

Android studio Android源码开发环境搭建

1 起因

android studio工具对java代码自动提示和搜索等功能，把服务的代码全部导入整个工程，在加载的过程特别耗时，只需要我们自己开发的一部分代码，通过gradle脚本用scp命令在windows和linux之间复制文件，将android studio中的修改同步到服务器

如果需要将整个代码导入到Android studio中，可以参考下面这篇博客：

[AndroidStudio源码开发环境搭建 - Gityuan博客 | 袁辉辉的技术博客](#)

2 环境搭建

1 安装git工具，将git和git自带的bash全部加到windows环境变量中

> 此电脑 > Windows (C:) > Program Files > Git > bin

名称	修改日期	类型	大小
bash.exe	2021/8/24 10:09	应用程序	45 KB
git.exe	2021/8/24 10:09	应用程序	45 KB
sh.exe	2021/8/24 10:09	应用程序	45 KB

2 scp免密在windows和Linux服务器之间复制文件

- ssh-copy-id命令把本地的ssh公钥文件添加到远程主机对应的账户下

```
cd ~/.ssh/  
  
// remote_username = 远程Linux服务器用户名  
// remote_ip = 远程Linux服务器IP地址  
ssh-copy-id -i ~/.ssh/id_rsa.pub remote_username@remote_ip
```

- 用scp命令测试看看是否配置OK

```
// remote_username = 远程Linux服务器用户名  
// remote_ip = 远程Linux服务器IP地址  
scp local_file remote_username@remote_ip:remote_file  
scp remote_username@remote_ip:remote_file .
```

3 项目配置

1 将android studio工程克隆或者下载到本地

https://github.com/980725866/android_frameworks

2 Android Studio 配置

```
// project config start
ext {
    // 服务器用户名
    userName = "ubuntu"
    // 服务器IP
    serverIp = "1.1.1.1"
    // 服务器项目代码根目录路径
    projectPath = "/home/ubuntu/Android/"
    // 对应的board名
    targetBoard = "t982_ar301"

    SCP_PROJECT_PATH = "scp -p
${rootProject.ext.userName}@${rootProject.ext.serverIp}:${rootProject.ext.projectPath}/"
    PROJECT_PATH =
"${rootProject.ext.userName}@${rootProject.ext.serverIp}:${rootProject.ext.projectPath}/"
}
// project config end
```

