

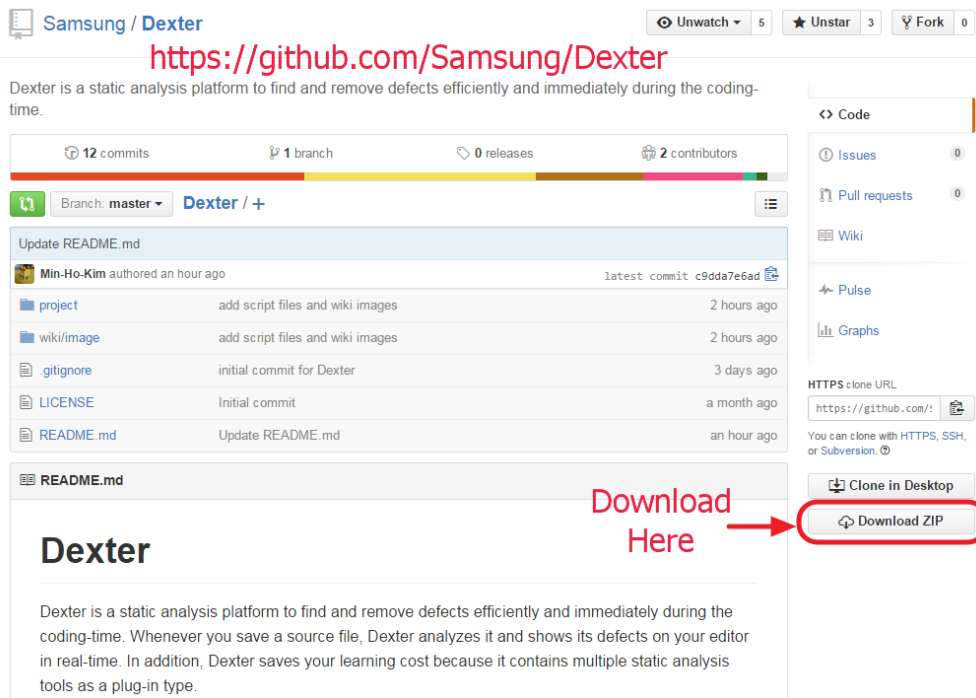
How to build Dexter

How to build and deploy Dexter

- How to build and deploy Dexter
 - 1. Prerequisites
 - 2. Import Dexter Projects into Eclipse
 - 3. Build Dexter CLI
 - 4. Build Dexter Daemon for Source Insight
 - 5. Build Dexter Eclipse Plug-ins

1. Prerequisites

- Install JDK: over 7 update 40
- install Gradle: over 2.5
- install NodeJs: over 0.12.1
- download and unzip Eclipse RCP/RAP version : over Juno(4.2)
 - We will import and build Dexter Projects with the Eclipse
- download Dexter source codes: <https://github.com/Samsung/Dexter>



(*) I downloaded D:/DEV/Dexter

- Folder Description

folder	description	etc.
project/core	Dexter Core Modules Group - main entities, biz logic - static analysis plugin management - utilities, etc	java, eclipse plugin, gradle, ant
project/client	Dexter Client Modules Group - supports Eclipse and Source Insight - eclipse features	java, eclipse plugin/rcp, ant
project/plugin	Static Analysis Plugins Group - dexter-cppcheck for cppcheck - dexter-findbugs for findbugs	java, eclipse plugin, ant
project/common-lib	Libraries that used in Dexter but, can't find in Maven Central Repository	jar

project/script	Script Group - Dexter database creation or deletion - macro file for Source Insight	*.sql, *.em
project/lic	License Files Group	*.txt

2. Import Dexter Projects into Eclipse

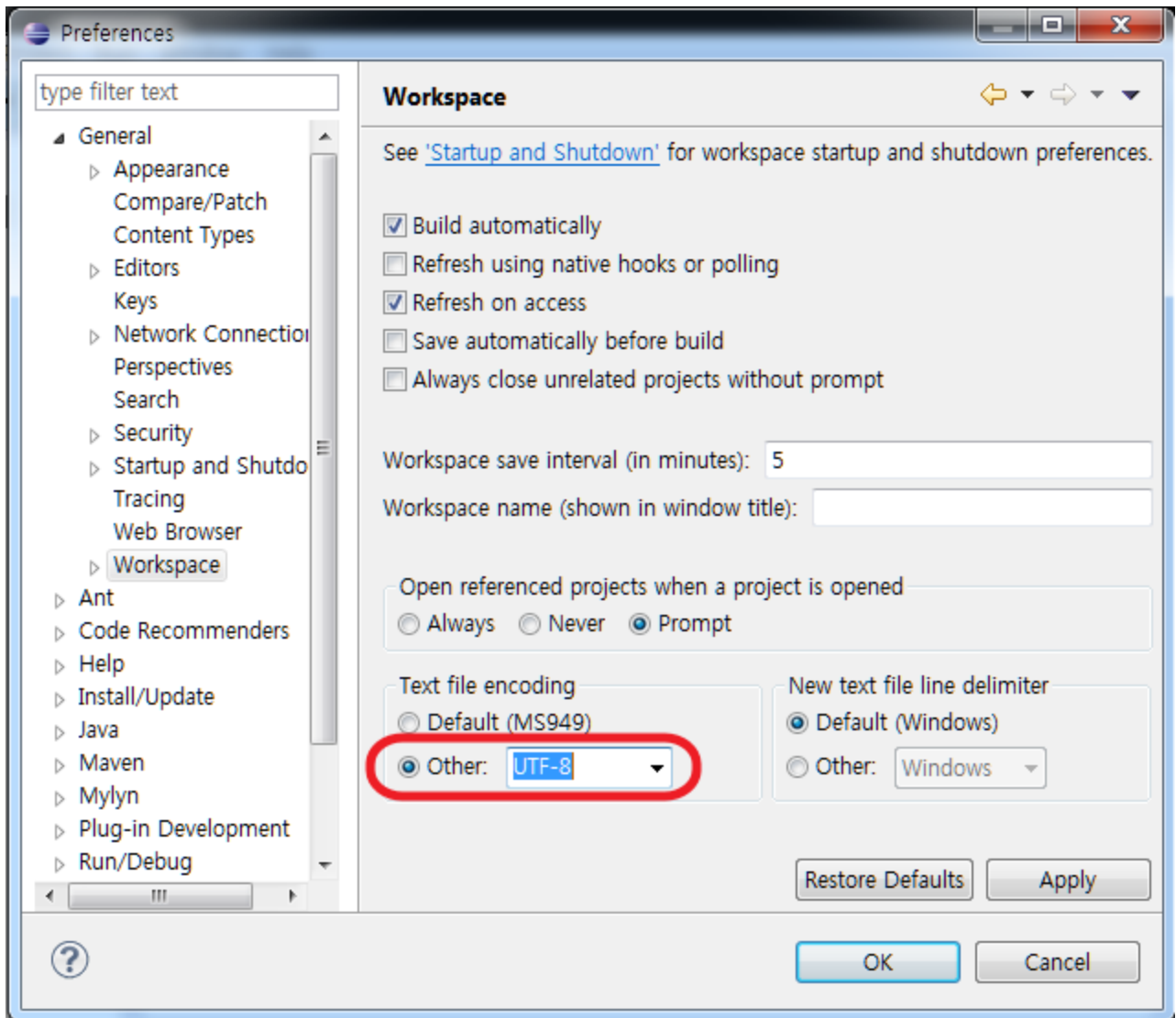
- edit 'eclipse.ini' file in your eclipse folder

```

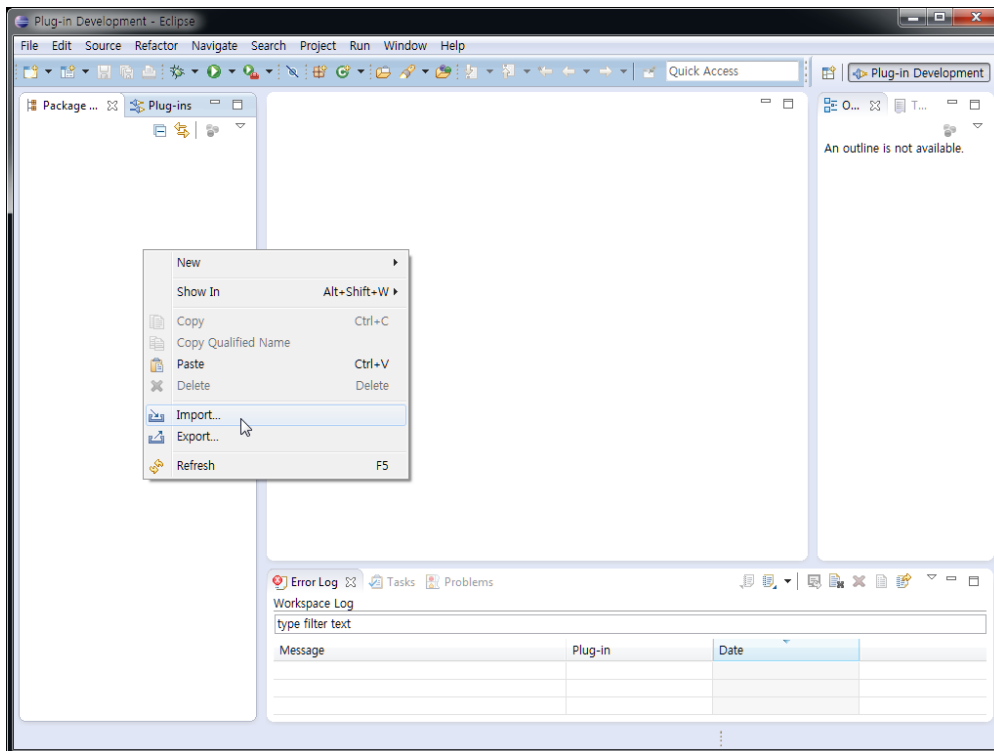
1 -startup
2 plugins/org.eclipse.equinox.launcher_1.3.0.v20120522-1813.jar
3 --launcher.library
4 plugins/org.eclipse.equinox.launcher.win32.win32.x86_64_1.1.200.v20120913-144807
5 -product
6 org.eclipse.epp.package.rcp.product
7 --launcher.defaultAction
8 openFile
9 --launcher.XXMaxPermSize
10 256M
11 -showsplash
12 org.eclipse.platform
13 --launcher.XXMaxPermSize
14 256m
15 --launcher.defaultAction
16 openFile
17 -vmargs
18 -Dosgi.requiredJavaVersion=1.5
19 -Dhelp.lucene.tokenizer=standard
20 -Xms768m
21 -Xmx1024m

