

NSP 开发配置指南

编写: _____

校对: _____

审核: _____ 年 ____ 月 ____ 日

标审: _____ 年 ____ 月 ____ 日

批准: _____ 年 ____ 月 ____ 日

郑州新开普电子股份有限公司

修订记录

日期	修订版本	描述	作者	审核人

分发记录

Copy No.	持有者和角色	分发日期

目录

1	开发环境搭建.....	4
1.1	开发工具.....	4
1.2	JDK 的安装和设置.....	4
1.2.1	JDK 的安装.....	4
1.2.2	设置环境变量.....	5
1.2.3	Tomcat 安装.....	7
1.2.4	Eclipse 安装.....	7
1.2.5	必备插件安装.....	8
2	SVN 版本库.....	27
2.1	访问路径.....	27
2.2	Eclipse 访问.....	28
3	创建工程.....	30
3.1	Check Out 检出代码.....	30
3.1.1	Bizplatform 服务开发人员建立工程.....	30
3.2	依赖管理.....	31
3.2.1	添加 Jar.....	31
3.2.2	项目间依赖.....	32

1 开发环境搭建

1.1 开发工具

FTP 下载地址: <ftp://192.168.0.66/pub>

1.2 JDK 的安装和设置

1.2.1 JDK 的安装

双击 jdk-6u22-windows-i586.exe

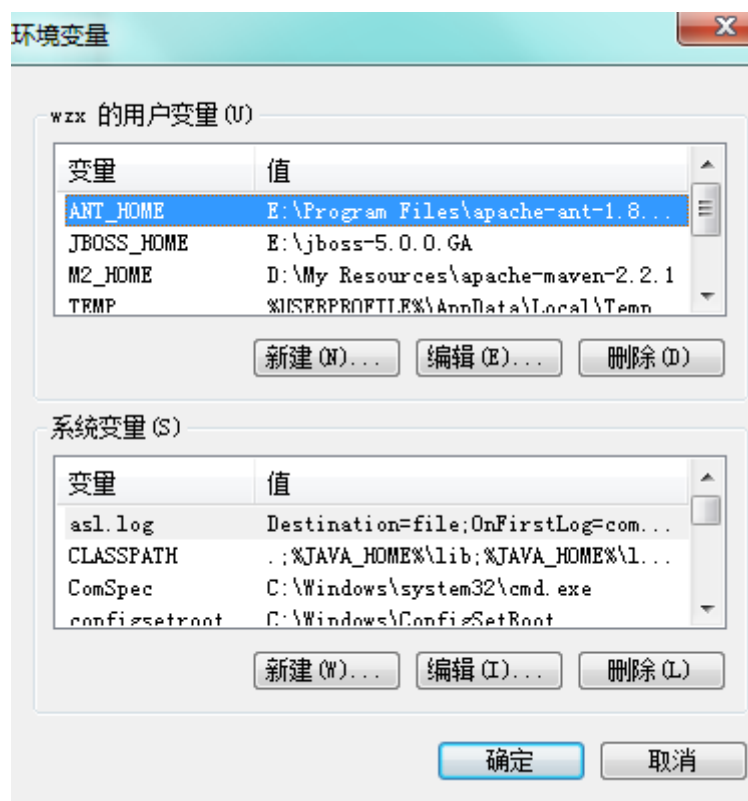




点击完成。JRE 的安装与上述过程相同。

1.2.2 设置环境变量

我的电脑，右键→属性→高级→环境变量



在“系统变量”中设置三个变量，JAVA_HOME,PATH,CLASSPATH（大小写无所谓），如果已存在则点击“编辑”，如果不存在则点击“新建”。

- JAVA_HOME:指定 JDK 的安装目录，如 **E:\Java\jdk1.6.0_22**
- CLASSPATH:指定 Java 加载类路径，设置
为 **.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\tools.jar**
- PATH:该变量可使得系统在任何路径下都能识别 Java 命令，在原有值前面加入 **%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin**

验证：“开始” → “运行”，输入“cmd”，然后回车，在命令行窗口中键入“java -version”，显示如图内容，表示设置成功。



1.2.3 Tomcat 安装

双击 apache-tomcat-6.0.29.exe，按照提示点击下一步即可。

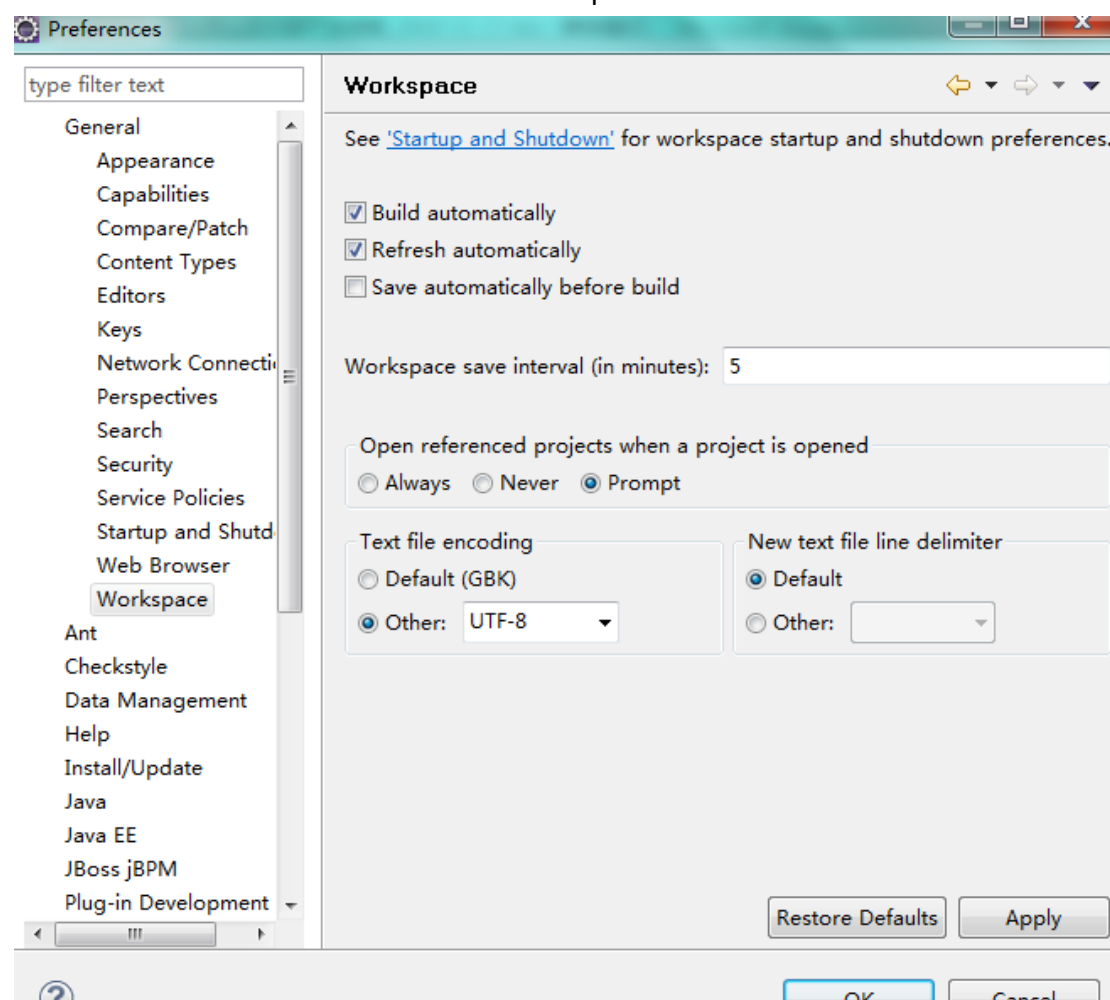
注：在选择 tomcat 的安装根目录时，最好选择安装在某个分区的根目录下，且其安装路径中最好不要有空格。

1.2.4 Eclipse 安装

直接将“eclipse3.5-jee-galileo-win32.zip”解压缩即可。

修改工作空间默认字符集：

Window→Peferences→General→Worspace



将工作空间默认字符集改为 UTF-8，然后点击 Apply，最后点击 OK 即可。

1.2.5 必备插件安装

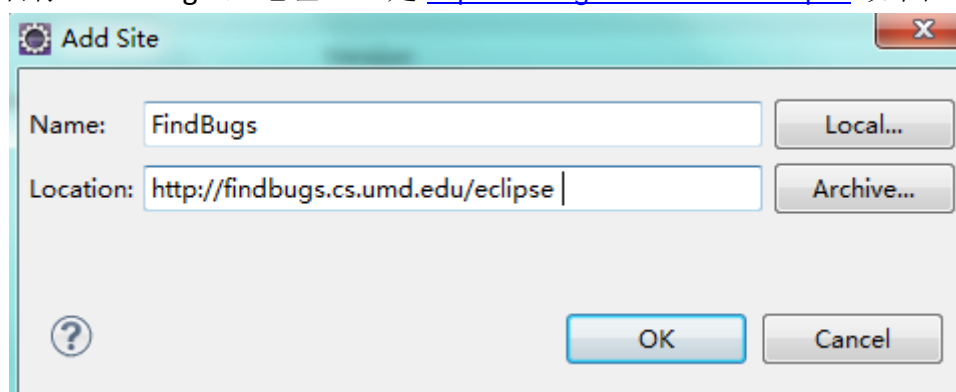
1.2.5.1 FindBugs 代码检测工具

FindBugs 是一个可以在 Java 程序中发现 Bugs 的程序。它是专门用来寻找处于“Bug Patterns”列表中的代码的。Bug Patterns 指很有可能是错误的代码的实例。

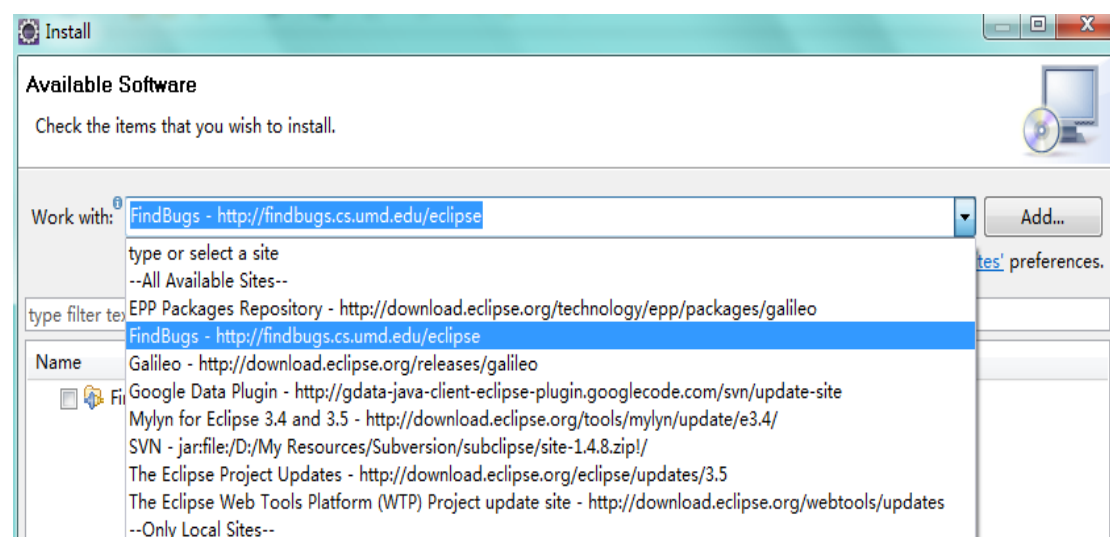
第一步：下载并安装插件

启动 Eclipse Help→Install new SoftWare→add

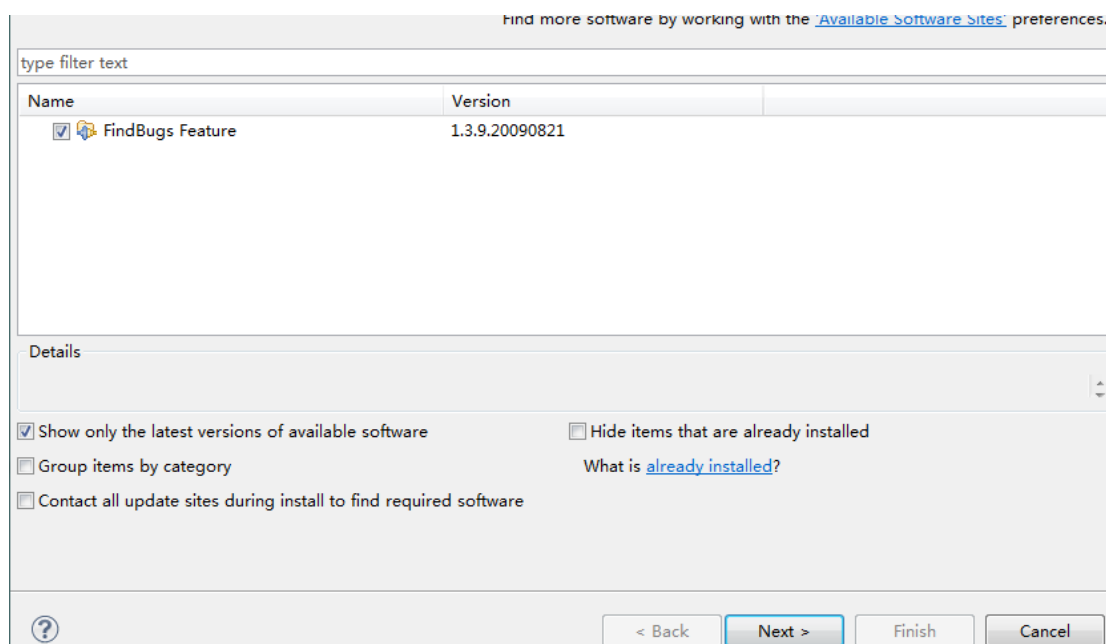
名称: “FindBugs”，地址 URL 是 <http://findbugs.cs.umd.edu/eclipse> 如图:



