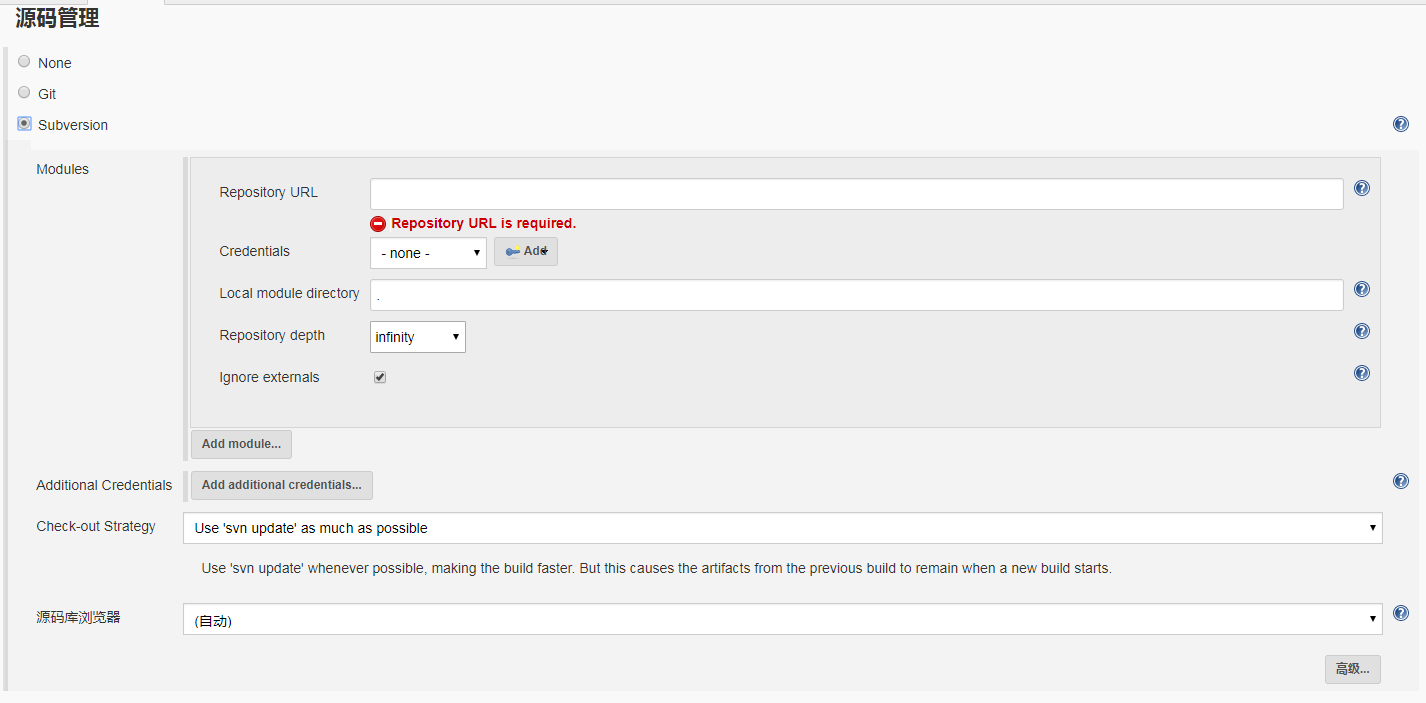
1. SVN插件：

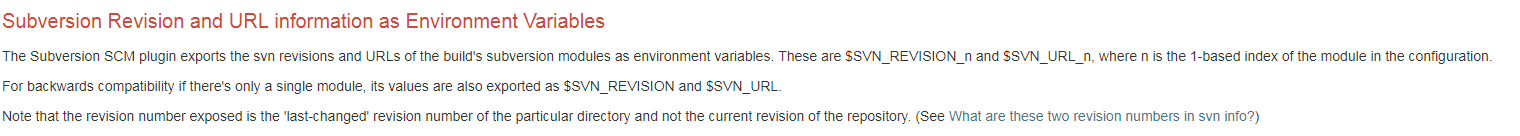
Jenkins中对svn进行操作，可通过插件和脚本两种方式进行。

* 插件方式

　　在插件管理中安装"Subversion Plug-in"，即可在任务的"源码管理"中配置SVN。（[插件官方说明地址](https://wiki.jenkins.io/display/JENKINS/Subversion+Plugin)）