```

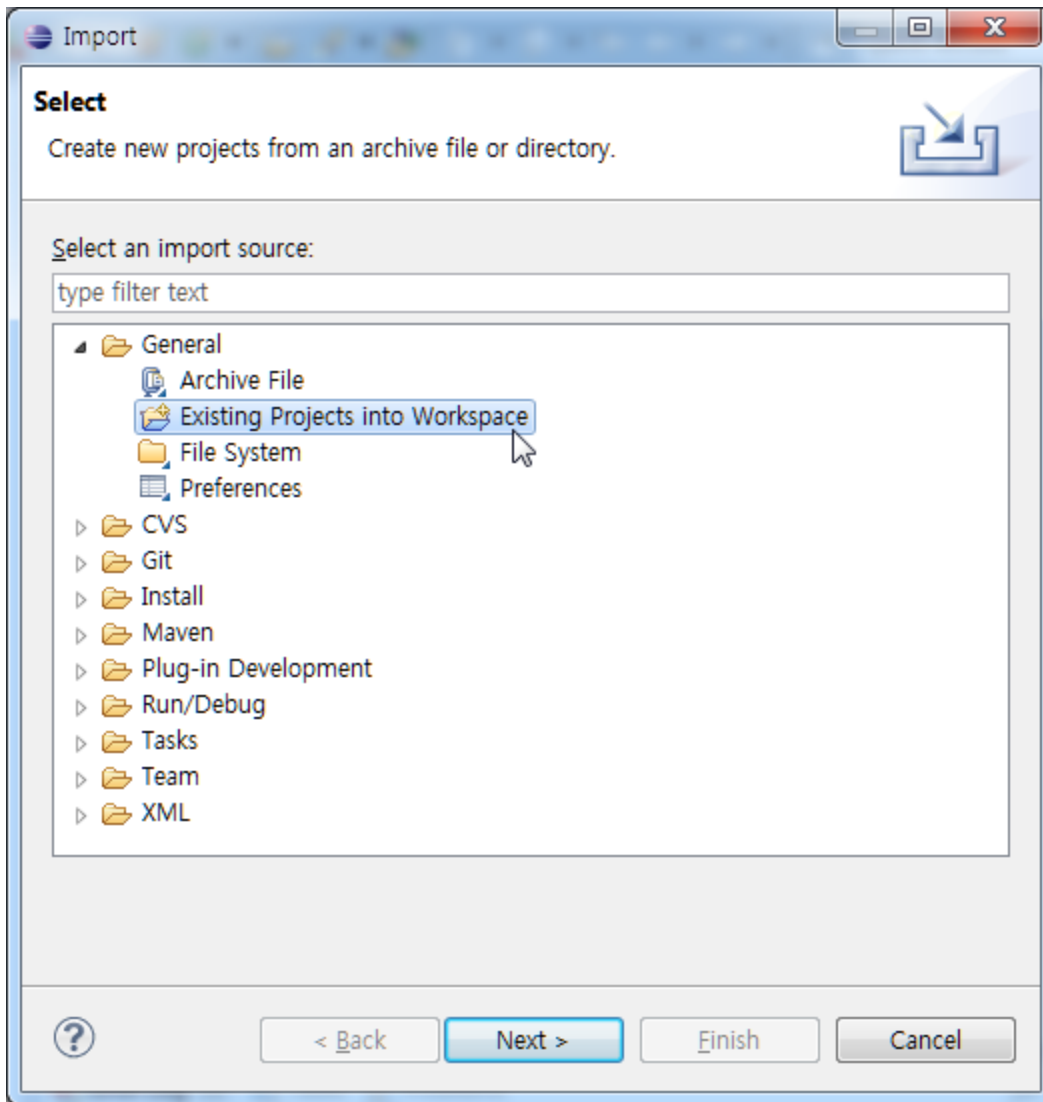
- --launcher.XXMaxPermSize256M
- --Xmx1024m
- run eclipse
- change encoding to 'UTF-8': top menu > Preferences > General > Workspace



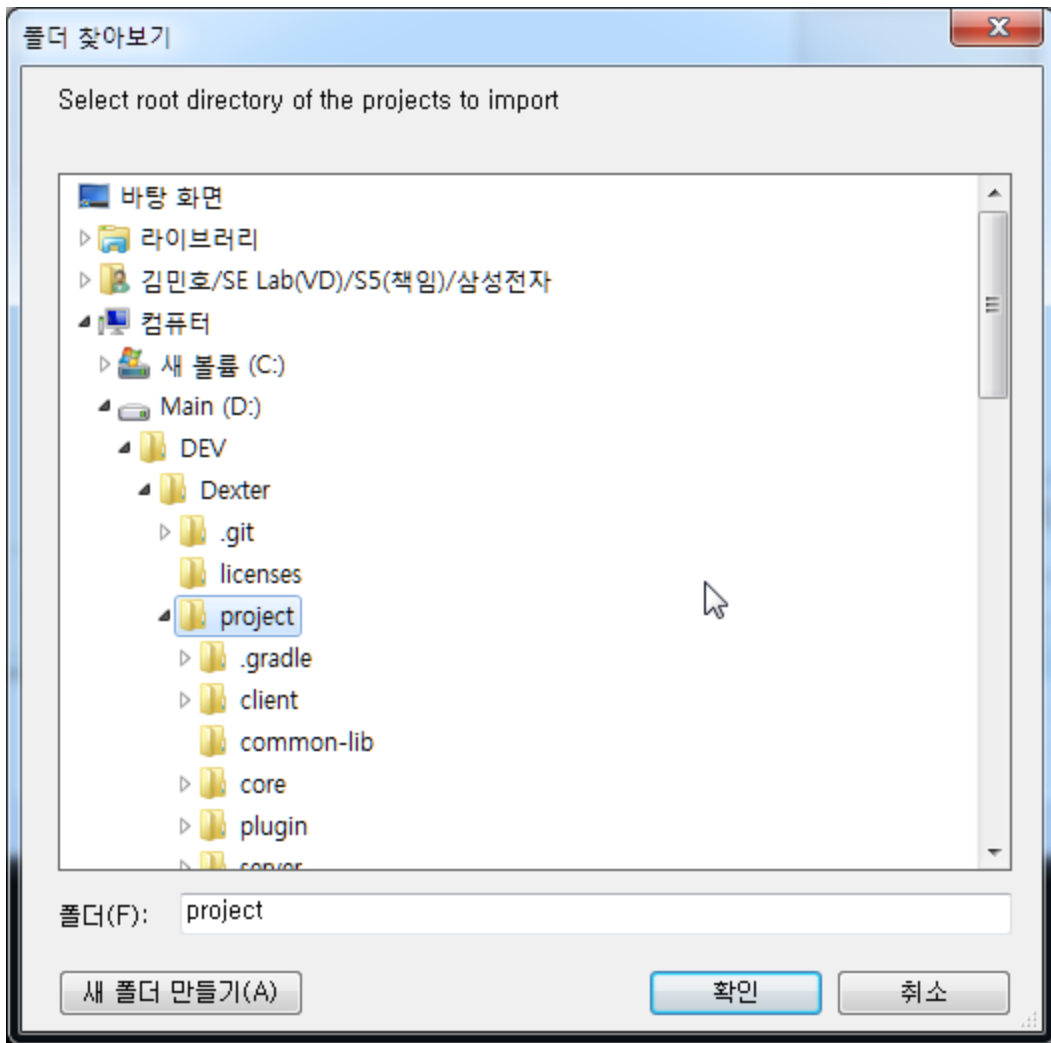
- if you want to test a clean eclipse rcp, you can add target platform in Preferences (optional)
- click 'Import' menu on the Package Explorer View.



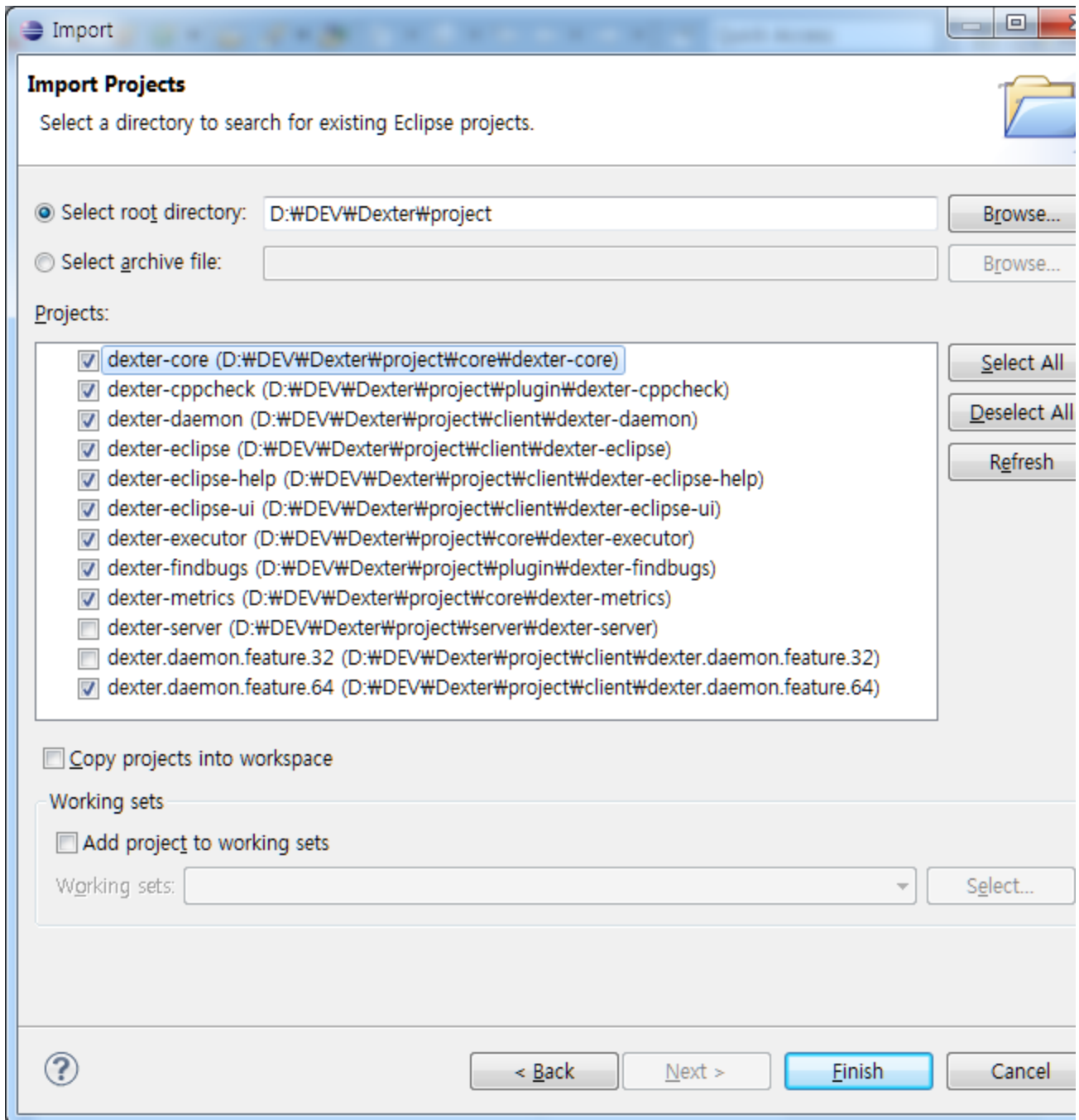
- select 'Existing Projects into Workspace' item