点击“OK”。接着在“Work with”的下拉框中选择刚刚添加的该插件，如下图:



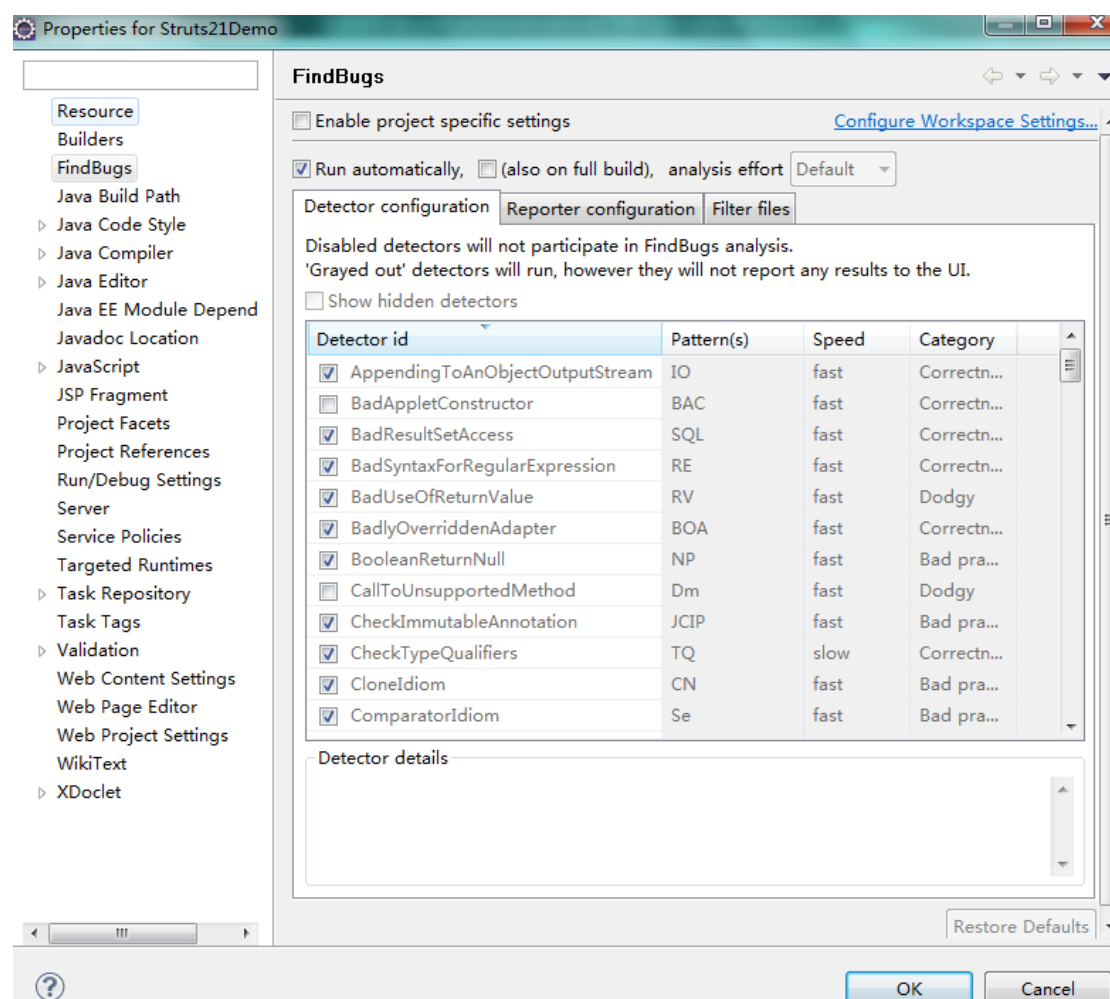
然后按照下图所示进行勾选，并且一直下一步，下载完成后提示是否重启，点击“**Yes**”重启 Eclipse 即可。



注：此时务必取消掉“Group items by category”和“Contact all update sites during install to find required software”这两项的勾选，否则会造成一些麻烦。

第二步：应用

在左侧“Project Explorer”中选中要应用 FindBugs 的项目，右击→Properties，出现如图对话框



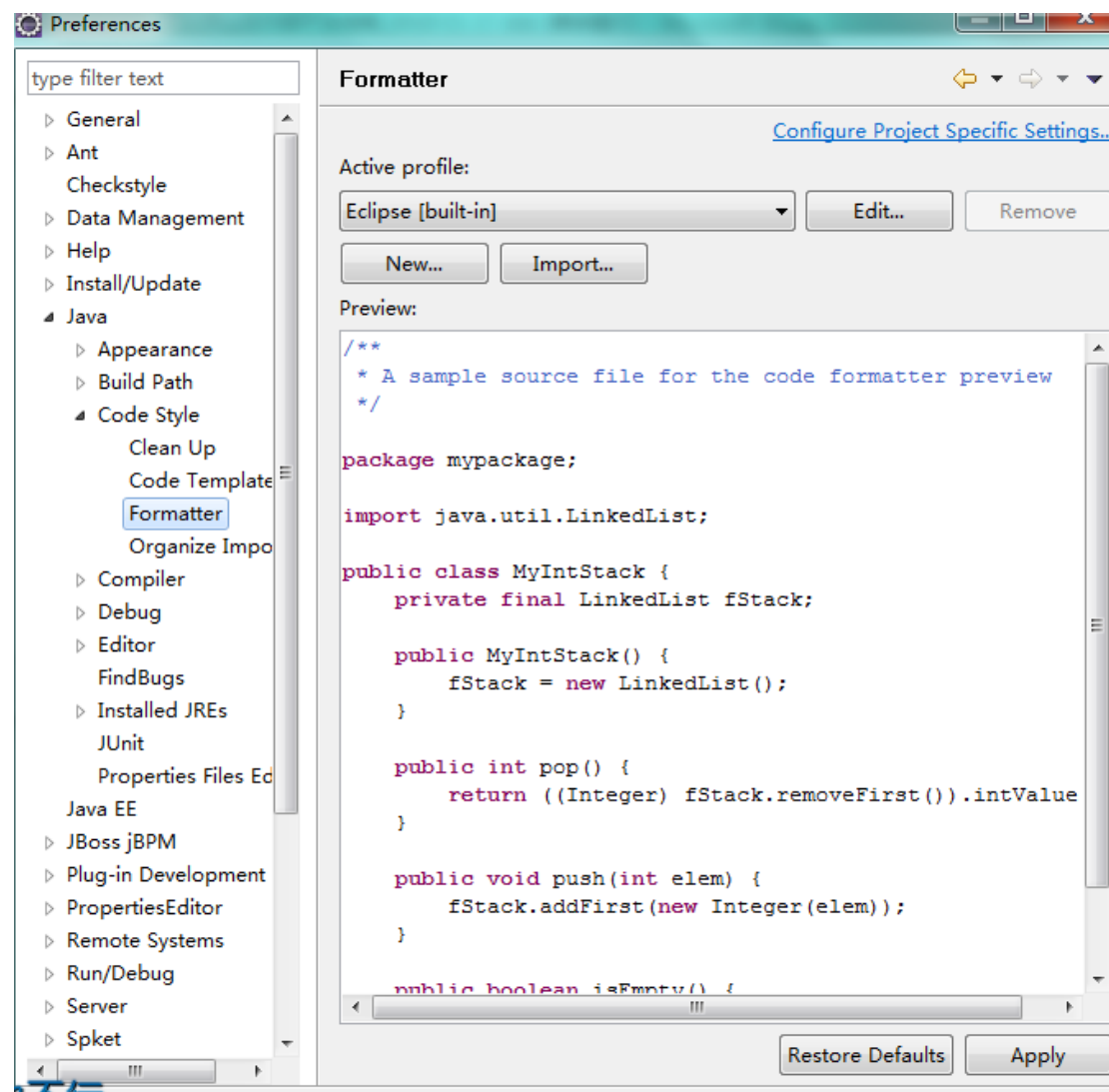
勾选上方的“Run automatically”这样在项目编译时就可自动执行 findbugs。

