**正式外服维护：**

**1、正式外服对应branch分支为release。**

**2、维护前， 从trunk合并代码到release，直接本地meld对比两个分支来合并就行（合并前记得先update，一般需要合server和doc两个svn目录，编译产物不用合），在需要维护的当天将release的代码copy到stable分支提交。**

**3、登录开发服，进入stable分支**

**4、执行./shell/make\_code.sh，进行代码更新和编译。(也就是在编译在内网机器完成)**

**5、执行./shell/sync\_control.sh，此脚本目的是将对应分支代码同步到外网总控服。**

**6、此时ssh到外网总控服，进入到stable分支。执行./shell/master/sync\_server.sh listname(listname为shell/master/list下的.list文件名) ，传输到shell/master/list/ listname.list文件中描述的一组服务器上。**

**7、执行./shell/master/gs\_close listname 保存数据并关闭gs**

**执行./shell/master/cs\_close listname 关闭cs**

**执行./shell/master/bs\_close listname 关闭bs**

**执行./shell/master/gs\_run.sh listname启动gs并不打开登录开关，等待测试。**

**执行./shell/master/open\_gate.sh listname 2开放玩家注册**

**执行./shell/master/open\_gate.sh listname 3开放玩家登录**

**8、serverkey对应的服务器ip在./shell/master/server.list中配置。（注意同时配置lualib/public/serverinfo.lua以确保客户端收到的服务器对应ip是正确的。）**

**外服在线更新：**

1. **将需要在线更新的内容文件，copy至stable分支当中，提交（切记不要将正在开发中内容提交）**
2. **ssh至开发服，在stable中分支目录执行./shell/update.sh。**
3. **执行./shell/sync\_control.sh，将代码同步至总控服。**
4. **ssh至总控服，同维护一样，在stable中执行./shell/sync\_server.sh listname**
5. **执行./shell/master/update\_code.sh listname filename (filename有多个，可用逗号隔开)。**
6. **res更新，执行./shell/master/update\_res.sh listname 。**
7. **fixbug 更新，执行./shell/master/update\_fixbug.sh listname funname。**