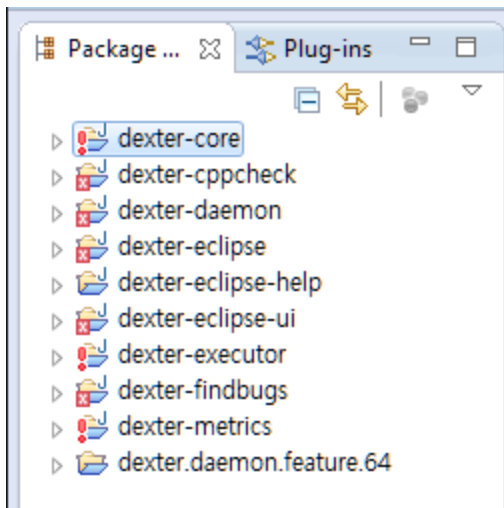
- select your project folder that you downloaded: (* in my case: D:/DEV/project)



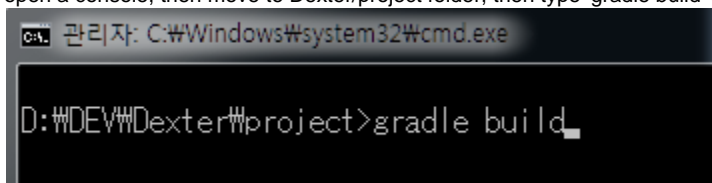
- check projects that you will use (except dexter-server project)



- click 'Finish' button
- you will see the 'error' marks on all of the projects, because there are no library files yet



- open a console, then move to Dexter/project folder, then type 'gradle build' command



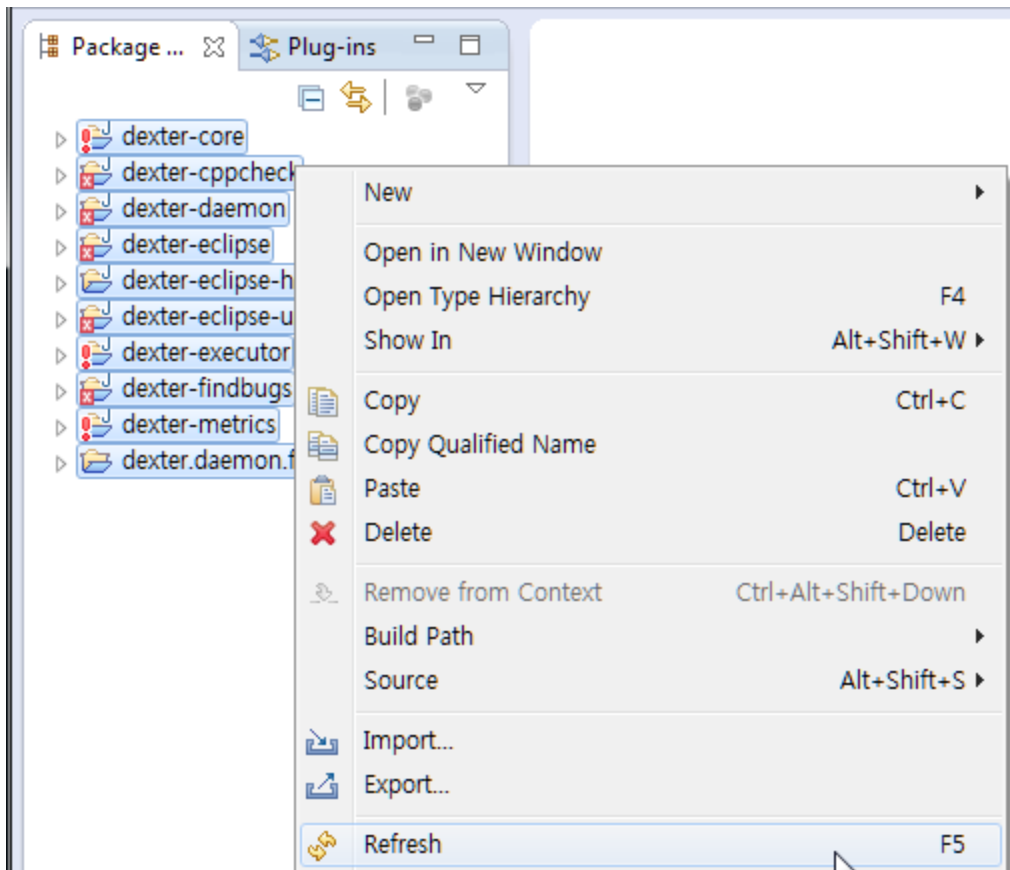
- it will download libraries that we need


```
관리자: C:\Windows\system32\cmd.exe
D:\DEV\Dexter\project>gradle build
:core:compileJava UP-TO-DATE
:core:processResources UP-TO-DATE
:core:classes UP-TO-DATE
:core:jar
:core:assemble
:core:compileTestJava UP-TO-DATE
:core:processTestResources UP-TO-DATE
:core:testClasses UP-TO-DATE
:core:test UP-TO-DATE
:core:check UP-TO-DATE
:core:build
:core:dexter-core:compileJava
Note: D:\DEV\Dexter\project\core\dexter-core\src\java\com\samsung\sec\dexter\core\config\DexterConfigFile
ecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
:core:dexter-core:processResources
:core:dexter-core:classes
:core:dexter-core:jar
:core:dexter-core:assemble
:core:dexter-core:compileTestJava
Note: D:\DEV\Dexter\project\core\dexter-core\src\test\com\samsung\sec\dexter\core\util\DexterClientIT.jav
ides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
:core:dexter-core:processTestResources UP-TO-DATE
:core:dexter-core:testClasses
:core:dexter-core:test
:core:dexter-core:check
:core:dexter-core:build
:core:dexter-metrics:compileJava
:core:dexter-metrics:processResources UP-TO-DATE
:core:dexter-metrics:classes
:core:dexter-metrics:jar
:core:dexter-executor:compileJava
:core:dexter-executor:processResources
:core:dexter-executor:classes
:core:dexter-executor:jar
:core:dexter-executor:assemble
:core:dexter-executor:compileTestJava
:core:dexter-executor:processTestResources UP-TO-DATE
:core:dexter-executor:testClasses
:core:dexter-executor:test
:core:dexter-executor:check
:core:dexter-executor:build
:core:dexter-metrics:assemble
:core:dexter-metrics:compileTestJava UP-TO-DATE
:core:dexter-metrics:processTestResources UP-TO-DATE
:core:dexter-metrics:testClasses UP-TO-DATE
:core:dexter-metrics:test UP-TO-DATE
:core:dexter-metrics:check UP-TO-DATE
:core:dexter-metrics:build

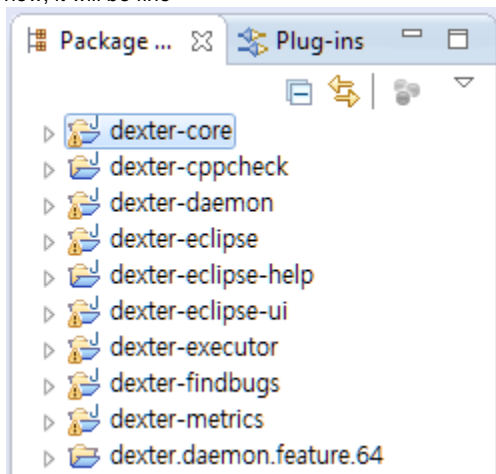
BUILD SUCCESSFUL

Total time: 28.542 secs
D:\DEV\Dexter\project>
```