1.2.5.2 CheckStyle 代码样式检测工具

Checkstyle 是一款代码格式检查工具它可以根据设置好的编码规则来检查代码 比如符合规范的变量命名，良好的程序风格等等。

注：在用此插件之前请先导入 java 格式化文件：NSPEclipseStyle.xml

Window→Preferences→Java→Code Style→Formatter



点击 Import 导入 NSPEclipseStyle.xml

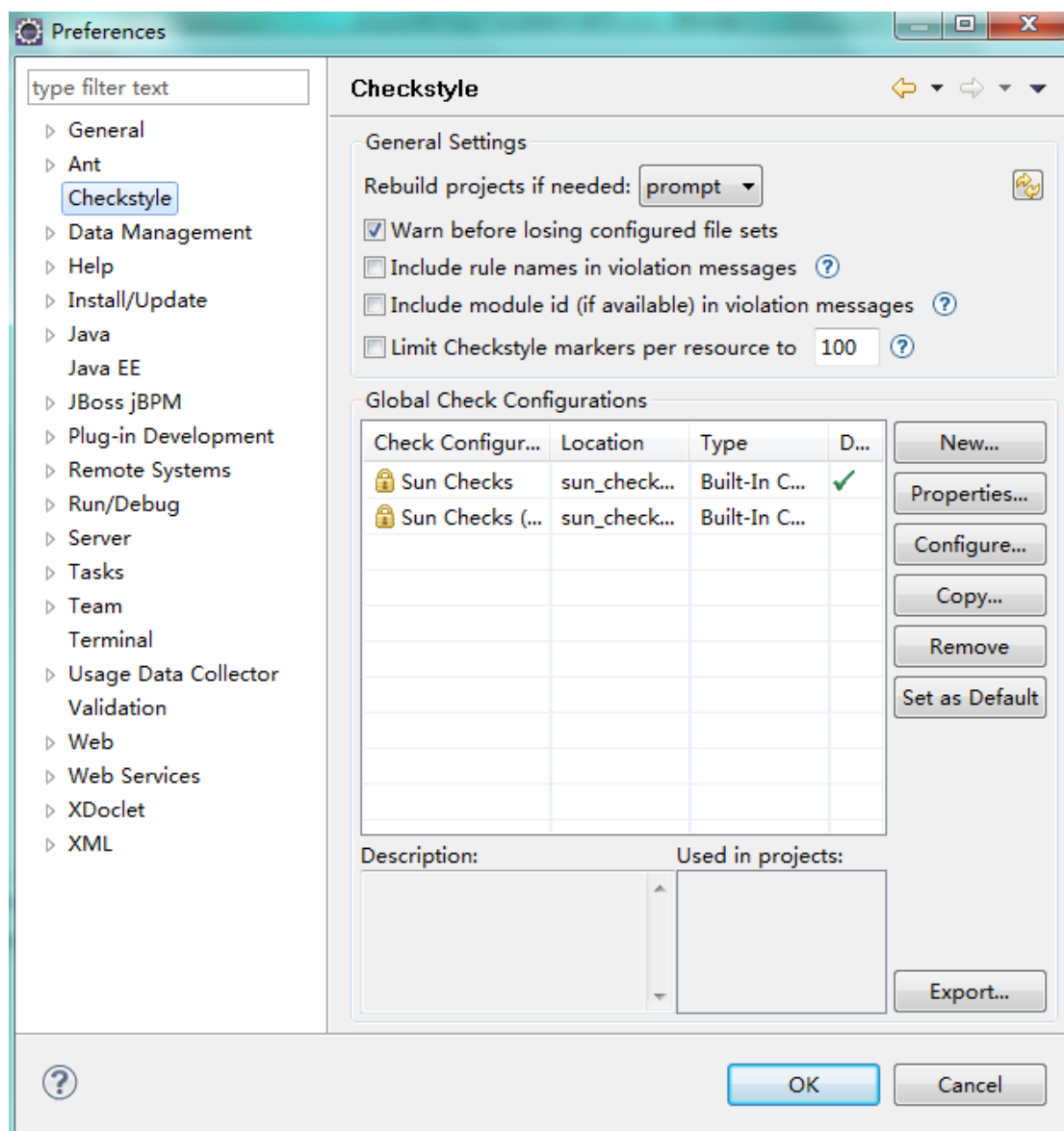
第一步：下载并安装插件

安装步骤参照 FindBugs 第一步。

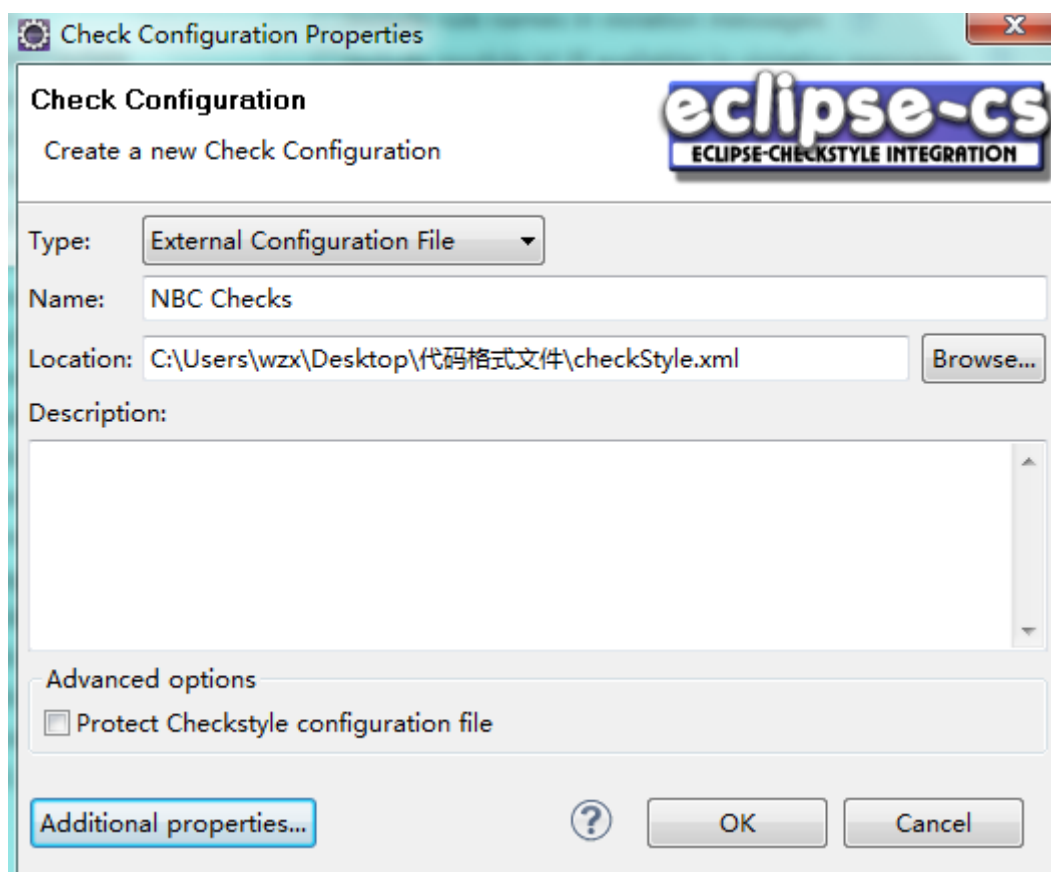
名称：“Checkstyle” 下载地址 URL 是 <http://eclipse-cs.sourceforge.net/update>

第二步：应用

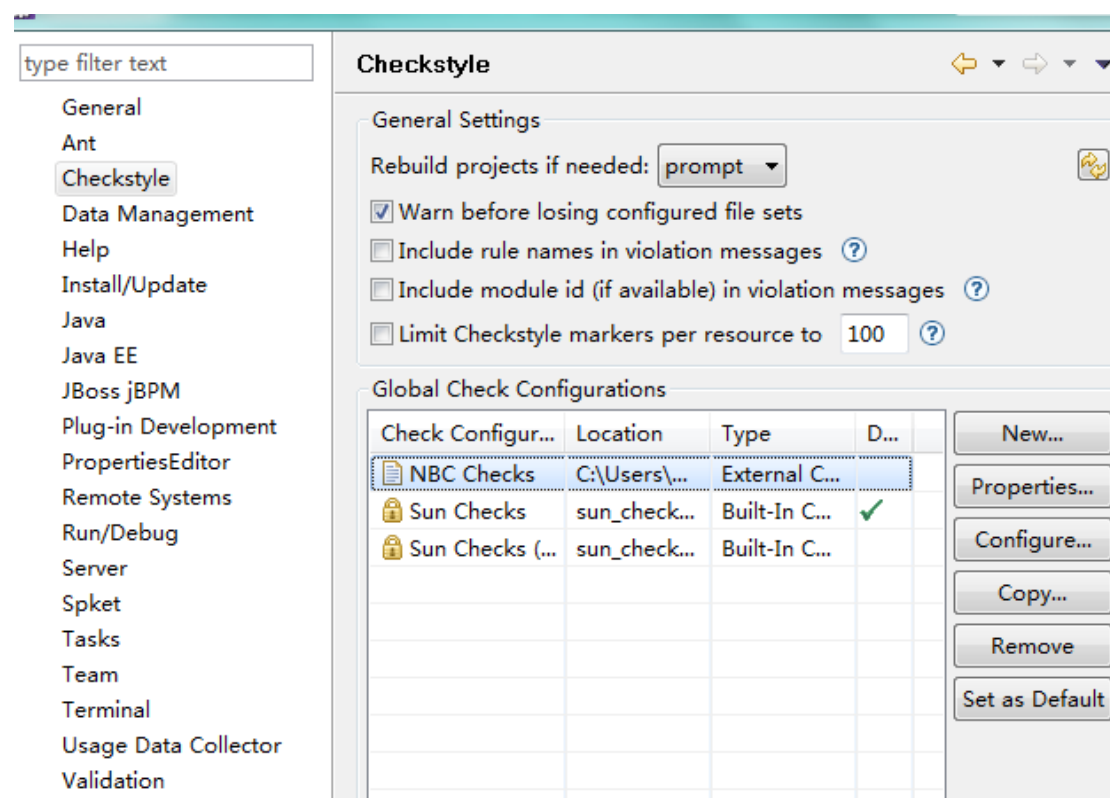
安装完成后 Window→preferences，左侧会出现 Checkstyle 选项如图：



点击页面中的 new



如上图所示，在 **Type** 中选择 **External Configuration File**，**Name** 填写配置的名称，点击 **Browse** 找到 **checkStyle.xml** 文件，最后点击 **OK**。



选中刚刚添加的配置项，然后点击 Set as Default，最后点击 OK 完成。

在想要使用 CheckStyle 来检查的工程上右键 → Checkstyle → Activate Checkstyle 即可。

1.2.5.3 Properties Editor 属性文件编辑插件

Properties Editor 是什么？

Properties Editor 是一个便利的属性文件编辑插件。它可以直接打开.properties 文件并保存为 Unicode 格式，省去了使用 native2ascii 进行转换的麻烦。

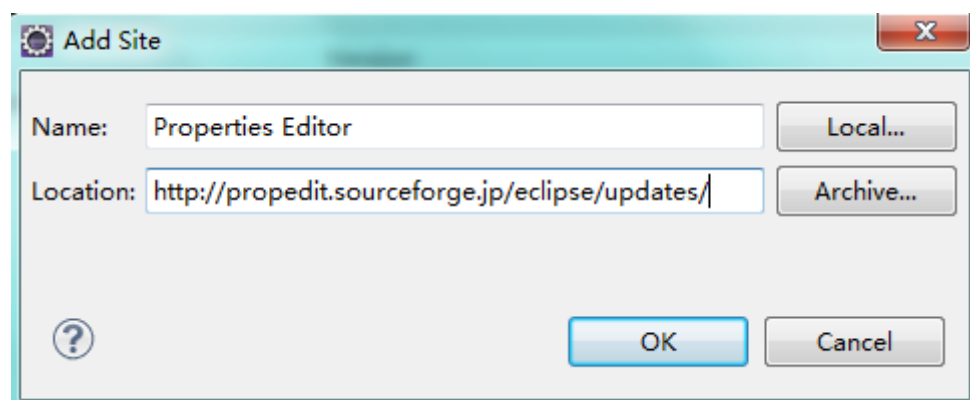
如何安装：

第一步：下载并安装插件

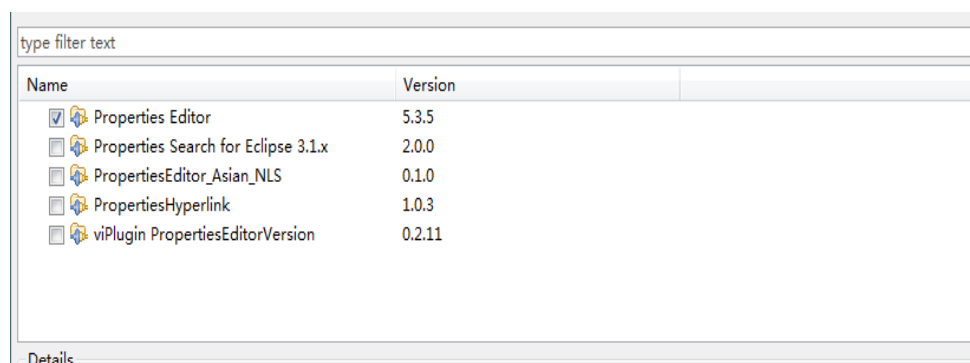
启动 Eclipse Help → Install new SoftWare → add

名称: “Properties Editor”，地址 URL 是

<http://propedit.sourceforge.jp/eclipse/updates/>，如图所示：



选择如下图所示的更新，



接下来步骤参照 FindBugs 的第一步即可。

第二步：应用

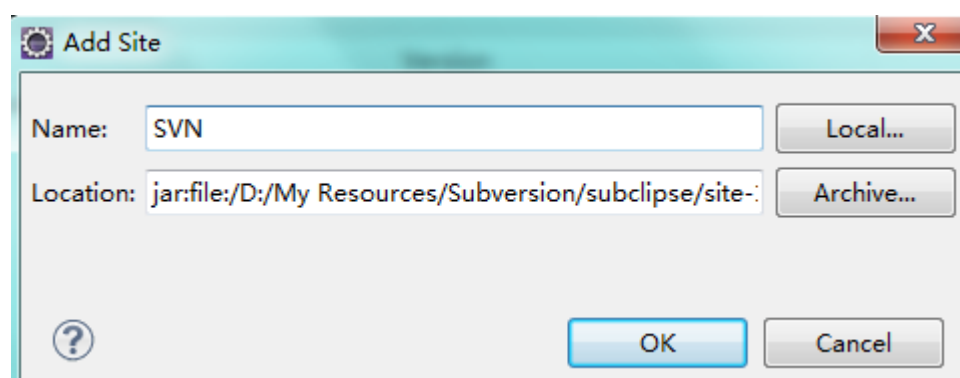
直接双击后缀名为 `properties` 的文件就默认会使用 `Properties Editor` 打开。

1.2.5.4 Subclipse 版本控制工具

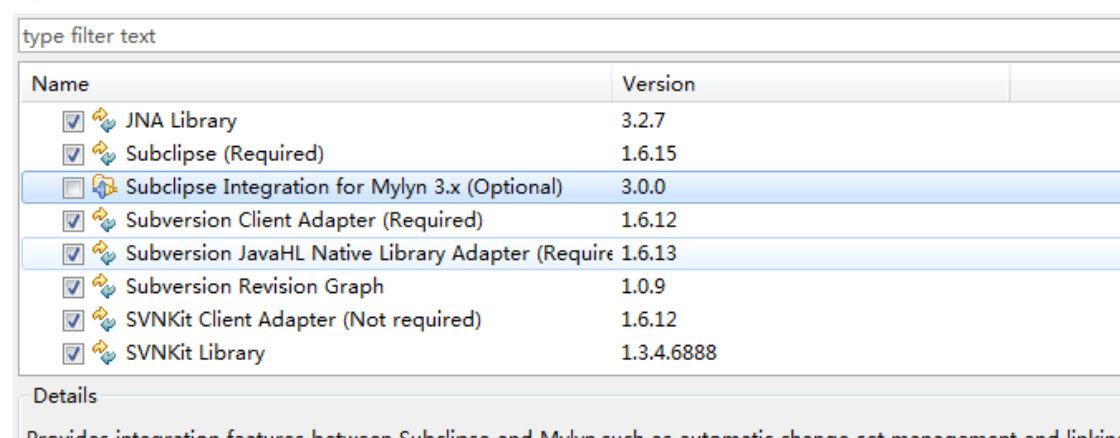
Subclipse 是一个为 Eclipse IDE 添加 Subversion 支持的项目。支持几乎所有版本的 Eclipse。