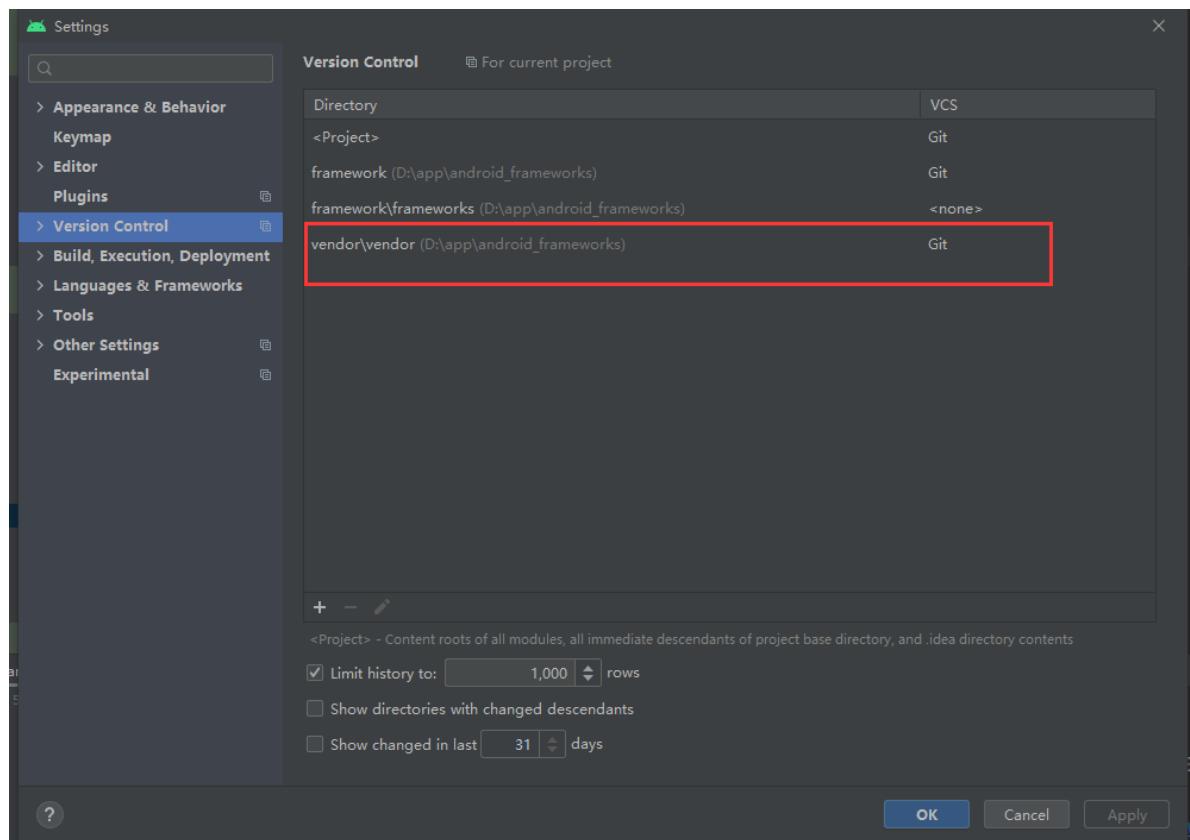
4 拉取代码

点击运行pullFrameworkCode task, 拉取服务器代码

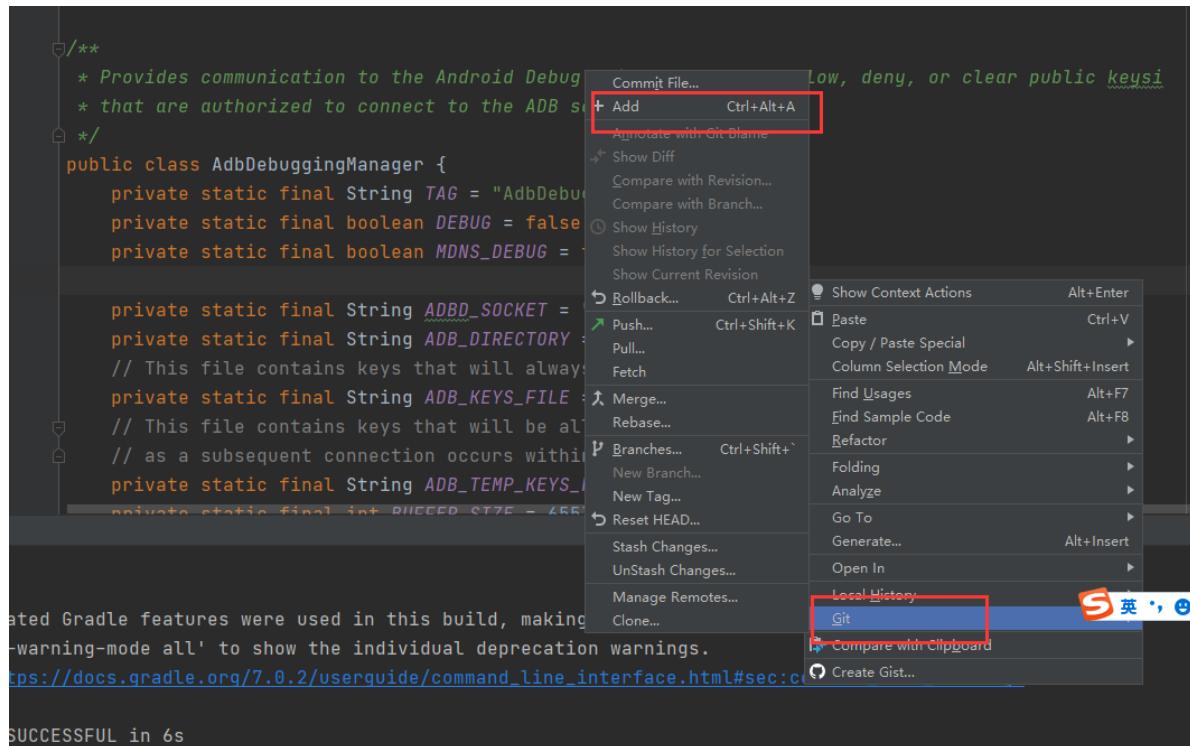
```
87  
88 > task pullFrameworkCode() {  
89     doLast [  
90         println("pullFrameworkCode")  
91         def out = new ByteArrayOutputStream()  
92         def cmd = ""  
93  
94         for (filePath in FRAMEWORK_CODE) {  
95             def scpCmd = "scp -rp " + PROJECT_PATH + filePath + "*" + filePath + ";"  
96             def directoryPath = filePath.substring(0, filePath.lastIndexOf("/"))  
97             def windowsDirectoryPath = directoryPath.replace("/", "\\\\")  
98             def file = new File(rootProject.rootDir, windowsDirectoryPath)  
99             if (!file.exists()) {  
100                 cmd += "mkdir -p " + directoryPath + ";"  
101             }  
102             cmd += scpCmd  
103         }  
104  
105         exec {  
106             ExecSpec execSpec ->  
107                 executable 'bash'  
108                 args '-c', cmd  
109                 standardOutput = out  
110         }  
111         println(out.toString())  
112     }  
113 }
```