- select all projects and refresh them

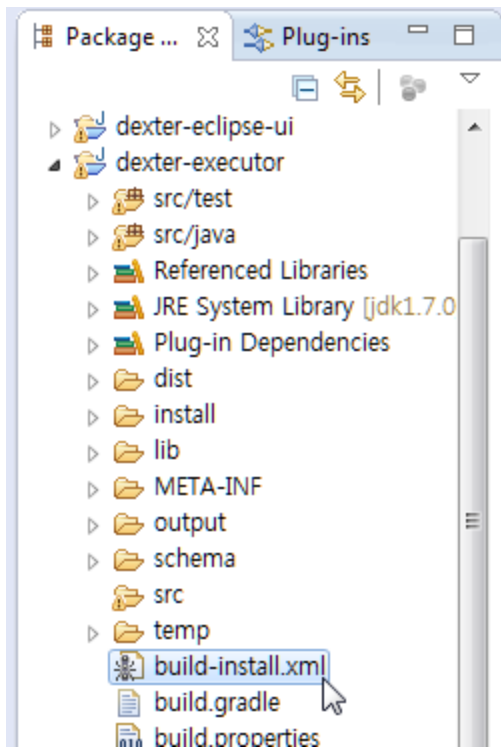


- now, it will be fine

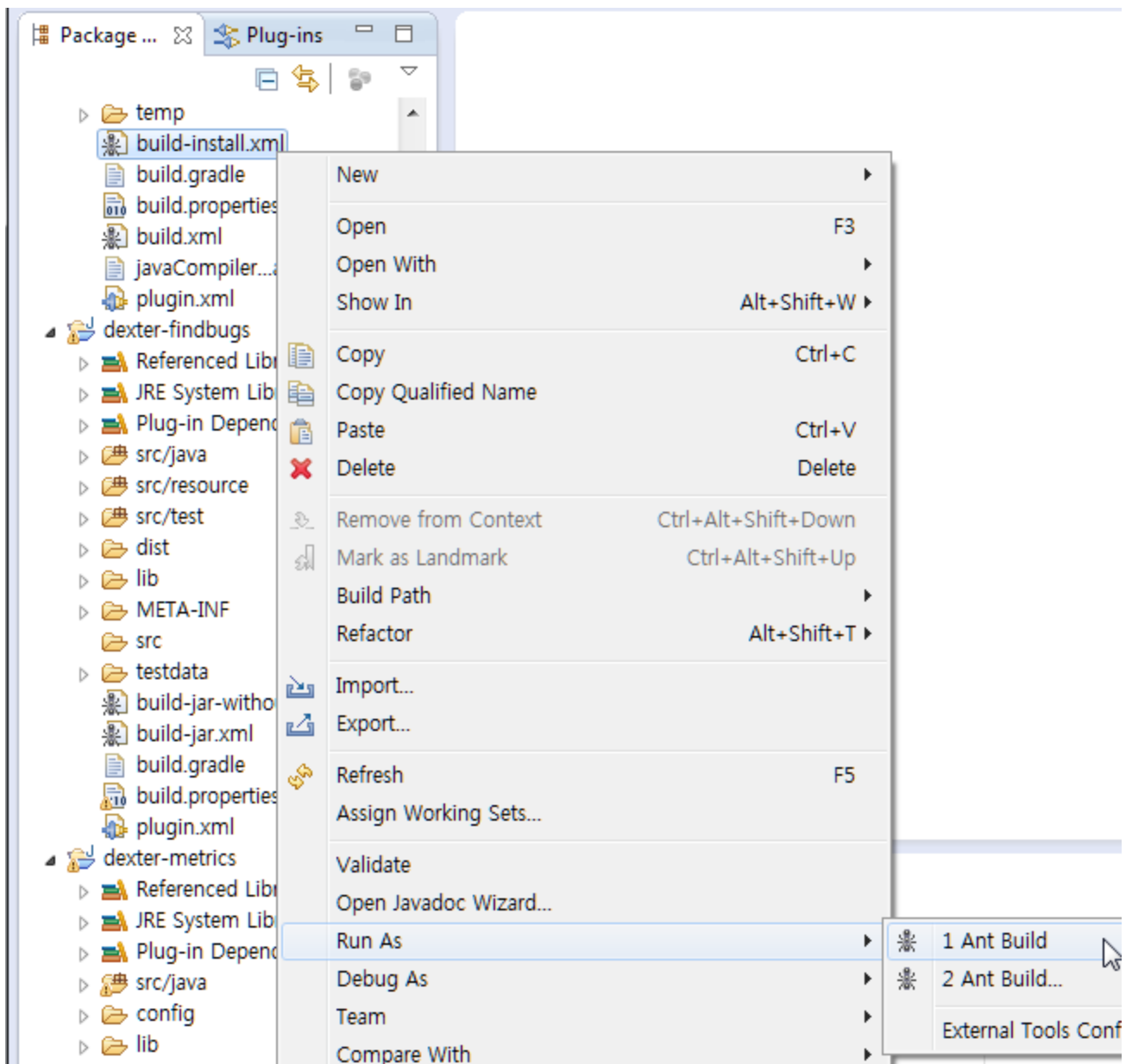


3. Build Dexter CLI

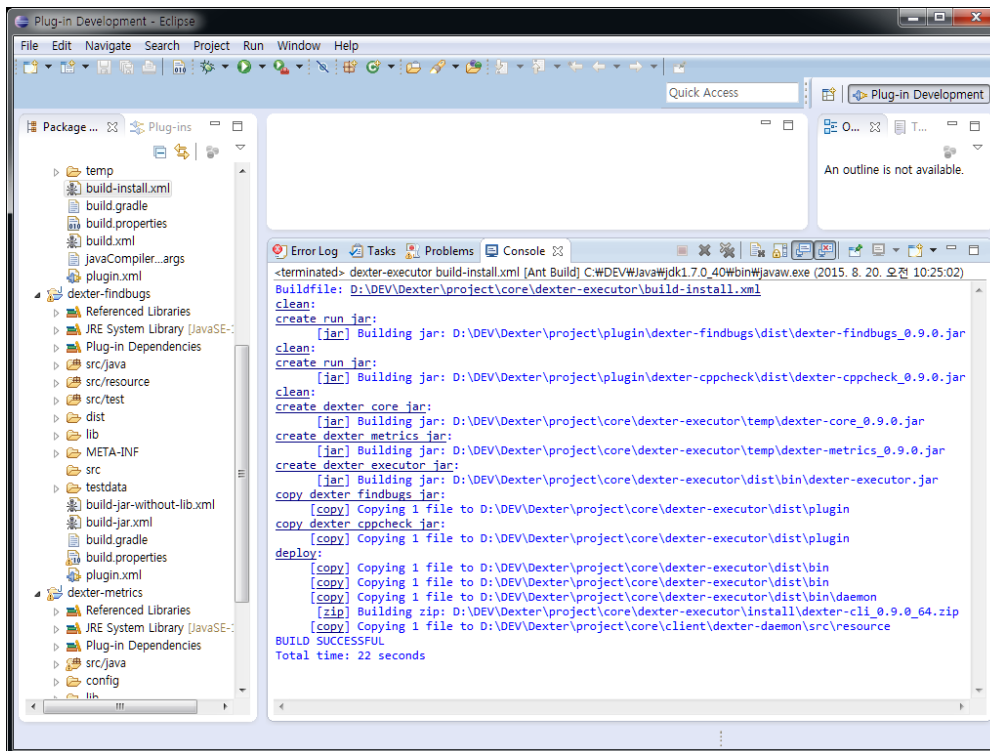
- open dexter-executor project



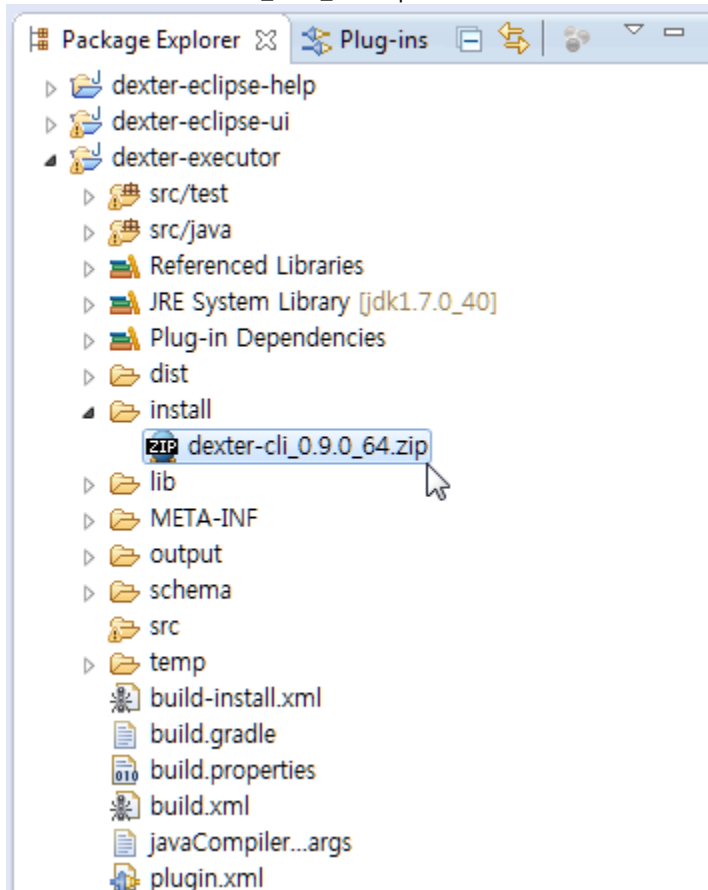
- select 'build-install.xml' file, then 'Run As' and 'Ant Build'



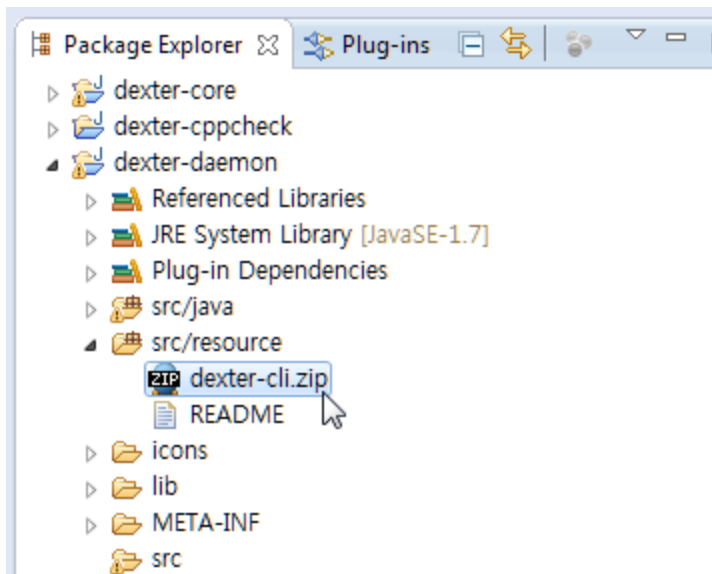
- if build is fine, you will see the "Build SUCCESSFUL" message on the Console view



- then there will be 'dexer-cli_#. #. #_osbit.zip' file under the 'dexer-executor/install' folder