第一步：安装

Help→Install New Software→Add→Archive, 然后找到 `site-1.6.15.zip` 的存放路径, 名称 `SVN`, 如下图所示



点击 `OK`, 除 `Subclipse Integration for Mylyn 3.x (Optional) 3.0.0` 外, 其余的全部选中, 如图所示



接下来一直点击下一步即可。

第二步：应用

Window→Open Perspective→Other→SVN 资源库研究, 点击 `OK`。打开 `SVN` 透视图后在左侧右键→新建→资源库位置, 在 `url` 文本框中输入相应的地址即可, 如下图所示。



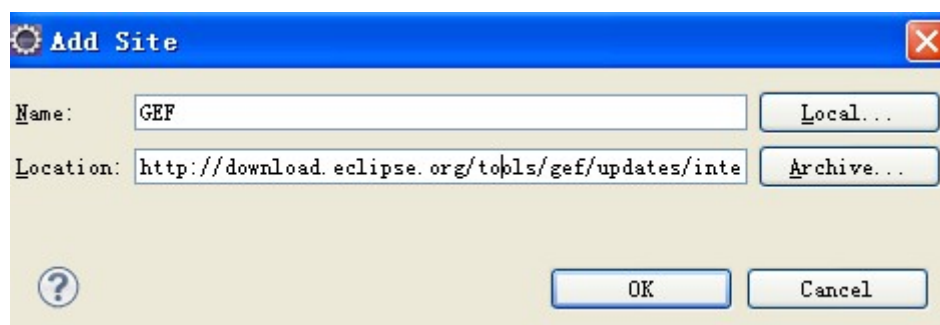
1.2.5.5 m2eclipse—Maven 插件

M2eclipse 是 Maven 在 Eclipse 中的插件可以用来管理依赖、插件,进行 Maven 项目的构建等。

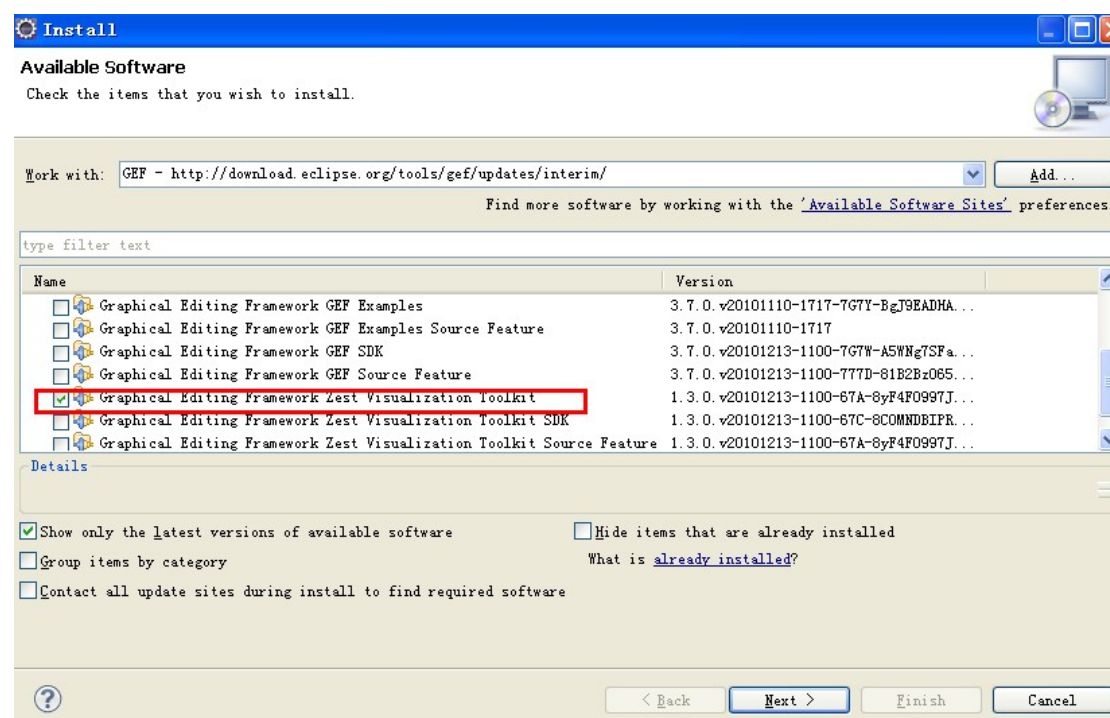
第一步: 安装

m2eclipse 的运行需要 GEF 插件的支持,所以安装 m2eclipse 之前要先安装 GEF 插件。

- a) GEF 插件: Help→Install New Software→Add, 名称: “GEF”, 地址 URL 是 <http://download.eclipse.org/tools/gef/updates/interim> , 如图所示:

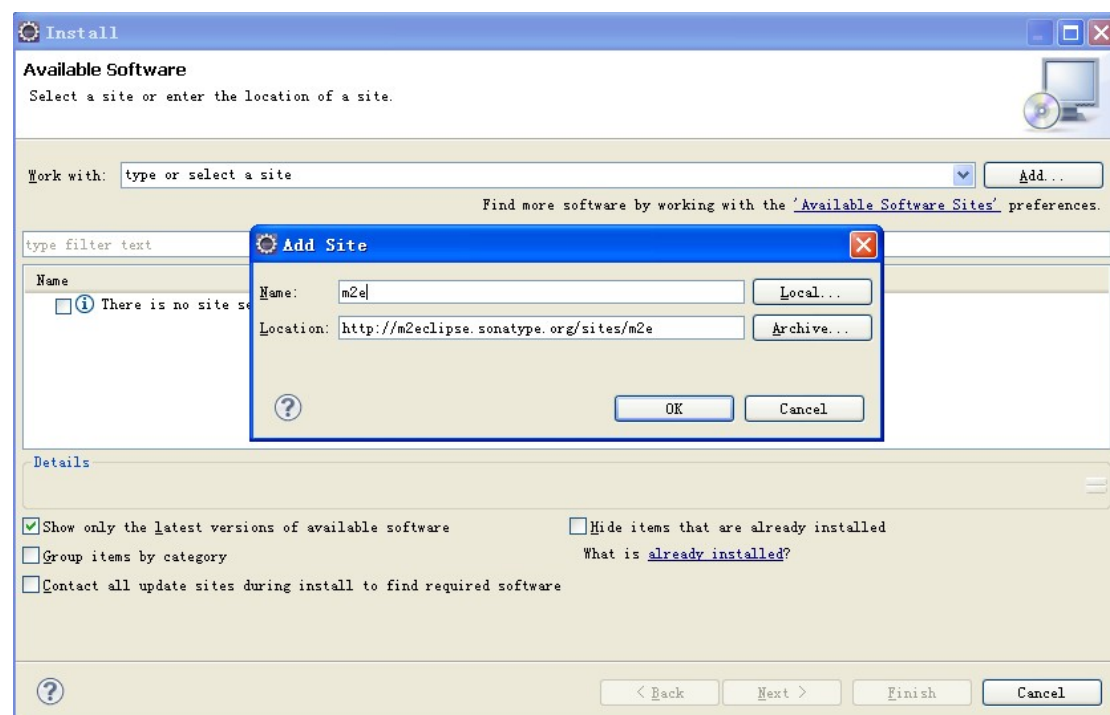


选择如下图所示的更新

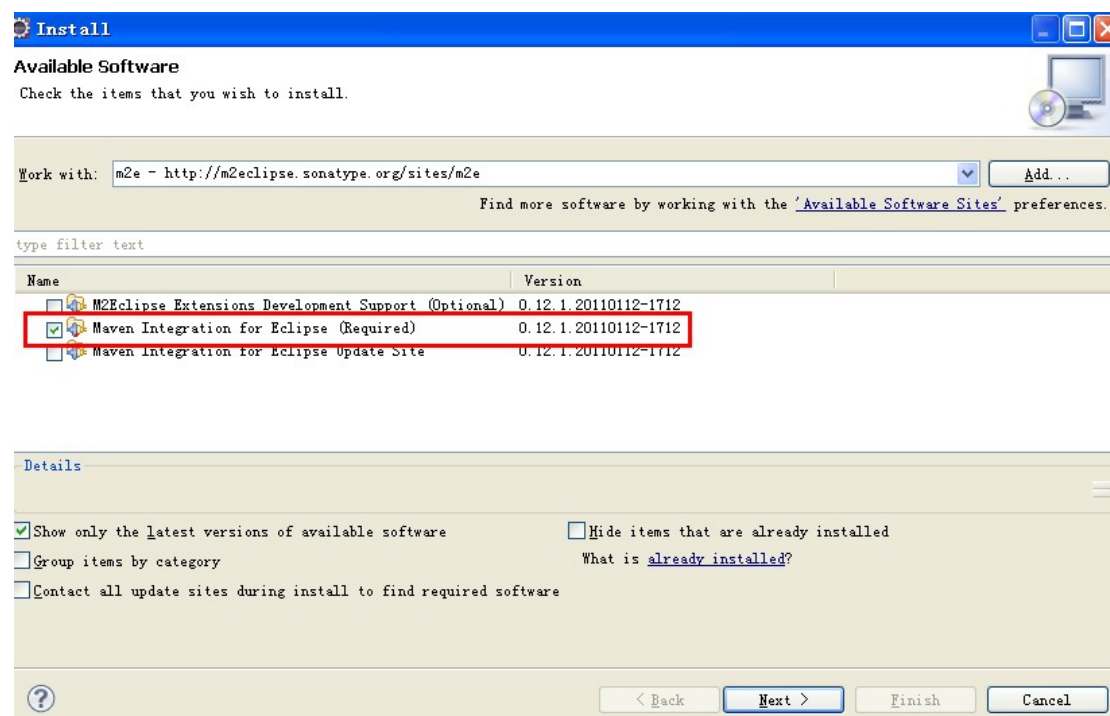


接下来一直下一步即可。

- b) m2eclipse 插件: Help→Install New Software→Add, 名称: “m2e”, 地址 URL 是 <http://m2eclipse.sonatype.org/sites/m2e> , 如图所示:



