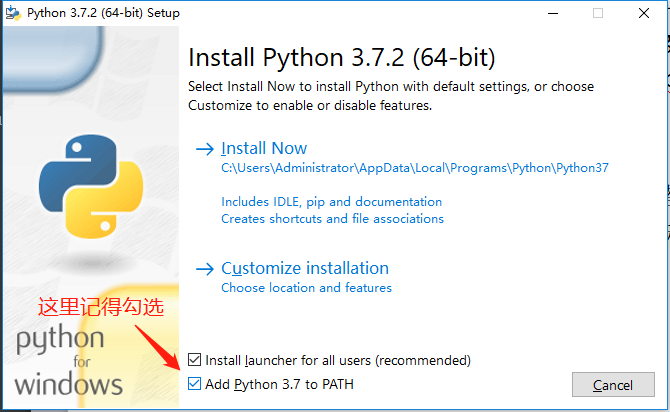
项目采用svn(公司内使用),git(shine小组成员使用)的双版本管理模式

1. 开始link部分，(首先确认你已经安装python 3 (python2不行)

下载地址https://www.python.org/downloads/ )

首先双击homeGit/core/link/link.py，来进行core内的link



1. 接下来,如果你是为个人电脑搭建git工程,看第二步

如果你是为公司电脑搭建svn工程,看第三步

等完成之后，阅读当前目录下的《新人必读.docx》

可能会遇到的问题

1.java版本低 -illegal-access=deny是java9中新增虚拟机参数



2.windows同时安装了多版本 如何选择想用的jdk版本



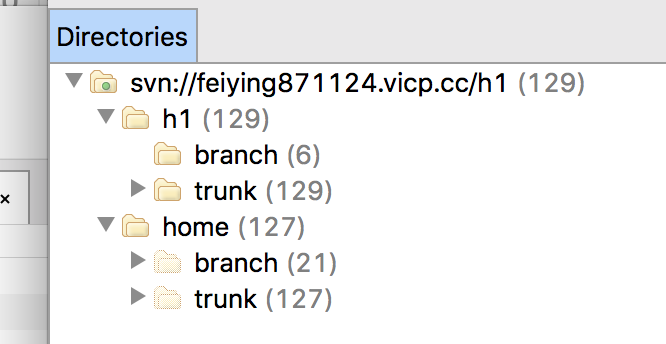
如果有新版jdk配置好的环境变量 Oracle\Java\javapath 就要把自己配置的路径放在前边

二.git工程构建

1. 打开limitlessGit/develop/link目录,双击linkCoreForGit.py，然后输入你core的目录(如E:/home/homeGit/core),回车，来完成工程的link部分
2. 如果是暂时不启用客户端C#热更的项目，再双击一下linkHotfix.py，来完成热更工程与主工程的软连接。
3. svn工程构建
4. 如果是windows先安装svn命令行工具,在core/docs/tools/Setup-Subversion-1.8.17.msi

安装后即可使用svn命令行,如是mac，则无需安装(自备了)

2. 在你公司的svn库中，创建home和你的项目(如h1,h1和上文的limmitless一个含义,只是工程名不同)的子目录,如



3. checkOut你的home工程到homeGit下,即使用git和svn同时管理该目录(注意svn本地目录的选择,别选择了子目录,要完整覆盖,(svn会弹个确认提示给你))(如果你的项目有trunk/branch的划分,只checkOut对应分支,如home/trunk)

4. 执行homeGit/core/link/ makeSvnForCore.py来构建并提交svn的home部分(每个svn库只执行一次即可,未来再新增项目无需再执行此步骤)

5. 接下来到你的项目目录，通步骤2一样，checkOut你的h1工程到testGit下

6. 如该目录曾经是git控制的，此时要执行一次deleteLink.py来删除git的软连接部分,如不是,则跳过此步骤

7. 执行makeSvn.py来构建并提交svn的test部分，需要输入home/core所在的svn路径,(注意是svn版本库路径,并且必须和该子项目在同一个svn版本库),(在这里例子里home的svn路径为home/trunk/core),(每次新创项目时(初次上传svn),执行一次即可)

8. 接下来执行linkClientSvn.py来创建本地项目软连接(每个项目组成员,(程序,策划,测试)在初次checkOut项目之后都需要执行此步骤)

9. 如果是暂时不启用客户端C#热更的项目，再双击一下linkHotfix.py，来完成热更工程与主工程的软连接。

10. shine小组成员如需要开发工具工程,则执行linkCoreForSvn.py,输入home/core所在的本地目录(就是上面的homeGit目录)

11. 接下来让你项目的每个成员直接checkOut你的项目svn地址即可(最好给svn的home目录设置只读权限,除了作为主程的你自己)

12.更新与提交。

svn->git:

1. 如只修改了h1工程的内容，直接svn 更新然后git提交并推送即可
2. 如果修改了home的内容,先提交svn，再到homeGit目录下svn更新,再git提交推送(即home的内容需要到homeGit目录下提交)

git->svn:

1. 如只修改h1工程,也是直接git更新，再svn提交即可
2. 如果修改了home的内容,先在homeGit目录更新git，然后提交svn，再到项目目录(h1Git)svn更新即可