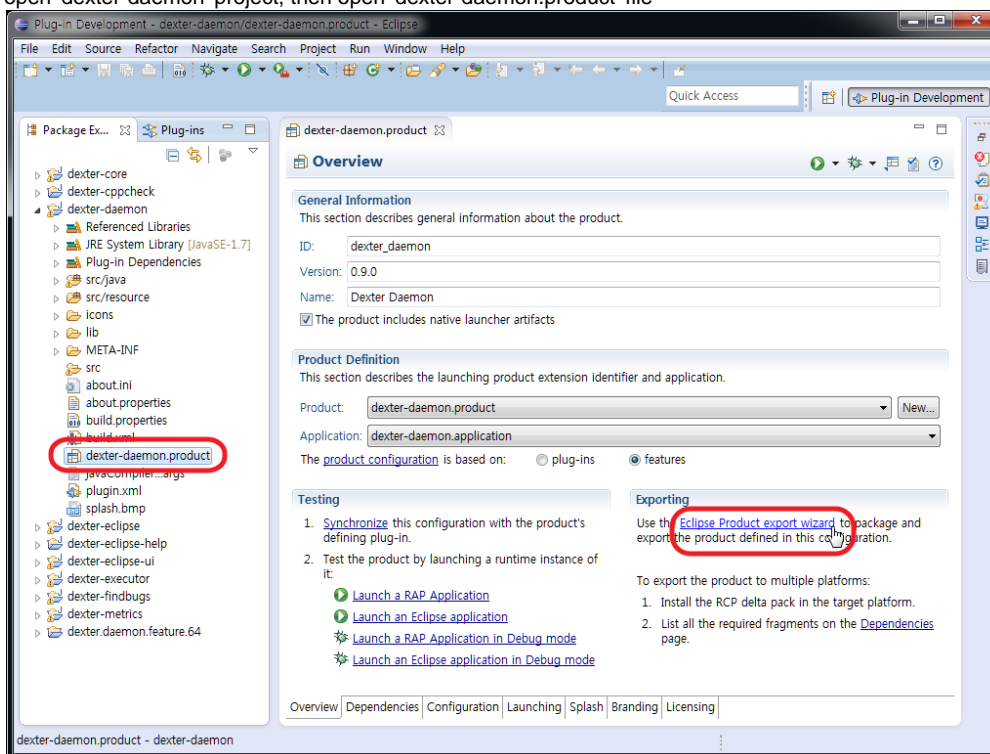


- you can use the zip file for Dexter CLI (refer to "Dexter CLI Guide")
- also same file is copied to 'dexer-daemon/src/resource/dexer-cli.zip' file

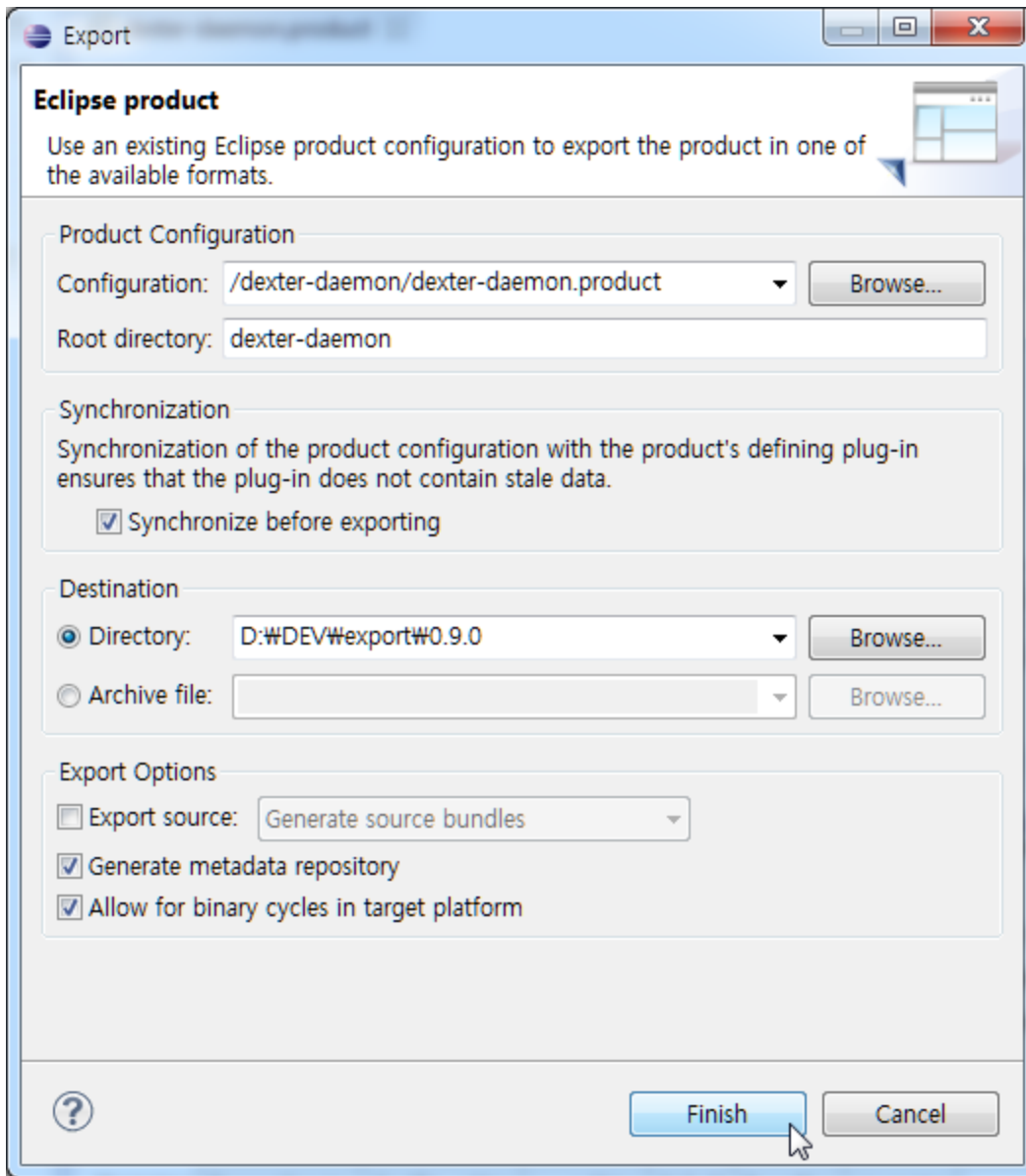


4. Build Dexter Daemon for Source Insight

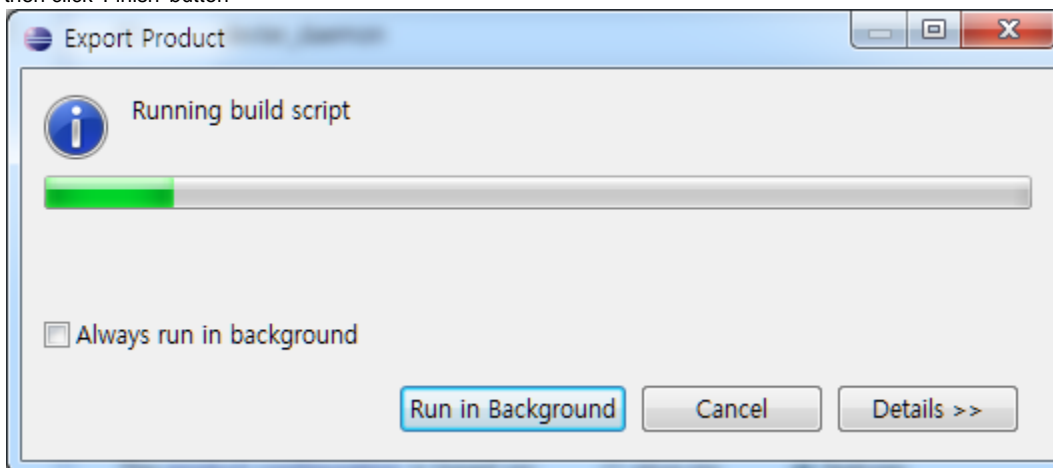
- open 'dexter-daemon' project, then open 'dexter-daemon.product' file



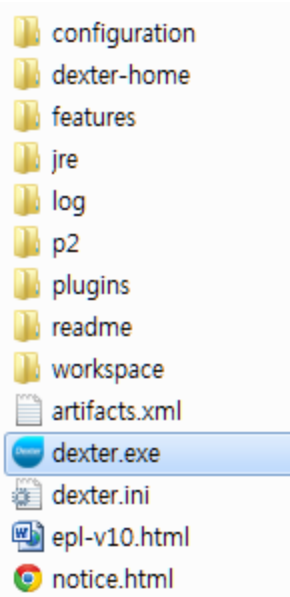
- then click "Eclipse Product export wizard" link on the Exporting tab
- fill the form



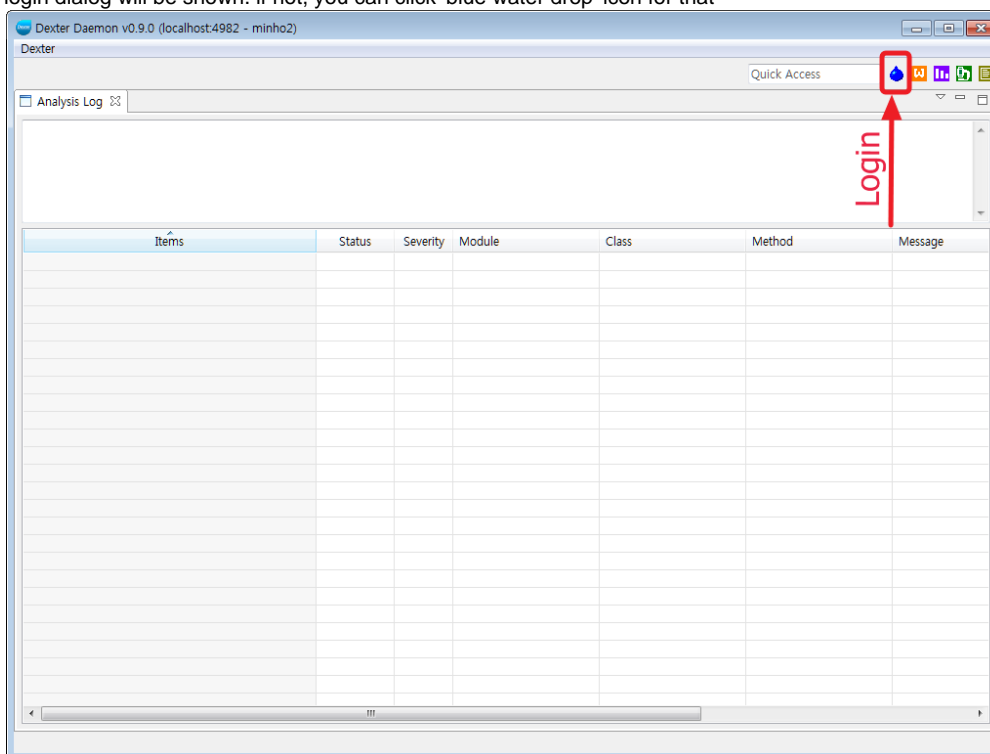
- Root directory : dexter-daemon (* it can be folder name)
- Directory : export folder (*in my case, D:/DEV/export/0.9.0)
- then click 'Finish' button