选择如下图所示的更新

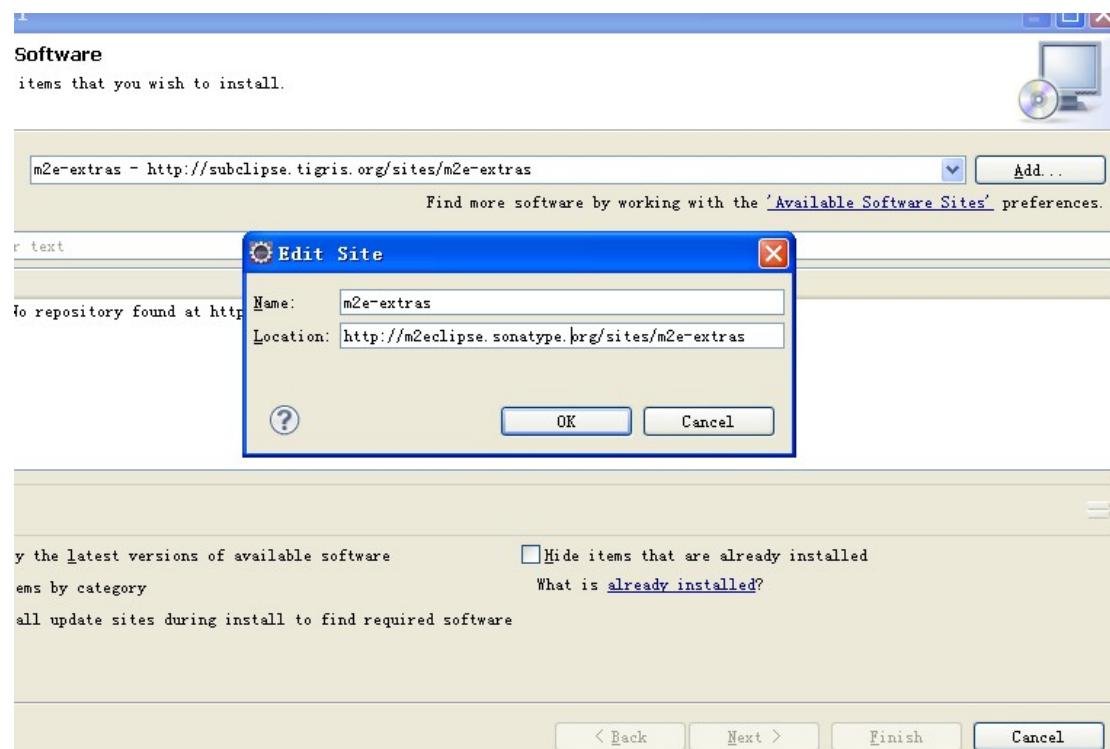


接下来一直点击下一步即可。由于需要和 SVN 做结合所以还需要安装 Maven SCM 相关的插件。

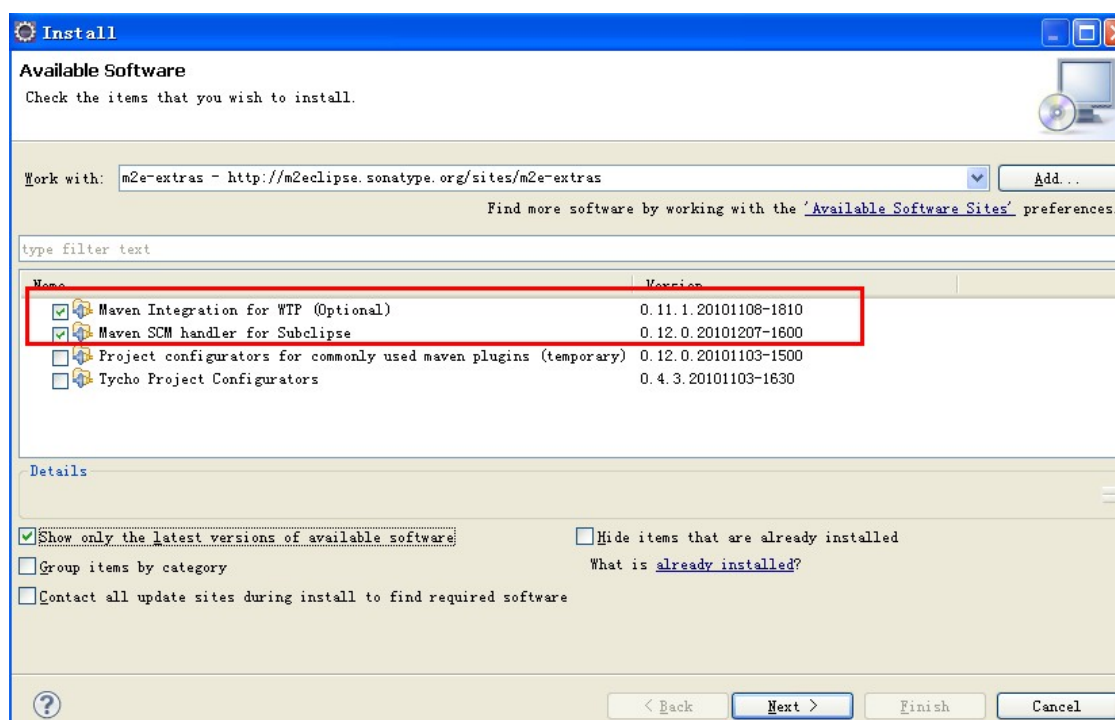
c) m2e-extras 插件: Help→Install New Software→Add, 名称:

“m2e-extras”, 地址 URL 是

<http://m2eclipse.sonatype.org/sites/m2e-extras> , 如图所示:



选择如下图所示的更新



最后一直点击下一步，重启 Eclipse 完成。

第二步：配置

在自己本地新建名为“settings.xml”的文件，将以下内容拷贝到该文件中

```
<?xml version="1.0" encoding="UTF-8"?>
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
          xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
          xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <localRepository></localRepository>

  <pluginGroups></pluginGroups>

  <proxies></proxies>

  <servers></servers>

  <mirrors>
    <mirror>
      <id>nexus</id>
      <mirrorOf>*</mirrorOf>
      <url>http://192.168.0.65/nexus/content/groups/newcape</url>
    </mirror>
  </mirrors>
```

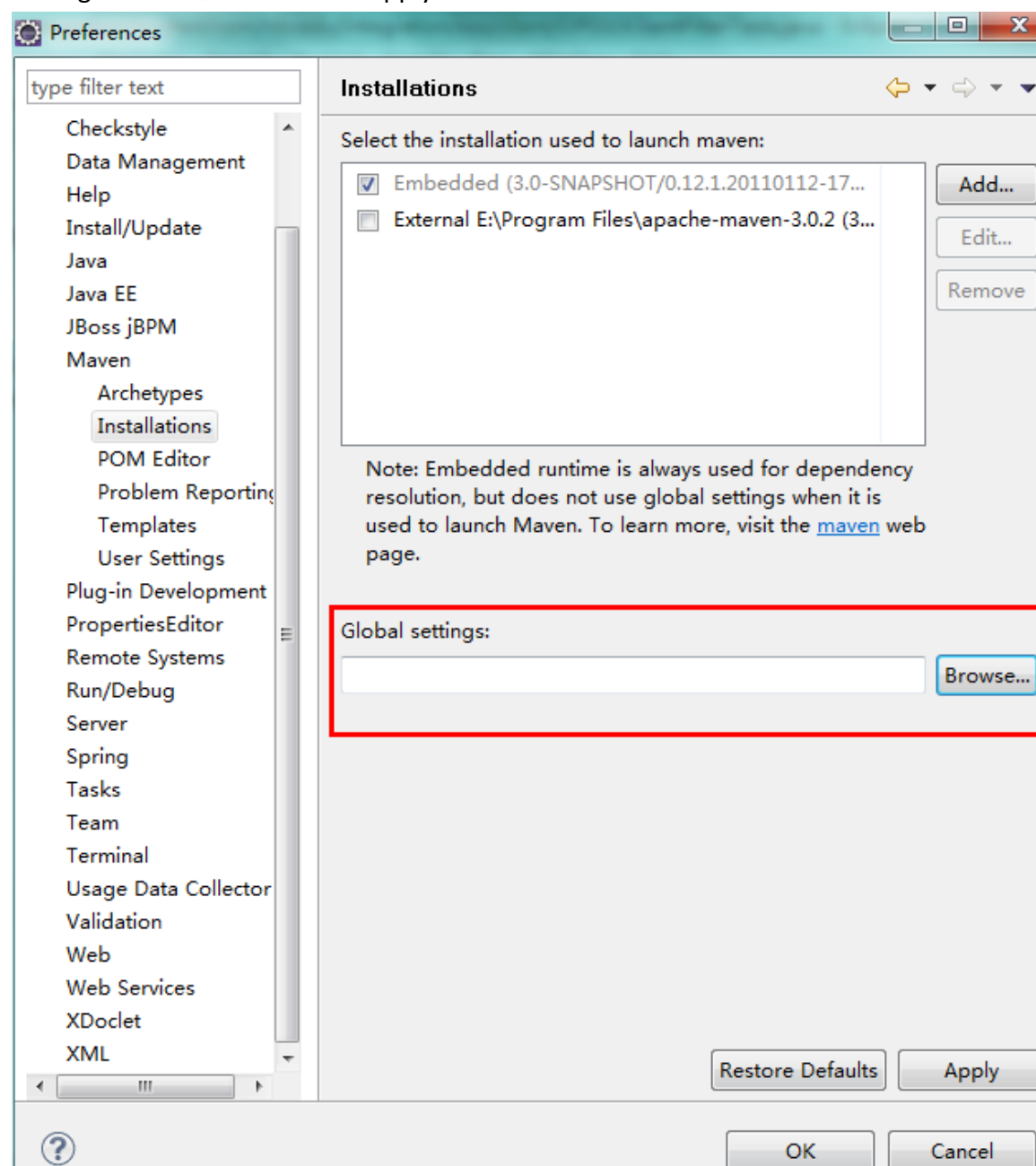
```
<profiles>
  <profile>
    <id> development</id>
    <repositories>
      <repository>
        <id>central</id>
        <url>http://central</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </repository>
    </repositories>
    <pluginRepositories>
      <pluginRepository>
        <id>central</id>
        <url>http://central</url>
        <releases>
          <enabled>true</enabled>
        </releases>
        <snapshots>
          <enabled>true</enabled>
        </snapshots>
      </pluginRepository>
    </pluginRepositories>
  </profile>
</profiles>

<activeProfiles>
  <activeProfile> development</activeProfile>
</activeProfiles>
</settings>
```

通过以上配置使得 m2eclipse 不从中央仓库下载构件，只从我们配置的私服来下载，默认情况下 Maven 会在当前用户的目录下建立如：~\.m2\repository 目录来存放下载的构件，例如：C:\Users\{用户名}\.m2\repository（win7），如果不想过多占用 C 盘空间，可修改上述配置中的本地仓库位置，例如

```
<localRepository>{自定义本地仓库存储的位置}</localRepository>
```

建立好配置文件后在 Eclipse 中，Window→Preferences→Maven→Installations，如下图所示浏览找到刚刚创建的 settings.xml 文件，最后点击 Apply 即可。

