本。**
2. **在同步服middle目录下，先后执行**

**~~./shell/master/bs\_run.sh outertest、~~**

**~~./shell/master/cs\_run.sh outertest、~~**

**./shell/master/gs\_run.sh outertest**

**启动~~bs、cs~~、gs。**

**然后执行./shell/master/open\_gate.sh h7doutertest 3开放登陆。客户端打包，登陆外网测试服测试。**

1. **维护当天早上，检查是否有需要从trunk合并到h7d-middle的代码。合并完成则将h7d-middle分支代码copy到h7d-release分支，包括server和doc两个svn目录。**
2. **ssh h7d-oc跳转到同步服。进入release目录，7点执行**

**./shell/master/gs\_close.sh h7dproduction**

**./shell/master/cs\_close.sh h7dproduction**

**./shell/master/bs\_close.sh h7dproduction关闭正式服gs、cs、bs。**

1. **执行./shell/master/mongobak.sh h7dproduction备份正式服数据，确认各服务器备份成功，因为权限问题备份失败，联系运维处理。**
2. **切换回开发服，进入release目录，执行./shell/update.sh和make更新编译代码，看是否需要make clean。**
3. **执行./shell/master/sync\_control.sh h7d，将最新代码同步到同步服务器。**
4. **跳转到h7d-oc同步服，进入release目录，执行**

**./shell/master/sync\_server.sh h7dproduction**

**同步代码到正式游戏服。选择登陆一台外服查看version.out是否正确。**

1. **在h7d-oc同步服release目录下，先后执行**

**./shell/master/bs\_run.sh h7dproduction**

**./shell/master/cs\_run.sh h7dproduction**

**启动bs、cs。然后执行**

**./shell/master/gs\_run.sh h7dpro\_gs10001**

**启动1服让测试和发行测试（由于公司ip会变动，咨询运维ip白名单是否正确）。**

1. **测试没问题执行./shell/master/open\_gate.sh h7dpro\_gs10001 3开放登陆。**
2. **登陆一段时间没问题，起全服。执行**

**./shell/master/gs\_run.sh h7dproduction**

**./shell/master/open\_gate.sh h7dproduction 3。**

**外服在线更新说明：**

1. **从trunk合并到release。合并注意用工具对比，不要整个文件合并，只提交自己修改的内容。如果是doc，让策划自己填写excel并run.cmd，不要整个合并过来，只修改要提交的excel单元格，程序检查svn的lua提交记录看是否有遗漏，多余的提交。**
2. **在开发服release目录执行./shell/update.sh。**
3. **执行./shell/master/sync\_control.sh h7d，将代码同步到同步服。**
4. **ssh h7d-oc登陆同步服，进入release目录，执行**

**./shell/master/sync\_server.sh h7doutertest同步到外网测试服。**

1. **更新lua文件：在同步服release目录执行**

**./shell/master/update\_code.sh h7doutertest。**

1. **更新res：在同步服release目录执行**

**./shell/master/update\_res.sh h7doutertest。**

1. **更新fixbug：在同步服release目录执行**

**./shell/master/update\_fix.sh h7doutertest fix\_name。**

1. **通知测试在外网测试服测试。**
2. **测试通过后，看svn提交记录，确认自上次提交后没有人提交release，如果有，等待全部测完一起更新。**
3. **在同步服的release目录下，执行./shell/master/sync\_server.sh h7dproduction同步到游戏服务器。**
4. **在同步服务器，执行已经执行过的3-5中的指令，h7doutertest换成h7dproduction。**

**外服客户端代码在线更新说明：**

1. **客户端修改服务器trunk目录cs\_common/code/src/clientupdatecode.lua文件，将在线更新代码追加到lualib/public/version.lua的CLIENT\_UPDATE\_CODE字符串后面，提交，在开发服执行update\_cs\_common，然后执行client\_code指令，测试。**
2. **测试通过，通知服务器组员同步到外网测试服测试，合并步骤1中文件到release，登陆开发服，进入release目录，执行./shell/update.sh，**
3. **执行./shell/master/sync\_control.sh h7d，将代码同步到同步服。**
4. **登陆同步服，进入release目录，执行**

**./shell/master/sync\_ server.sh h7doutertest同步到外网测试服。**

1. **执行./shell/master/client\_code.sh h7doutertest更新外网测试服客户端在线更新代码。**
2. **通知测试在外网测试服测试。**
3. **测试通过后，在h7d-oc同步服的release目录下，执行./shell/master/sync\_ server.sh h7dproduction同步到游戏服务器。**
4. **在同步服务器执行./shell/master/client\_code.sh h7dproduction。**

**外服客户端导表在线更新说明：**

1. **确保release分支 lualib/public/version.lua的CLIENT\_UPDATE\_RES = true**
2. **策划将需要在线更新的excel表合并到release，客户端在策划目录的release分支修改并执行clientzip.cmd，提交生成的文件。**
3. **剩余步骤同上，其中第5、9步命令修改为./shell/master/client\_res.sh。**