- you can run 'dexter.exe' file the export folder. (* in my case, C:/DEV/export/0.9.0/dexter-daemon)



- login dialog will be shown. if not, you can click 'blue water drop' icon for that



- if you don't have Dexter Server yet, click 'Run in Standalone mode' checkbox

Dexter 서버 로그인

Dexter 로그인

* 전달받은 Dexter Server IP와 Port를 입력하세요. (예: 100.100.100.100:4982)

Single ID: minho

Password: ●●●●

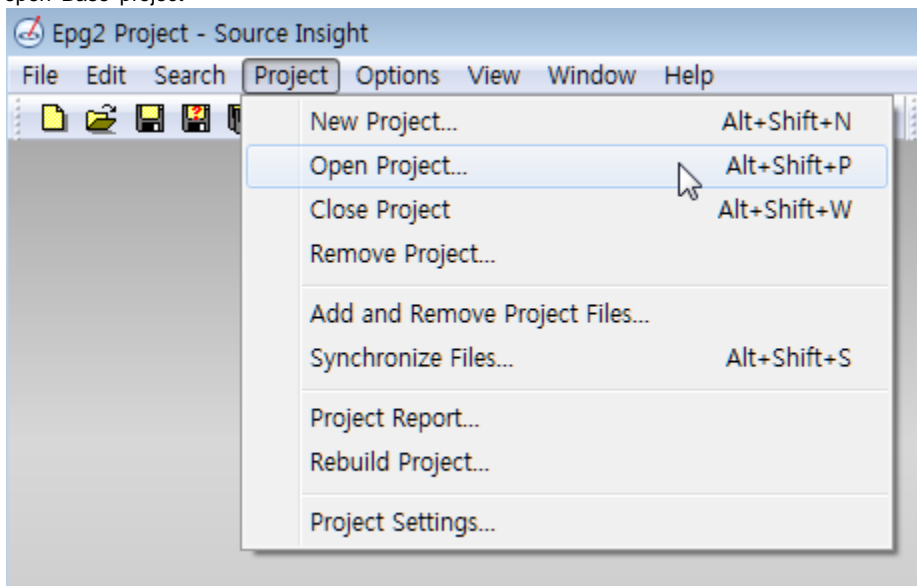
Dexter Server(IP:Port): localhost:4982 접속 테스트

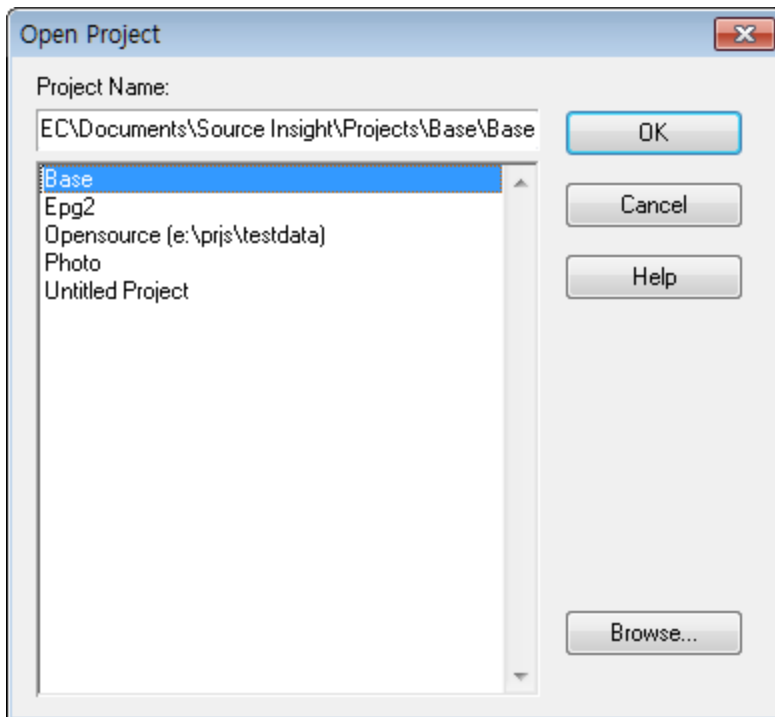
Dexter Home Path: D:/DEV/export/dexter-daemon/dexter-home 찾기...