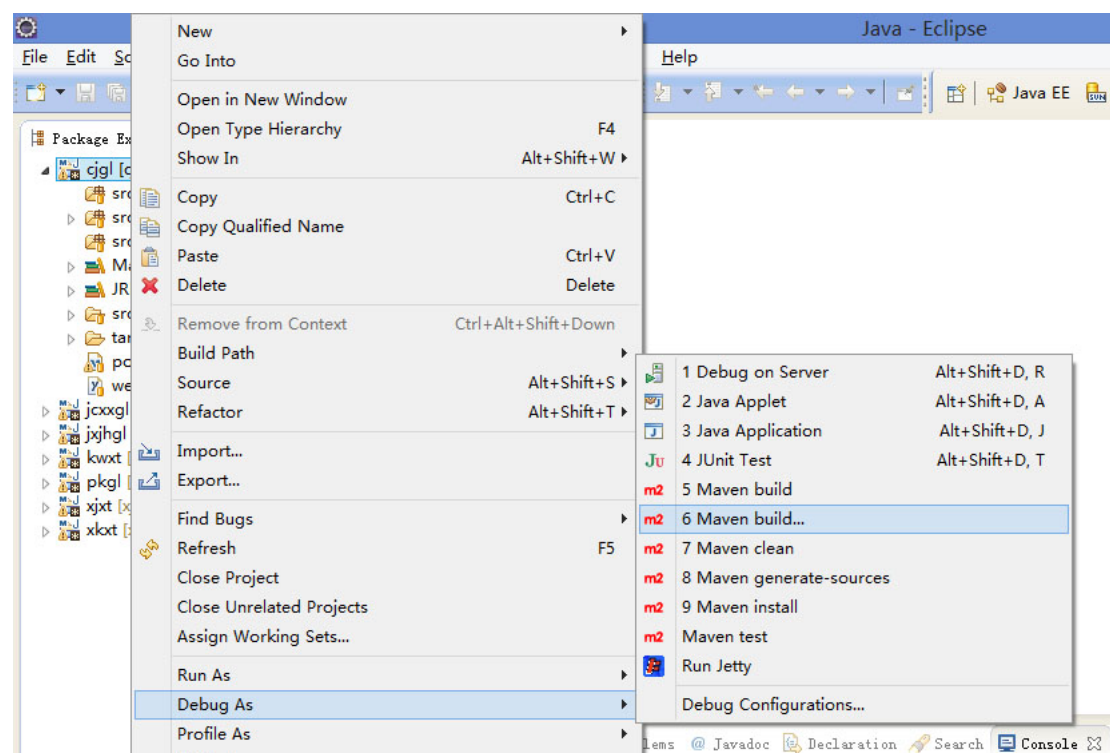


第三步：应用

在项目上右键→Maven→Enable Dependency Management，这样就将一个项目纳入了 Maven 的管理中。

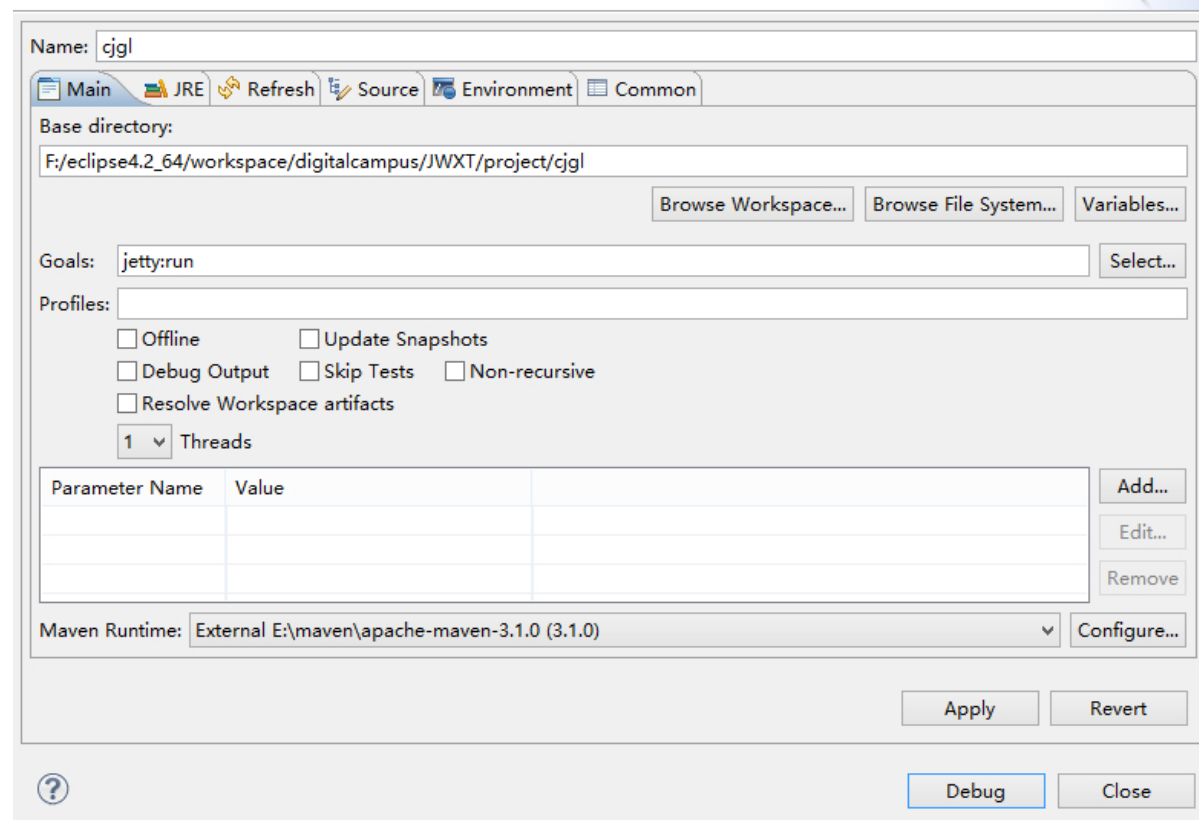
1.2.5.6 Eclipse 中配置 Maven 的 Jetty 插件

选中目标项目，选中【Maven build...】

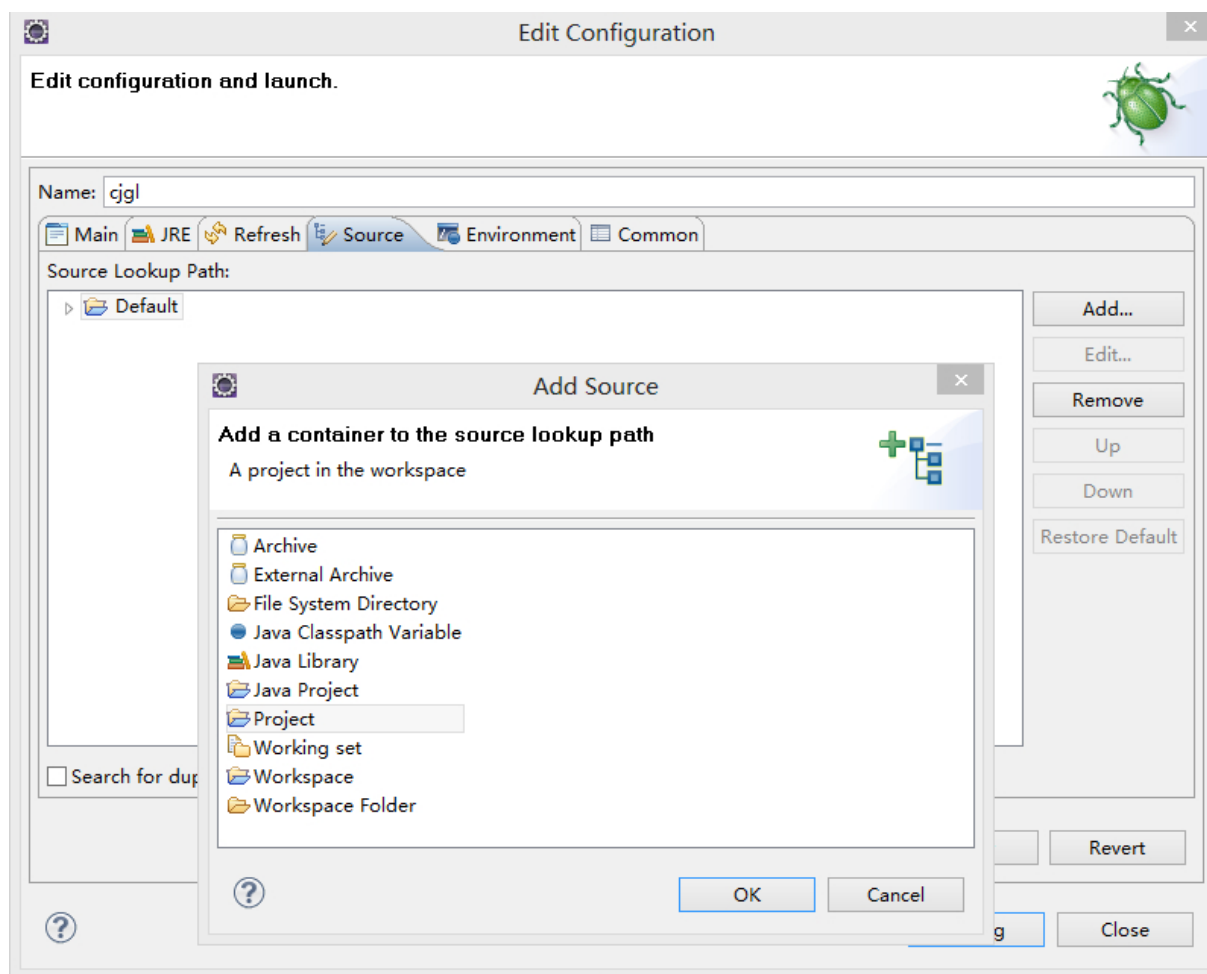


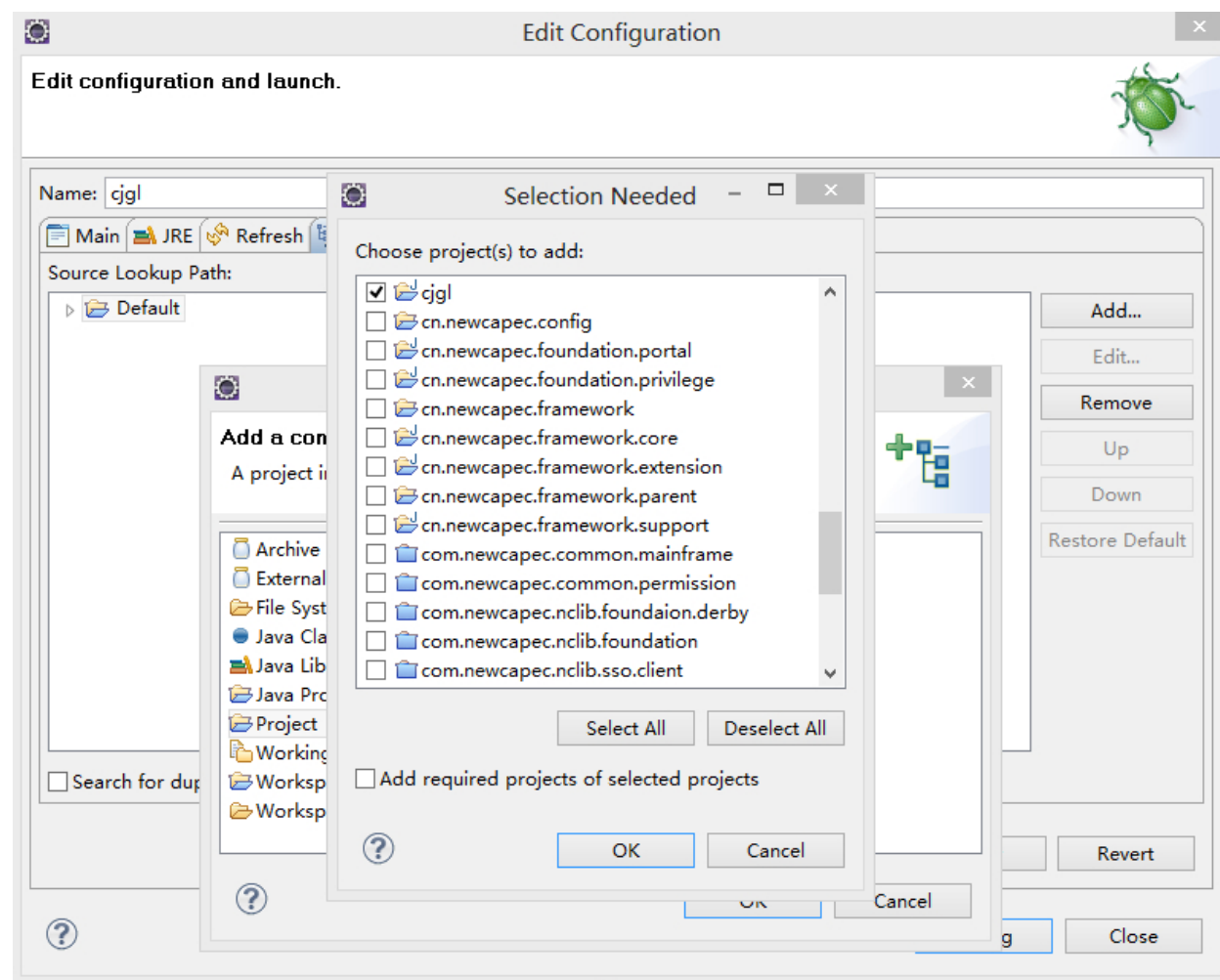
配置标签页 **Main** 中，配置如下图：

Edit configuration and launch.

