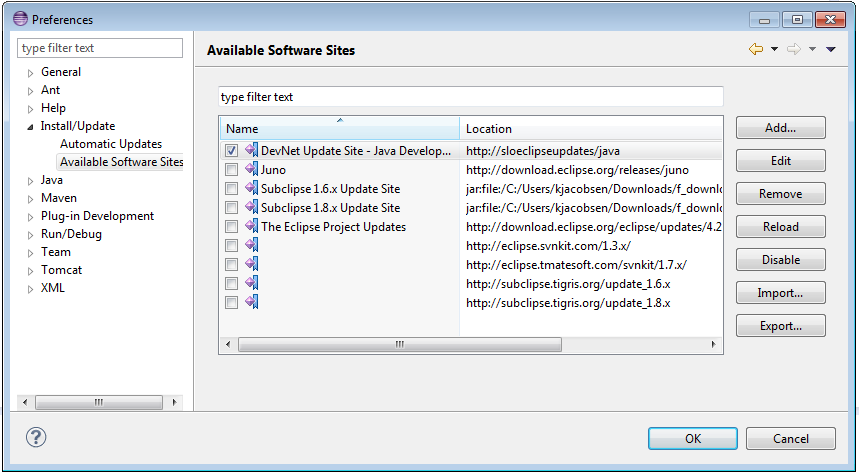
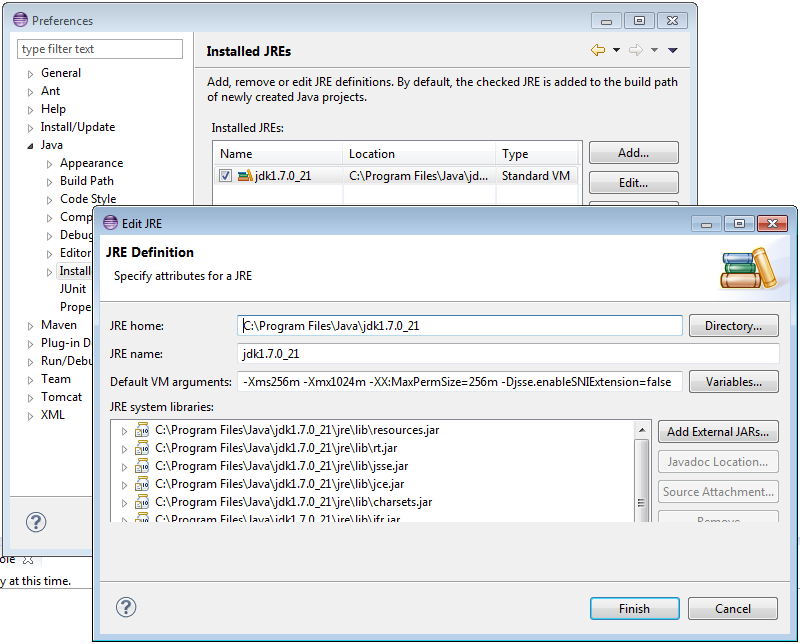
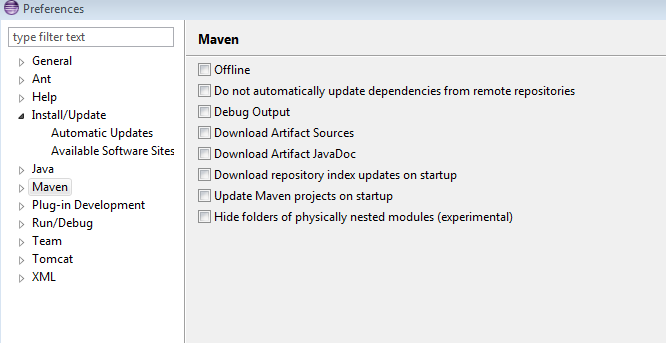
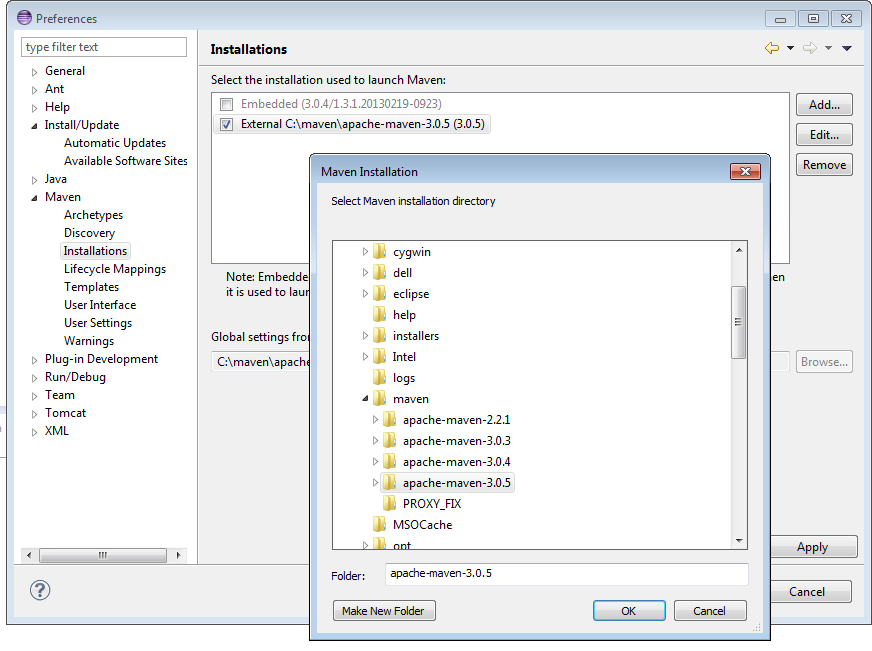
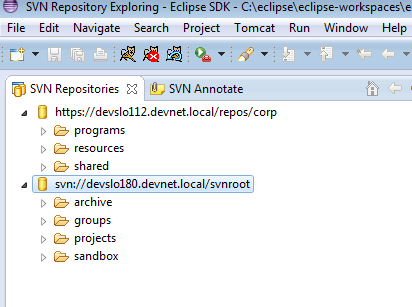
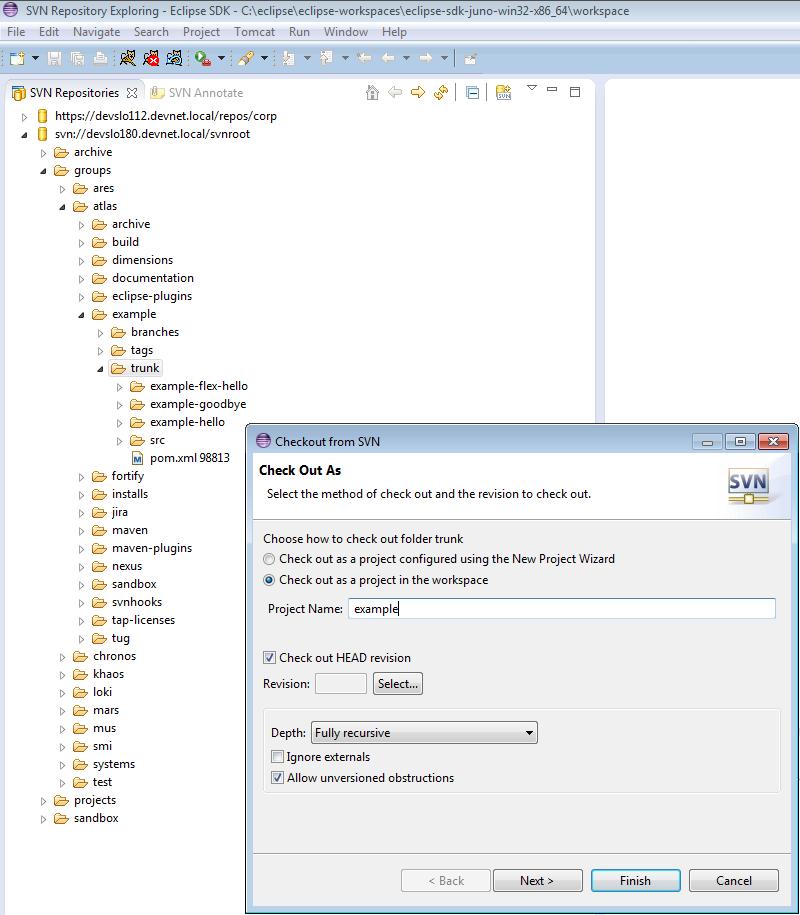
1. Install Java JDK
2. Install Maven under directory like c:\maven\apache-maven-<maven-version>
   1. Example – Maven 3.0.5 would be installed at C:\maven\apache-maven-3.0.5
3. Install Eclipse under directory like C:\eclipse\eclipse-installations\<eclipse-bundle-name>
   1. Example - C:\eclipse\eclipse-installations\eclipse-sdk-juno-win32-x86\_64
   2. Note: By default, the eclipse bundle usually unpacks with just ‘eclipse’ as the directory name. Rename this folder to the package name so that you can have multiple eclipse installations on a single machine.