☒ Run in Standalone mode

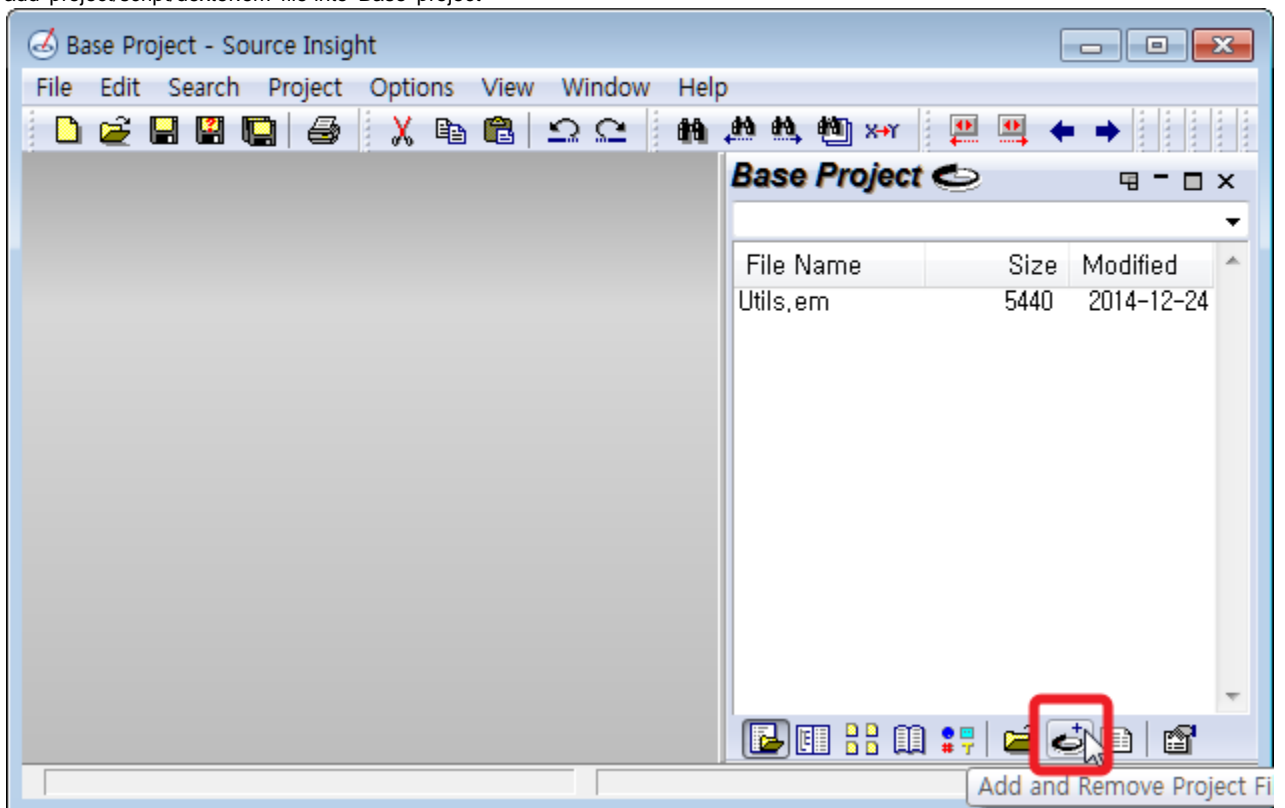
OK Cancel

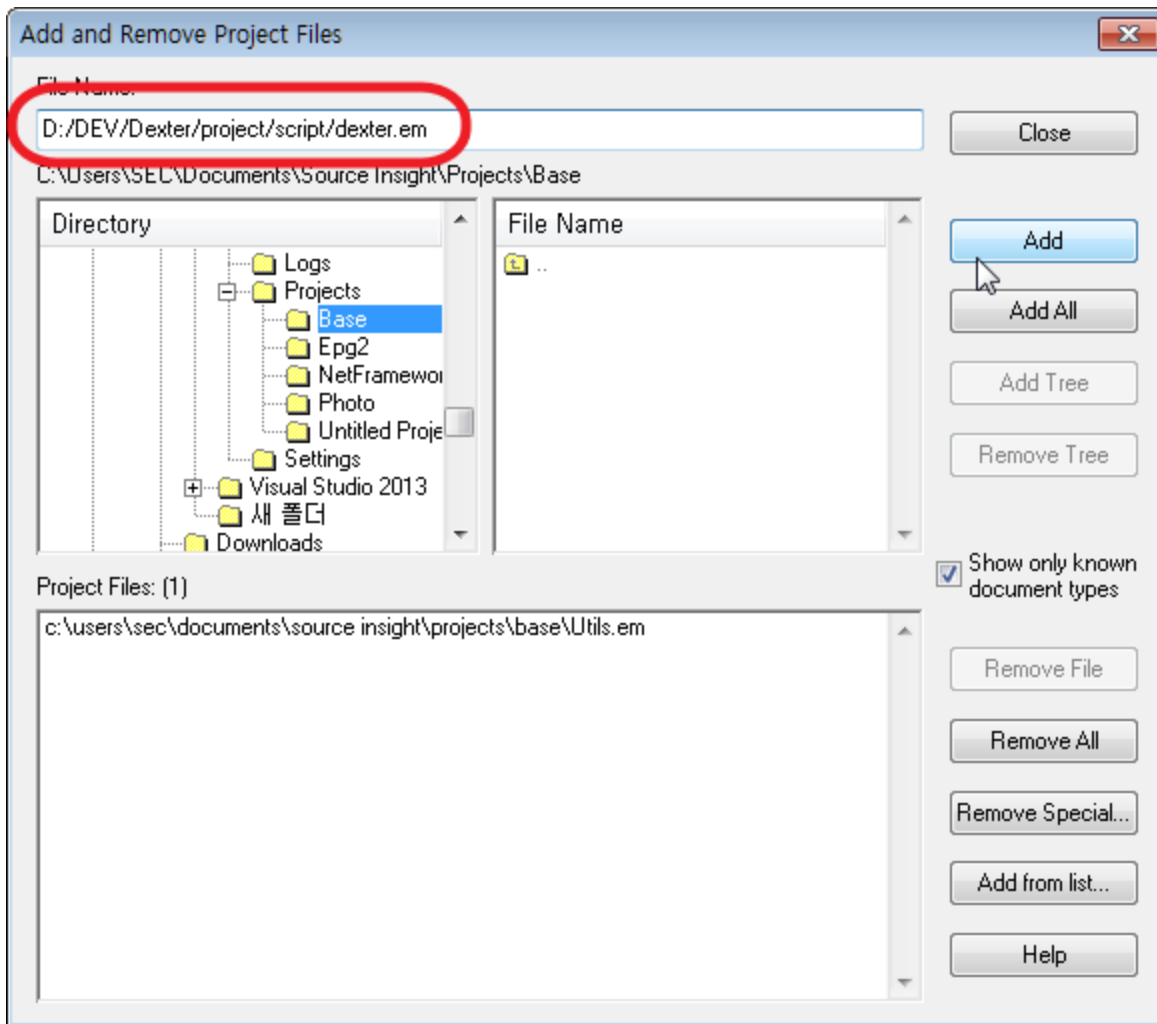
- run Source Insight
- open 'Base' project





- add 'project/script/dexter.em' file into 'Base' project

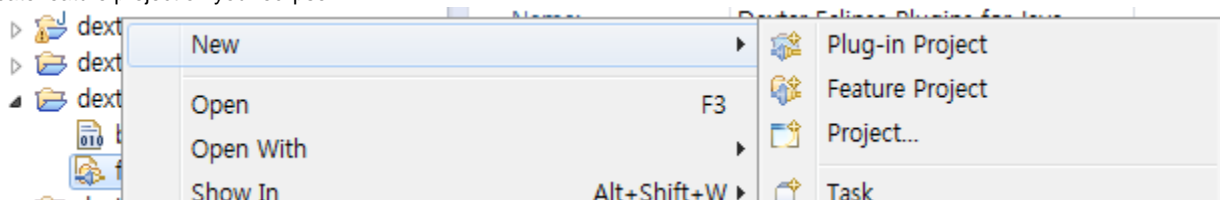




- now, reopen your project that you are working on
- after editing and saving a source file and if there is defects, you can see the error mark.

5. Build Dexter Eclipse Plug-ins

- create feature project on your eclipse



New Feature

Feature Properties
Define properties that will be placed in the feature.xml file

Project name:

☒ Use default location

Location:

Feature properties

Feature ID:

Feature Name:

Feature Version:

Feature Vendor:

Install Handler Library:

- open and edit 'feature.xml' file, then click the "

Package Explorer: **dexter.eclipse.feature.64**

- dexter-core
- dexter-cppcheck
- dexter-daemon
- dexter-eclipse
- dexter-eclipse-help
- dexter-eclipse-ui
- dexter-executor
- dexter-findbugs
- dexter-metrics
- dexter-daemon.feature.64
- dexter.eclipse.feature.64**
 - build.properties
 - feature.xml**
 - dexter.eclipse.update.64

Dexter Eclipse Plugins for Java

General Information
This section describes general information about this feature:

ID:

Version:

Name:

Vendor:

Branding Plug-in:

Update Site URL:

Update Site Name:

Supported Environments
Specify environment combinations in which this feature can be installed. Leave blank if the feature does not contain platform-specific code.

Operating Systems:

Window Systems:

Languages:

Architecture:

Feature Content
The content of the feature is made up of five sections:

- Information:** holds information about this feature, such as description and license.
- Plug-ins:** lists the plug-ins that make up this feature.
- Included Features:** lists the features that are included in this feature.
- Dependencies:** lists other features and plug-ins required by this feature when installed.
- Installation:** sets advanced installation options, declares an optional install handler and non-plugin data in a feature.

Exporting
To export the feature:

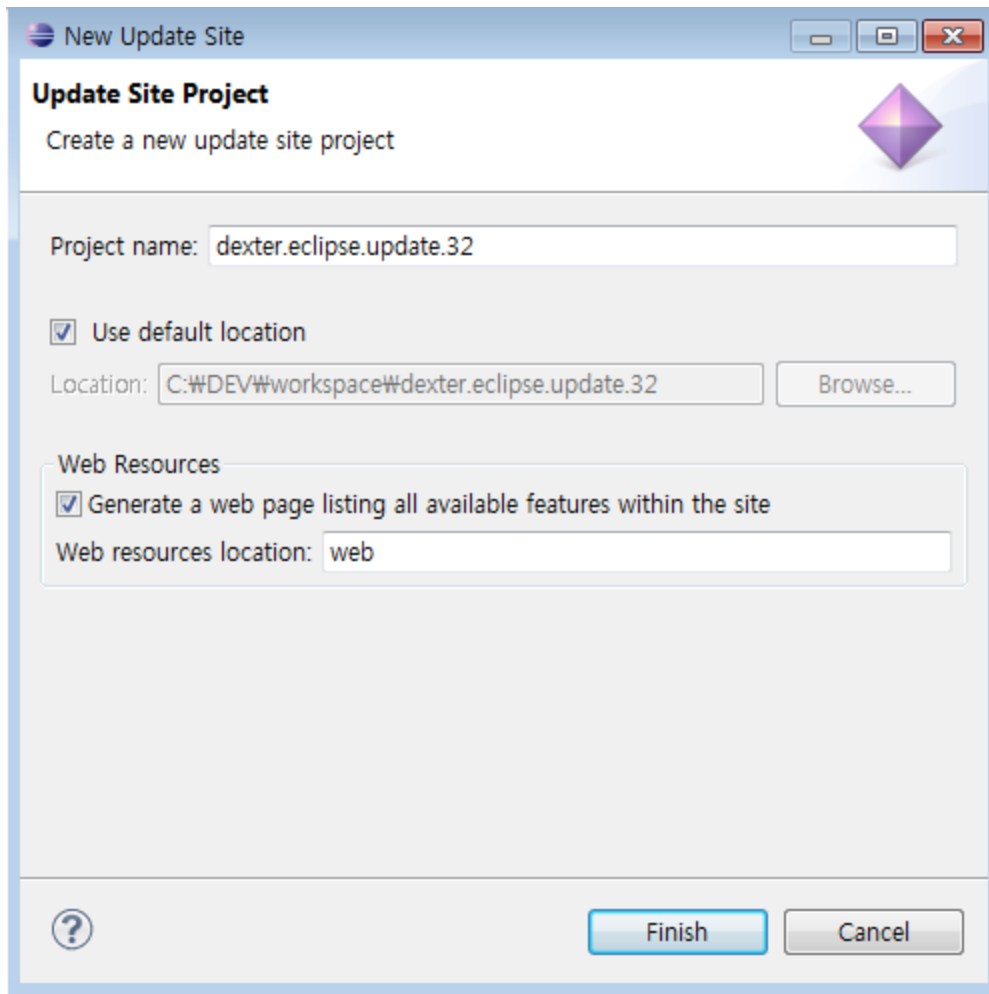
1. [Synchronize](#) versions of contained plug-ins and fragments with their version in the workspace
2. Specify what needs to be packaged in the feature archive on the [Build Configuration](#) page
3. Export the feature in a format suitable for deployment using the [Export Wizard](#)

Publishing
To publish the feature on an update site:

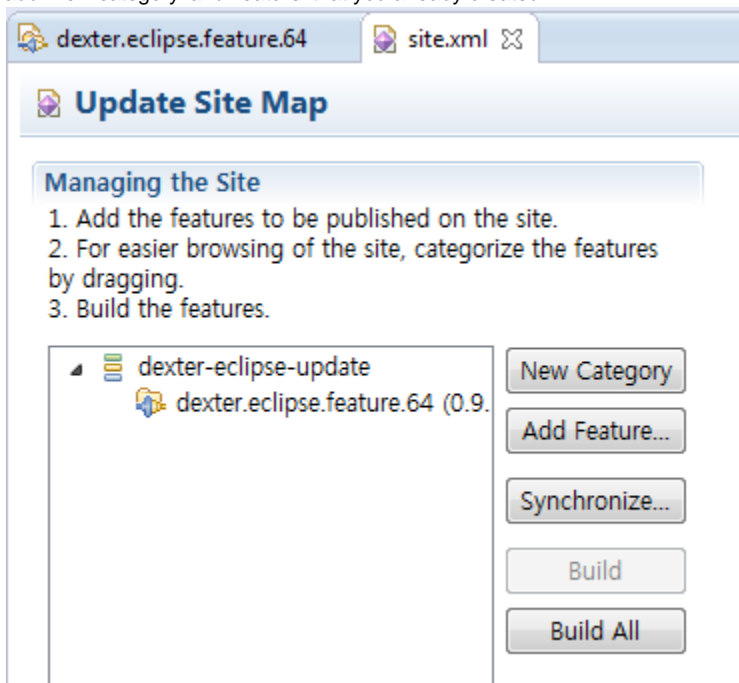
1. [Create an Update Site Project](#)
2. Use the site editor to add the feature to the site, and build the site

