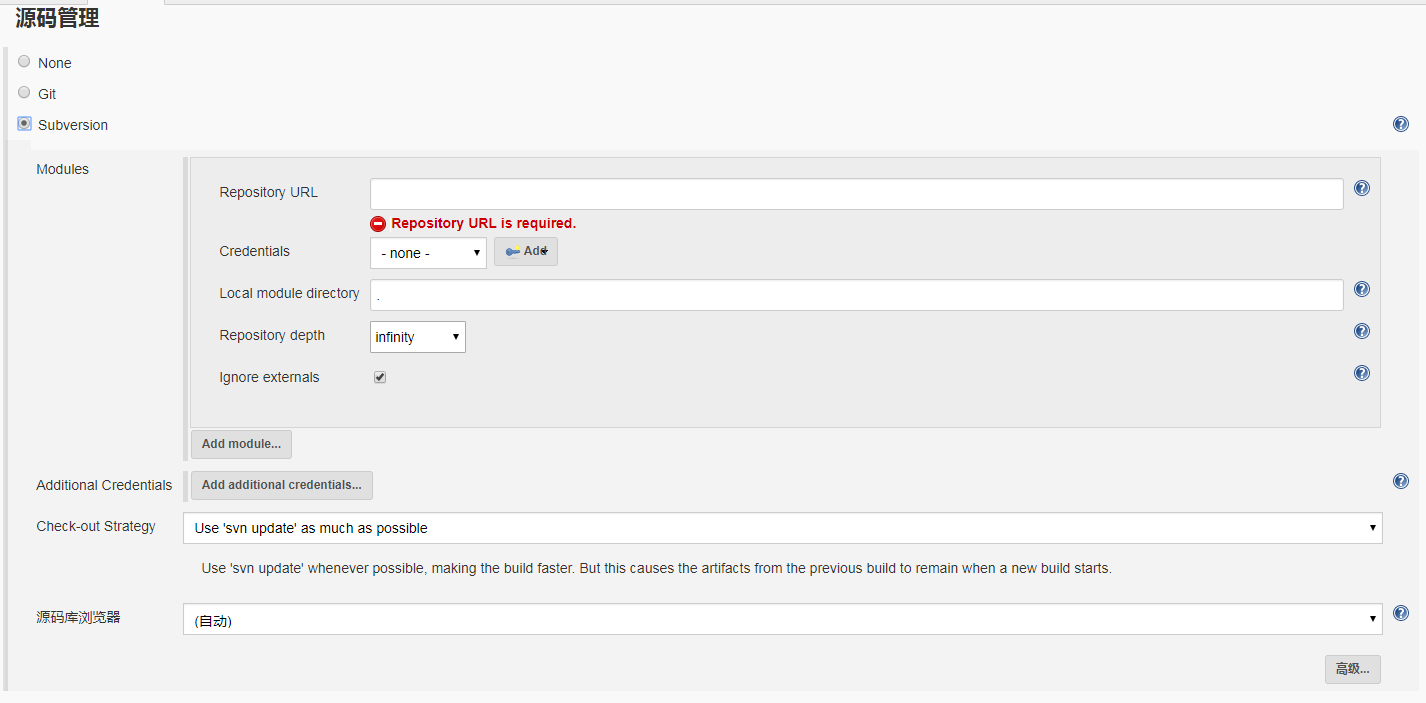
1. SVN插件：

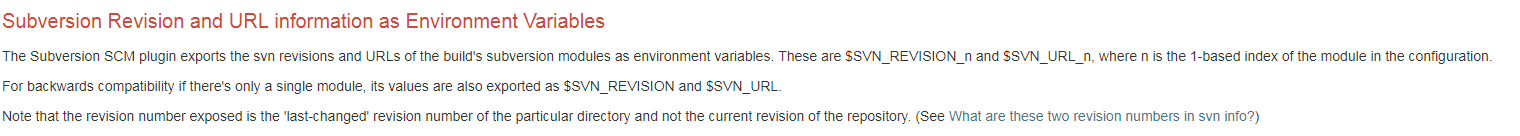
Jenkins中对svn进行操作，可通过插件和脚本两种方式进行。

* 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。