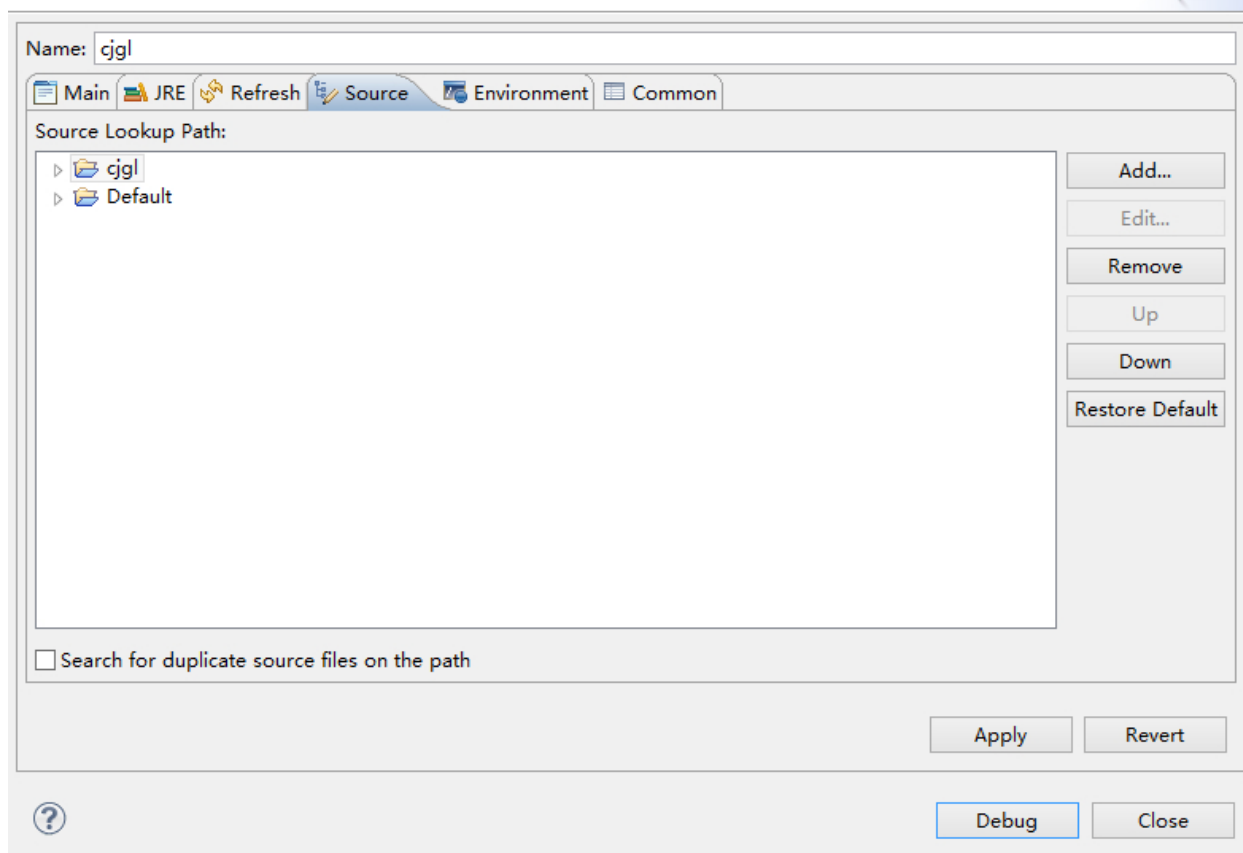


配置标签页 **Source** 中，点击 **Add** 按钮，添加当前项目，如下图所示



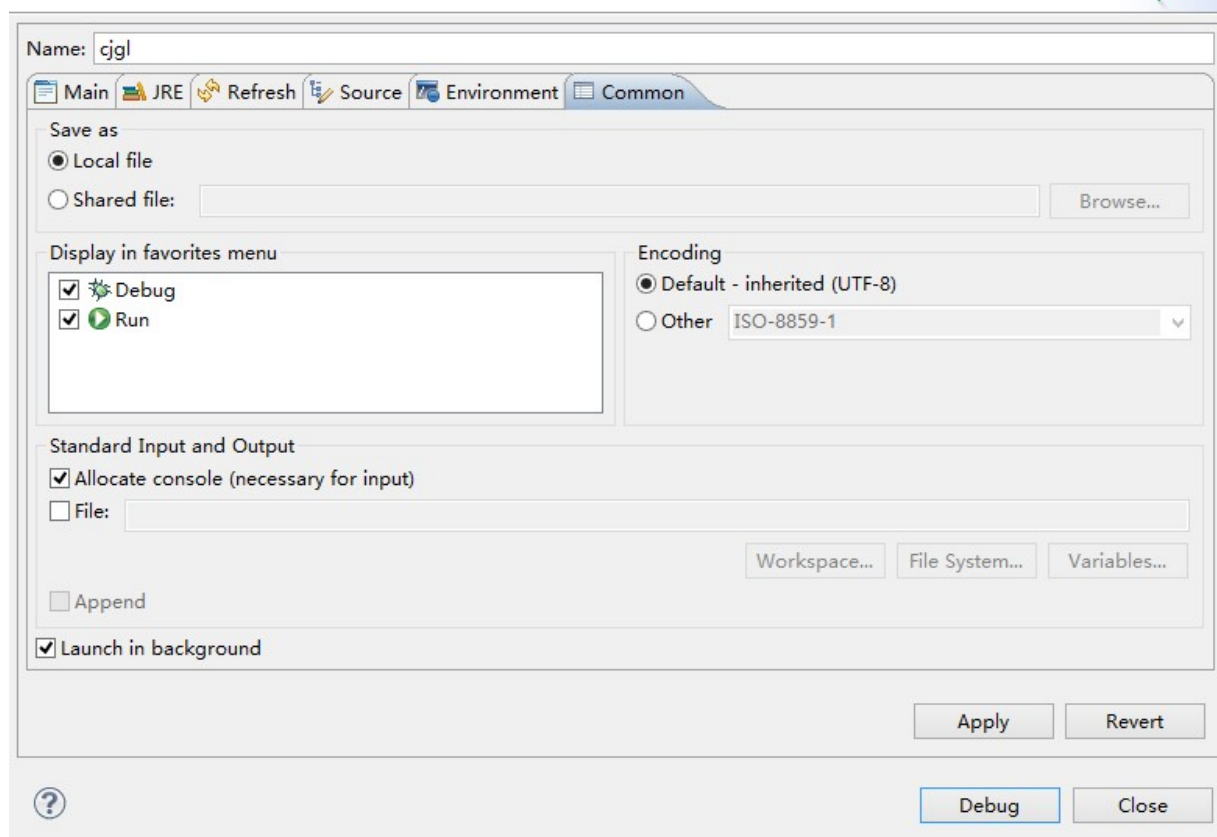


Edit configuration and launch.

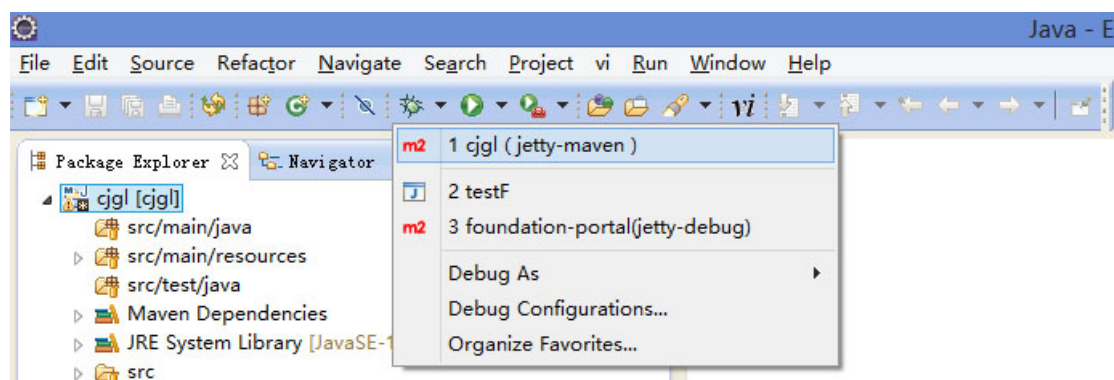


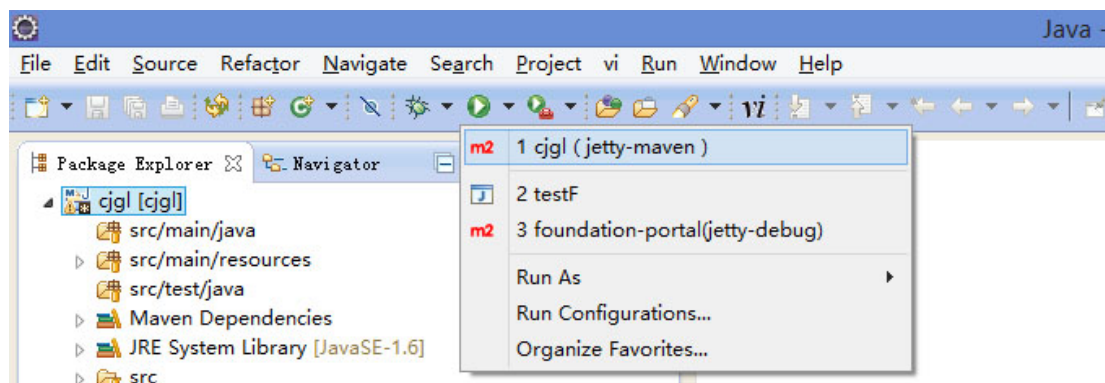
配置标签页 **Common** 中，配置如下图：

Edit configuration and launch.



点击【Apply】按钮后，使配置生效，此时在【运行】与【调试】菜单中可以出现相应的选项，如下图所示：





如上所示，选择相应菜单项后，进入相应的执行状态，或进入相应的调试状态。

2 SVN 版本库

2.1 访问路径

SVN 版本库访问地址：

- 教务系统-基础信息管理：

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net/wcapec.function.digitalcampus.jwxt.jcxxgl>

- 教务系统-学籍系统：

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net/wcapec.function.digitalcampus.jwxt.xjxt>

- 教务系统-选课系统：

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net/wcapec.function.digitalcampus.jwxt.xkxt>

- 教务系统-排课管理：

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net/wcapec.function.digitalcampus.jwxt.pkgl>

- 教务系统-教学计划管理：

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net/wcapec.function.digitalcampus.jwxt.jxjhgl>

- 教务系统-考务系统:

<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net.wcapec.function.digitalcampus.jwxt.kwxt>

- 教务系统-成绩管理:

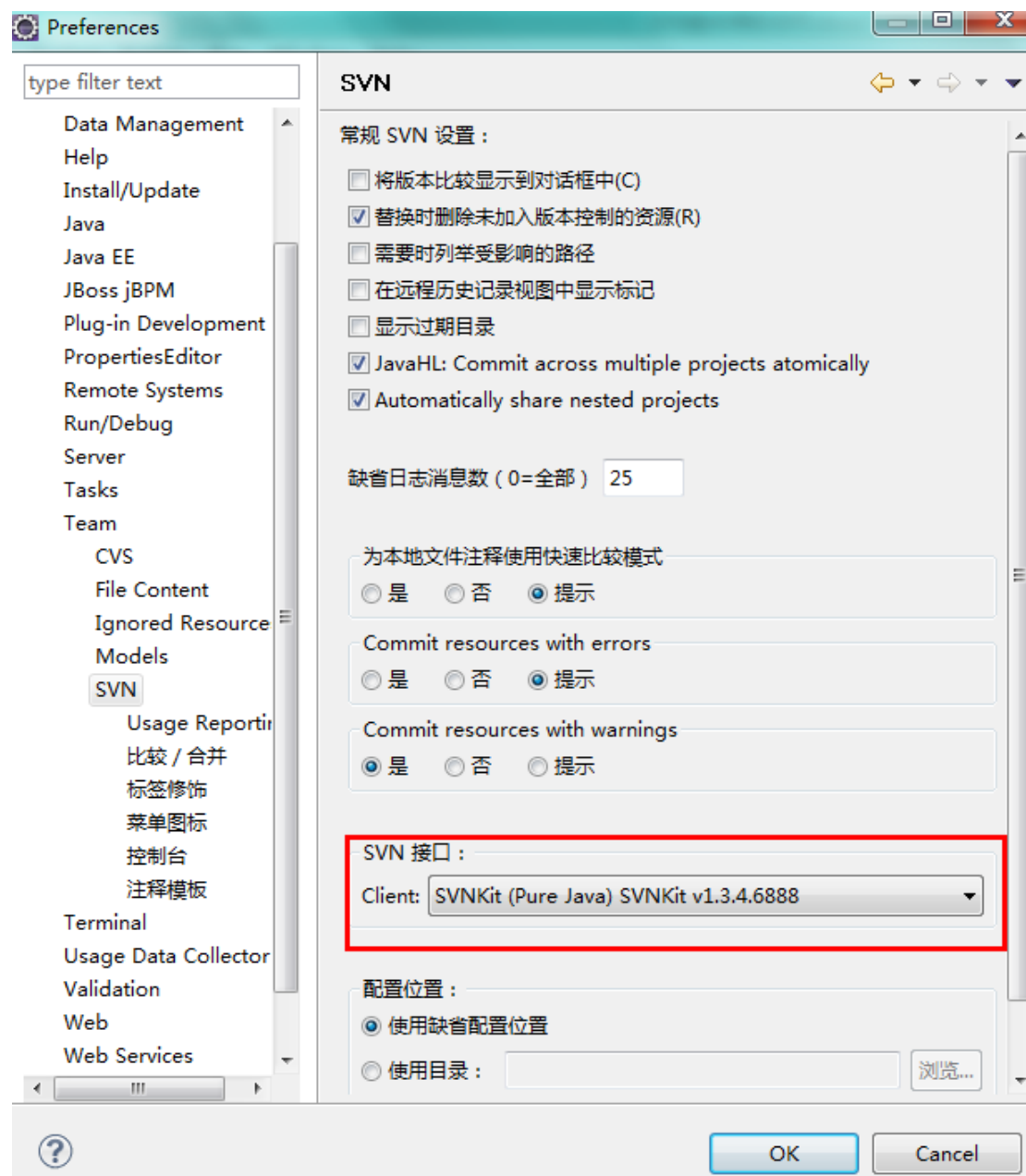
<http://192.168.0.65:8087/svn/digitalcampus/JWXT/Develop/trunk/project/cn.net.wcapec.function.digitalcampus.jwxt.cjgl>