* + Module：指svn上的模块。一个任务中可以添加多个不同来源的svn模块。
  + Repository URL：svn代码仓库的地址。
  + Credentials：访问svn代码仓库所需的账号密码。
  + Local module directory：svn检出到本机的文件夹路径
  + Repository depth：需要检出的文件夹深度，一般设为infinity（配置文件夹下的所有文件，包括子文件夹）具体说明可见插件帮助。
  + Ignore externals / Additional Credentials：未用到，不作说明。
  + Check-out Strategy：更新svn到本地的几种方式。建议使用 Use 'svn update' as much as possible, with 'svn revert' before update (先放弃所有本地修改，然后尽可能进行svn更新。避免本地修改导致本地项目与代码仓库不一致）
  + 通过插件进行svn更新的情况，插件可提供若干参数（svn版本号及svn路径），供其它shell或插件使用，可参考官方文档（注意当配置了多个模块时的参数名命名方式）：



* 脚本方式(Shell)
  + 脚本方式下，只需了解svn所支持的命令列表，即可在shell中自行配置。其自由度相比插件更高，可以方便地对特殊需求进行处理。
  + 常用命令包括：
    - 检出  svn checkout
    - 更新  svn update
    - 取消本地修改 svn revert
    - 清理本地项目 svn cleanup
    - 向代码仓库新增文件 svn add
    - 提交到代码仓库  svn commit
  + 脚本方式下取得svn 版本号可以通过shell的sed命令：

|  |  |
| --- | --- |
| 1 | build\_svn\_version=`svnversion  ${clientPath} |sed 's/^.\*://' |sed 's/[A-Z]\*$//'` |

　　　　上述语句中${clientPath}为本地svn项目的根目录（即包含了.svn隐藏文件夹的目录），build\_svn\_version存放了取出的svn版本号。

1. GIT

Github Plugin：要与GitHub持续集成的时候，需要安装的插件。

findbugs代码走查  
使用方式：   
目的：进行代码走查的自动化，能够提示垃圾代码或者提供代码优化的建议   
1.首先下载findbugs   
http://findbugs.sourceforge.net/downloads.html   
2.其次在ant中配置findbugs   
http://www.51testing.com/?uid-265524-action-viewspace-itemid-200942   
3.再次在jenkins内下载findbugs插件   
无需配置，如果是ant调用会在项目内自动生成findbugs.xml文件并且在jenkins在绘制findbugs曲线   
emma行测试代码覆盖率检查   
使用方式：   
目的：进行测试代码覆盖率检查   
如果使用maven那么只要指定target目标位emma:emma即可   
如果使用ant那么请继续看   
1.首先下载emma   
http://emma.sourceforge.net/downloads.html   
2.其次在ant中配置emma   
一个ant实用例子http://ouzhong.blog.hexun.com/32369273\_d.html   
ibm ant-emma配置：http://www.ibm.com/developerworks/cn/java/j-lo-emma/   
3.下载emma-jenkins插件   
https://wiki.jenkins-ci.org/display/JENKINS/Emma+Plugin   
  
其他做简单介绍需要的时候可以使用：   
1.ant插件   
2.disk usage plugin  这个插件可以监控每次build项目所花费的物理空间   
3.thinbackup 这个插件可以备份你job的配置   
4.jenkins workspace cleanup plugin 这个插件可以再每次build之前清空workspace   
5.build-name-setter 这个插件可以设置每次build显示的名字   
6.git插件    
7.PMD插件 这个插件   
8.python plugin 这个插件支持你像写shell一样写python代码   
9.email-ext plugin 这个插件提供你发送HTML格式的邮件   
10.hudson next build number plugin 如果你想把hudson每次build的版本号放到你的产品版本号中你可以使用这个插件   
11.HTML publisher plugin 支持HTML格式的report   
12.android lint plugin  支持android项目检查   
13.Copy Artifact Plugin  支持把build后产生的结果copy到其他项目去   
14.promoted 支持手工测试验证，或者发布产品，在build结束后手动执行   
15.Jenkins description setter plugin  这个插件可以支持在Job的描述修改，例如二维码   
16.Hudson Port Allocator Plug-in  开启关闭端口，或者选择一个自由的端口来进行build   
17.Android Emulator Plugin   
18.JobConfigHistory Plugin  这个插件用于查看jobconfig历史记录   
19.Jenkins Sounds plugin  这个插件用于结束build后会给出声音提示   
20.JIRA Plugin  这个插件用来和Jira配合使用