4. Create shortcut to the eclipse.exe
   1. Example values for shortcut…
      1. Target: C:\eclipse\eclipse-installations\eclipse-sdk-juno-win32-x86\_64\eclipse.exe -data C:\eclipse\eclipse-workspaces\eclipse-sdk-juno-win32-x86\_64\workspace –showlocation -vm "C:\Program Files\Java\jdk1.7.0\_21\bin\javaw.exe"
         1. The “-data <workspace>” option forces eclipse to update a certain workspace so that the user is not prompted each time.
         2. The “-showlocation” option makes eclipse display the workspace location in the title bar.
         3. The “-vm <jdk>” option forces the use of a particular JDK. This produced more consistent results than when the eclipse.ini file was edited…but that was about 3 eclipse versions ago, so maybe things have changed.
      2. Start in: C:\eclipse\eclipse-installations\eclipse-sdk-juno-win32-x86\_64
5. Create workspace location under directory like C:\eclipse\eclipse-workspaces\<eclipse-bundle-name>
   1. Example - C:\eclipse\eclipse-workspaces\eclipse-sdk-juno-win32-x86\_64
   2. You don’t have to create this manually. In the next step you will open eclipse using the shortcut and the workspace will automatically be created.
6. Open eclipse using the shortcut
   1. Add plugins
      1. Configure plugin update site – You can do this under “Windows”->”Preferences” or when installing new software or checking for updates.
      2. Remove out of network sites. Removing them will reduce the likelihood of someone accidently selecting one and then being unable to perform an update.
      3. Go to “Help”->”Install New Software…” and select the plugins that should be included in the bundle. At the time of this writing, all available plugins are being included, but that could change over time.

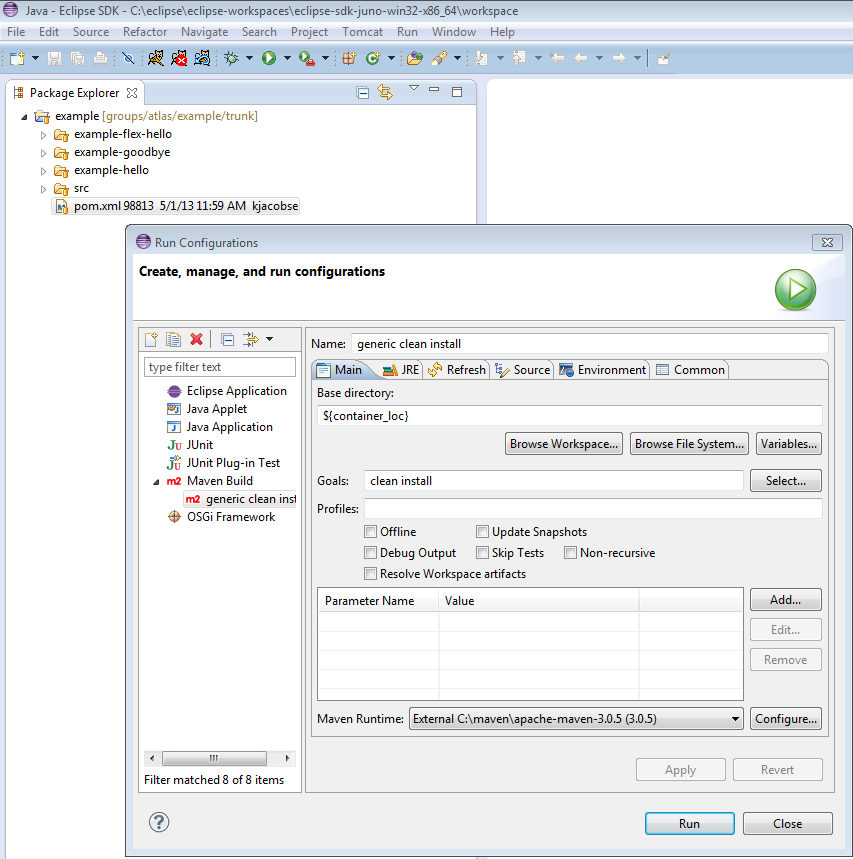
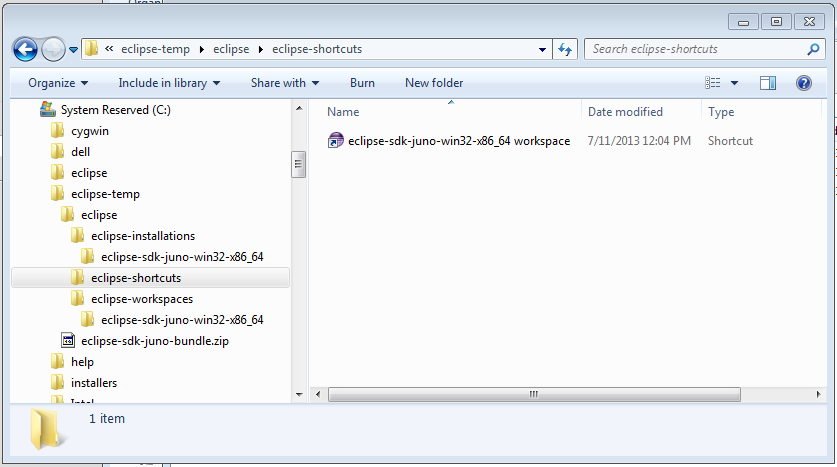


* 1. Configure Java