Overview | Information | Plug-ins | Included Features | Dependencies | Installation | Build | **feature.xml** | build.properties

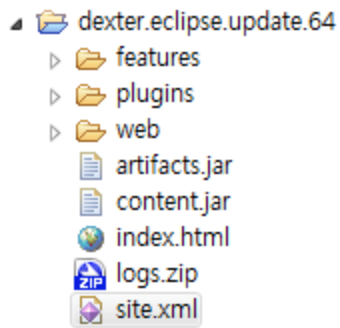
- click 'Update Site Project' on the "Publishing" tab
- then check 'Generate a web page listing all available features within the site'



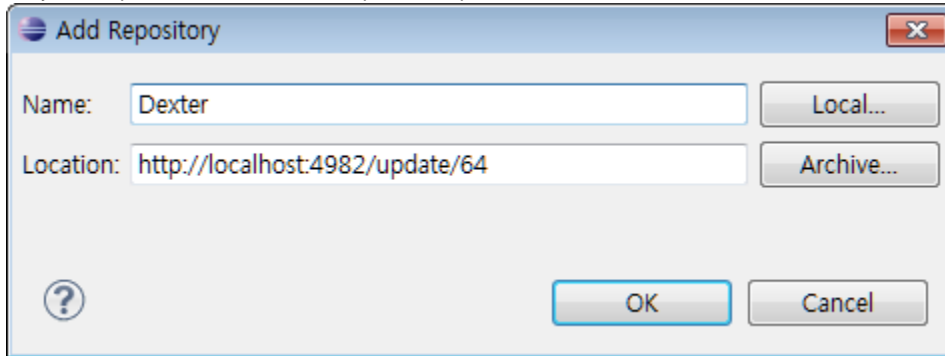
- open 'dexter.eclipse.update.64' or 32 project
- open 'site.xml' file
- add 'new category' and 'feature' that you already created



- then, click 'Build All' button
- now you have folders and files for eclipse update site

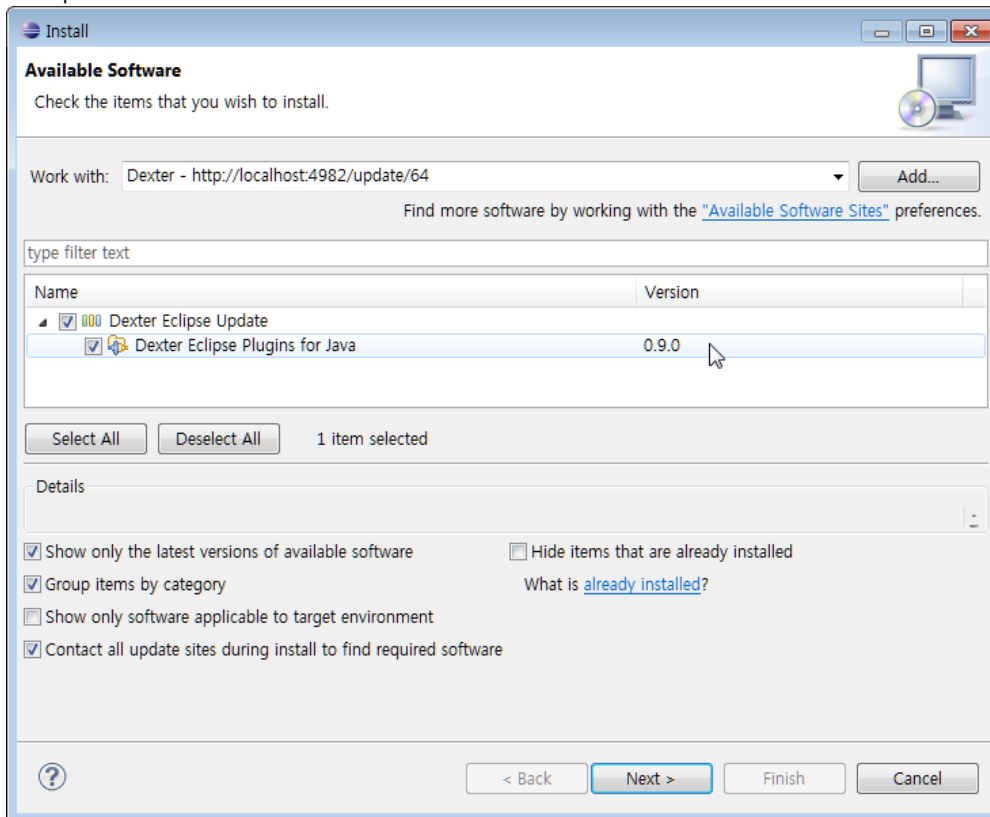


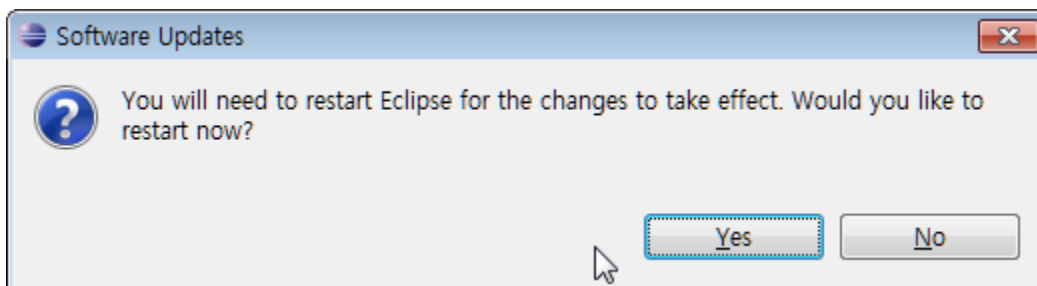
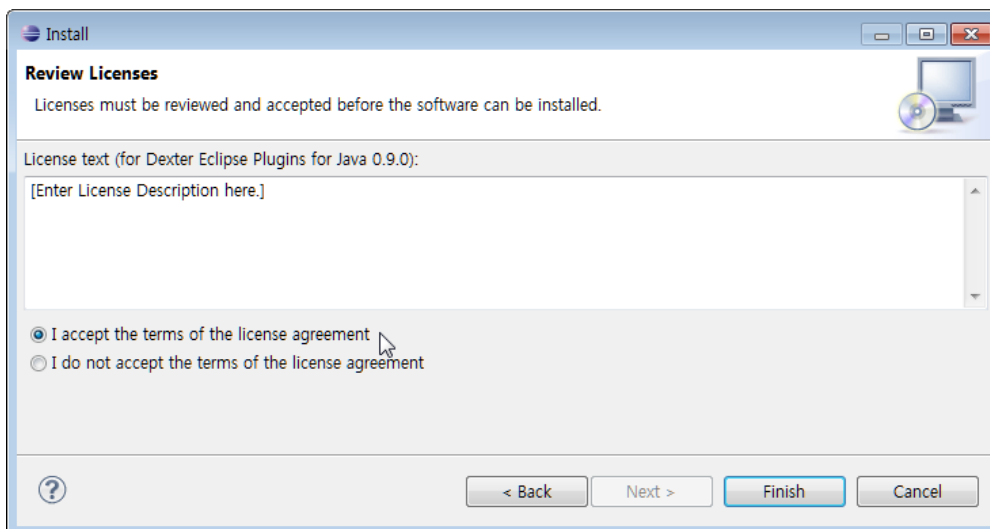
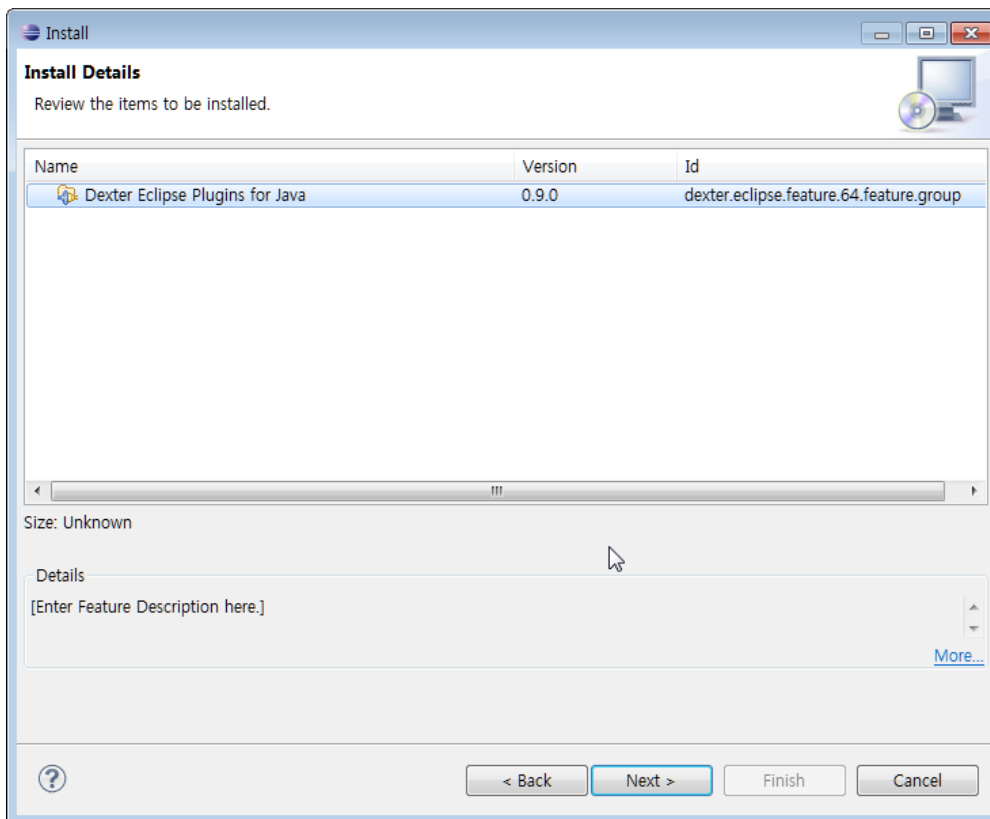
- you can make your own eclipse update site or just copy plugins folder to your eclipse
 - copy all folders and files into your web server, if you have
- run your eclipse, then click menu for update: Help > Install New Software... > click 'Add' button



(* I made my own update site)

- then update





- now login like Dexter Daemon version
- enjoy it ~