该 SVN 版本库不接受任何不加密的访问，只可通过 https 协议进行访问。若要访问版本库，需下文所述进行相关操作。

2.2 Eclipse 访问

通过 Eclipse 访问 https 协议的版本库地址时，需要对设置稍作改动，以避免每次操作都提示输入证书路径的问题。

Window→Preferences→Team→SVN



将 SVN 接口按如图所示进行选择，然后点击“OK”。

3 创建工程

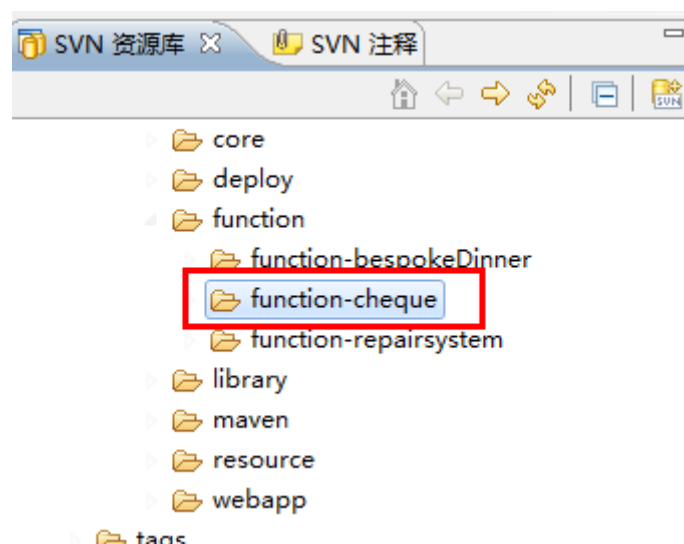
3.1 Check Out 检出代码

3.1.1 Bizplatform 服务开发人员建立工程

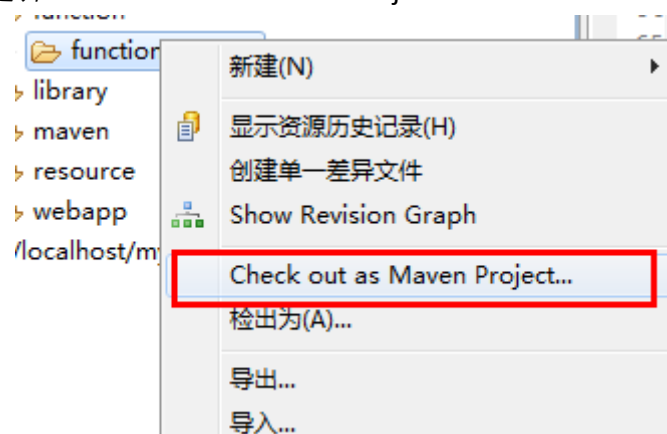
- 每个服务即每个 function 目录作为一个 Java Project 检出。
- Web 资源即 webapp 目录作为一个 Dynamic Web Project 检出。

检出步骤:

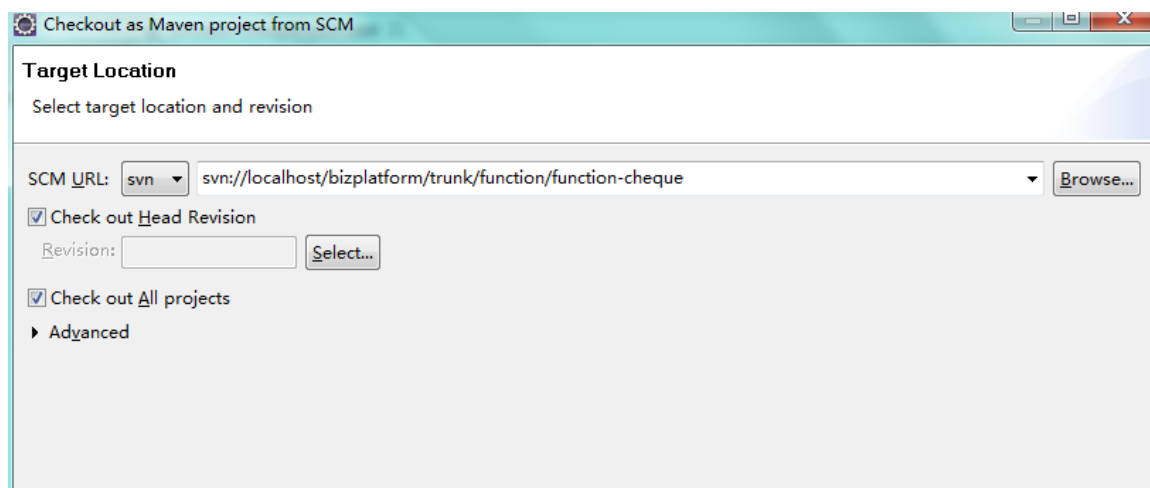
在“SVN 资源库”视图中选择要检出的服务所对应的目录



右键选择“Check out as Maven Project”



在弹出的对话框中不用做任何修改，直接点击 Finish 即可



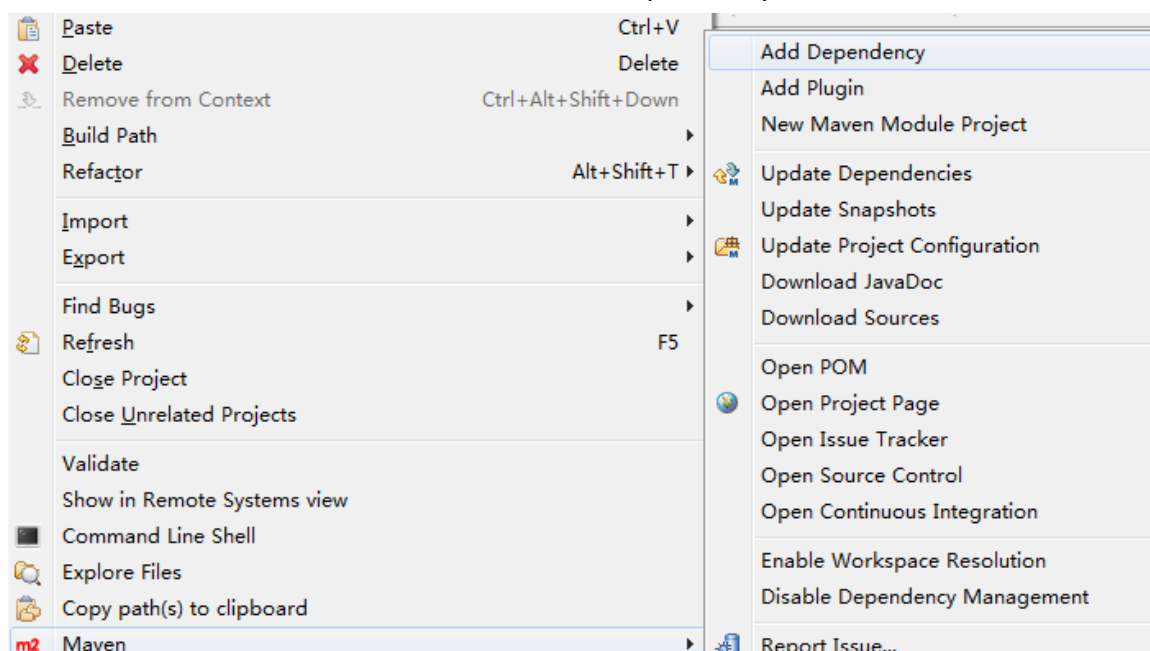
然后切换至 Java EE 透视图即可找到刚刚检出的项目。

注：在检出项目时会同时依据 maven 脚本（即 POM.xml）生成项目结构以及所配置的依赖，依赖会从公司的 Nexus 私服中下载，需下载的依赖数量会影响项目检出的时间。

3.2 依赖管理

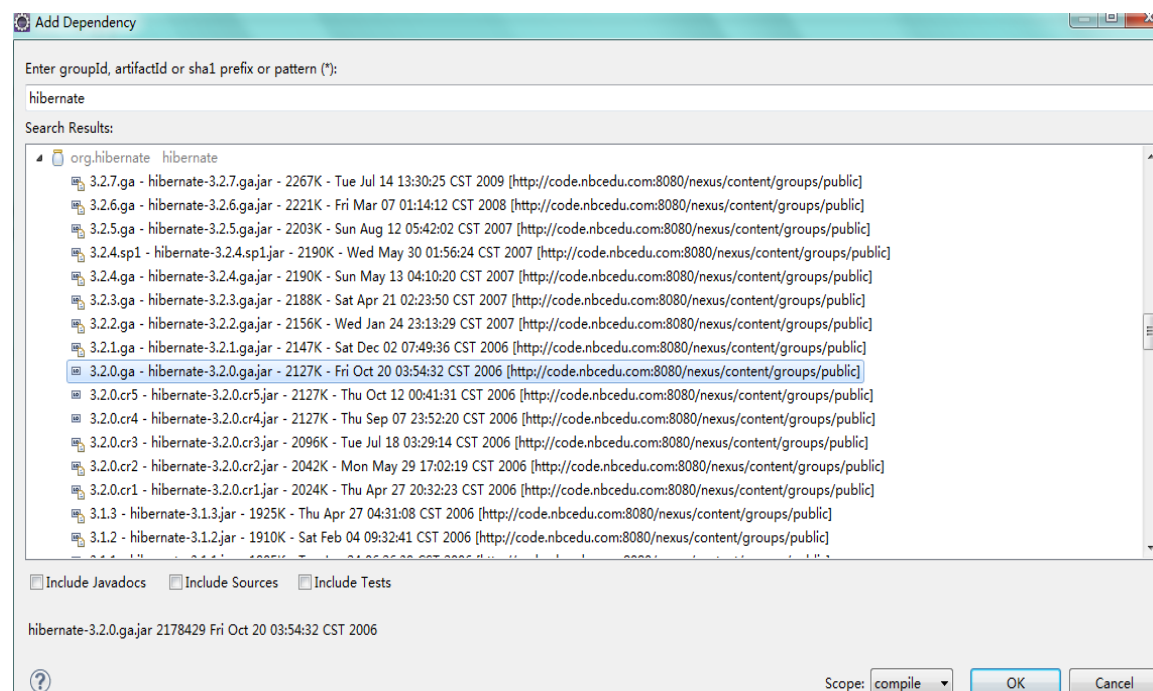
3.2.1 添加 Jar

项目检出后，右键单击项目名称→Add Dependency



在对话框中，输入要添加依赖的坐标关键字查询相应依赖，如要添加

hibernate 依赖如下图所示，并可根据需要勾选源码和 JavaDoc 等选项，同时可选择该依赖的应用作用域。

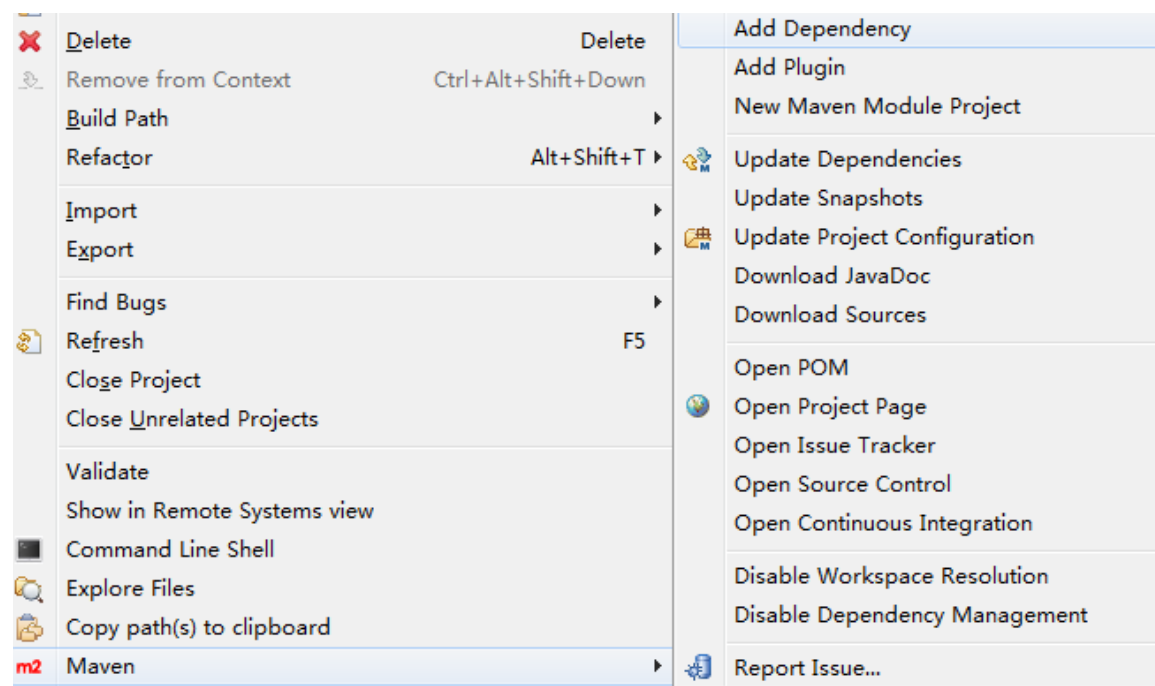


3.2.2 项目间依赖

由于 Java 源码和 Web 资源作为两个项目检出，如果开发人员需要在本地进行联调，需将两个项目之间建立起引用关系。

操作步骤如下：

选中检出的 Web 项目，右键→Maven→Add Dependency



在对话框中输入 Java 项目的项目名，然后选中其对应的构件，点击 OK。