     1. The Default VM Arguments should be: -Xms256m -Xmx1024m -XX:MaxPermSize=256m -Djsse.enableSNIExtension=false



* 1. Configure Maven
     1. I deselect “Download repository index updates on startup” but it might be good to have “Download Artifact Sources” and “Download Artifact Javadoc” selected. 
  2. Configure SVN
     1. Open “SVN Repositories” perspective
     2. Add relevant repositories
     3. Note: Do not save your user authentication. Users will be prompted for their info.
  3. Test out an example project
     1. Check out the example project (or some other simple project)



* + 1. Create generic build configuration. This can be left in the bundle for ease of use. Some other possible helpful configurations could include something like “generic clean install (Update Snapshots)” or “generic clean install (non-recusive)”
    2. Run a test build
    3. If everything checks out okay, remove the example project. Be sure to select the “Delete project contents on disk (cannot be undone)”, otherwise bundle users will have a rogue project hanging around in their new workspace.
  1. Bundle it up.
     1. Unless you have a better Zip utility than mine, you might have to manipulate this one a little. You want to end up with a zip that contains the eclipse directory at the root and the install, shortcut, and workspace for only the package you are bundling. In the past, I used an ant script. Most recently I accomplished this by temporarily moving the relevant content to a dummy eclipse directory, creating the zip, and then moving everything back.
     2. Name it something like eclipse-sdk-juno-bundle.zip
  2. Test it
     1. Give it to someone else to try it out.
  3. Make it available
     1. Copy it to [\\devslo020\data\devsupport](file:///\\devslo020\data\devsupport), or wherever this stuff is now being kept.