5 提交代码

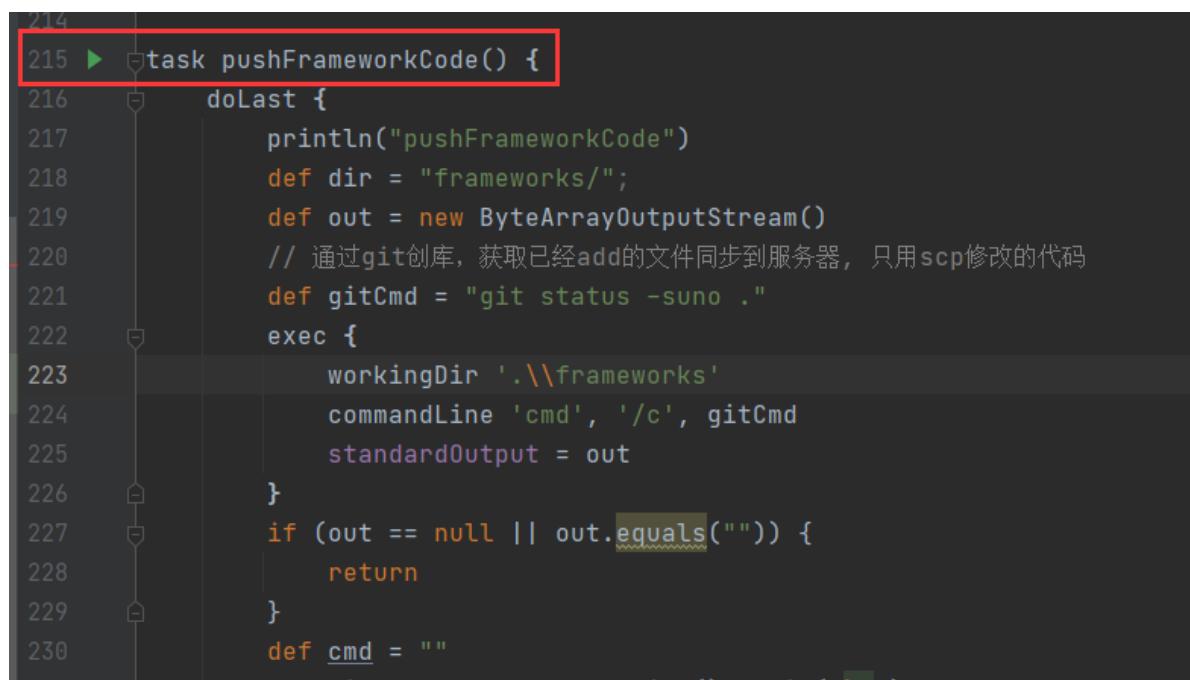
1 拉取代码的脚步会创建git创库，用来add修改的差异文件，查看vesrion control 确保创库配置对了。出错会显示红色，导致无法添加到创库，添加显示为灰色



2 修改代码后，通过git add 添加到git仓库 (add显示为灰色或者git选项不能出来，说明version control配置不对)



3 点击 pushFrameworkCode 将git add的修改复制到服务器



6 push到设备

服务器编译好后，通过pushFrameworkJar先从服务器下载jar等文件到本地，然后会自动通过adb push命令推送到对应的开发版上，重启

```
161  
162 ► task pushFrameworkJar() {  
163     dependsOn(pullFrameworkJar)  
164     doLast {  
165         println("pushFrameworkJar")  
166         def targetBoard = "${rootProject.ext.targetBoard}"  
167         def out = new ByteArrayOutputStream()  
168         def cmd = "adb root;adb remount;"  
169         for (filePath in FRAMEWORK_JARS) {  
170             def windowsFilePath = filePath.replace("/", "\\\\")  
171             def devicePath = filePath.substring(filePath.indexOf(  
172                 println(windowsFilePath + " " + devicePath)  
173                 cmd += "adb push ${windowsFilePath} ${devicePath};"  
174         }  
175  
176         cmd += "adb reboot"  
177         println(cmd)
```