SDK Quickstart

Set Eclpse for uDig Plug-in Development

Table of Contents

1 Introduction

This workbook is aimed at those doing plug-in development against the uDig platform. Follow these instructions to quickly set up a development environment for working on your own plug-ins.

If you would like to work on uDig itself please visit the project Policies and Procedures.

Eclipse is familiar to most developers as a Java Integrated Development Environment (IDE). The Eclipse IDE can be extended with additional “capabilities” to work with alternate programming languages (like C++ or Ruby), or additional subject matter such as Java Enterprise Edition or in this case Eclipse Plug-in development.

In this Quickstart we are going to use the the Eclipse Plug-in Development capability; with the uDIG SDK as the target platform. This workbook covers setting up a development environment for working on your own plug-ins.

If you have an existing Eclipse installation please do not skip this tutorial – we are going to very carefully set up a copy of Eclipse with a few more additional tools then you are perhaps used to.

The Eclipse developers themselves make use of the Plug-in Development Environment (PDE) day in and day out – so it has gotten a lot of polish over the years. In some respects it is more polished then the Java development environment (with custom editors for all kinds of little files).

Some of the terminology used when working with the PDE:

- Everything is a **Plug-in**
- A plug-in can implement an **extension**
- A plug-in can provide an **extension-point** for others

Please keep these ideas in mind – even an “Application” is considered an extension when working with the Eclipse Platform.

2 Downloads

We are going to start by downloading all the software we need; we will be able to proceed with installation as we wait for some of the larger downloads.

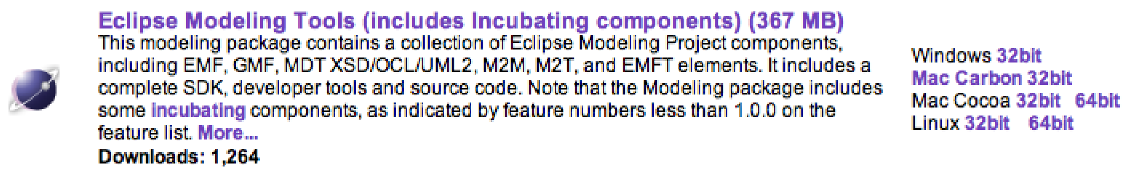
If you are using this work book in a lab setting you will find these downloads available on your course DVD.

1. For uDig 1.2 we are going going to download the latest SDK from here:
   http://udig.refractions.net/download/

   Normally you would grab the latest stable SDK from the public uDig download page.

   At the time of writing the latest uDig SDK was:
   udig-1.2.1-sdk.zip

2. Visit http://www.eclipse.org/downloads and click on the link for "Eclipse Modeling Tools" and download the appropriate

   Eclipse Modeling Tools (includes Incubating components) (367 MB)
   This modeling package contains a collection of Eclipse Modeling Project components, including EMF, GMF, MDT XSD/OCL/UML2, M2M, M2T, and EMFT elements. It includes a complete SDK, developer tools and source code. Note that the Modeling package includes some Incubating components, as indicated by feature numbers less than 1.0.0 on the feature list. More...
   Downloads: 1,264

   Windows 32bit
   Mac Carbon 32bit
   Mac Cocoa 32bit 64bit
   Linux 32bit 64bit

   Tested with **Eclipse 3.6.1 Helios Packages**:
   eclipse-modeling-helios-SR1-incubation-win32.zip

3. We have prepared an "dropins" download in the following folder:
   http://udig.refractions.net/files/downloads/extras/

   This download includes a developers guide for udig, platform language packs, and a resource bundle editor.

   At the time of writing: **dropins-3.6.1.zip**

GDAL support is provided by the imageio-ext project started by GeoTools alumni.

4. Download a Java Runtime Environment from this folder:
   http://udig.refractions.net/files/downloads/jre/

   This is a special JRE that has been extended with Java Advanced Imaging and Image IO and GDAL for spatial image formats.

   At the time of writing: **jre1.6.0_17.win32_gdal_ecw.zip**

5. Download Eclipse RCP Delta Pack

   Download the RCP-Delta Pack from http://download.eclipse.org/eclipse/downloads/ you will need to choose the "Latest Release" build; and the scroll down to the "Delta Pack" link to download.

   Eclipse-3.6.1-delta-pack.zip at the time of writing

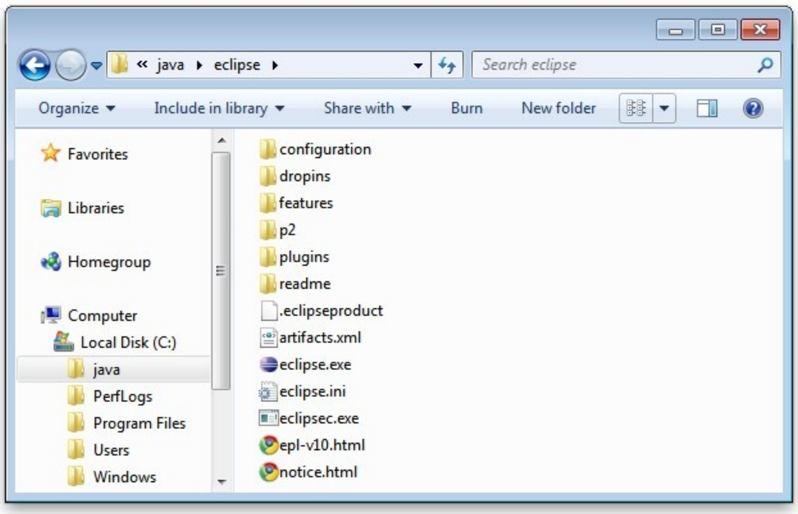
3 Eclipse SDK Installation

Hopefully by now your eclipse download has finished and we can begin to installation.

1. Create a folder: C:\java

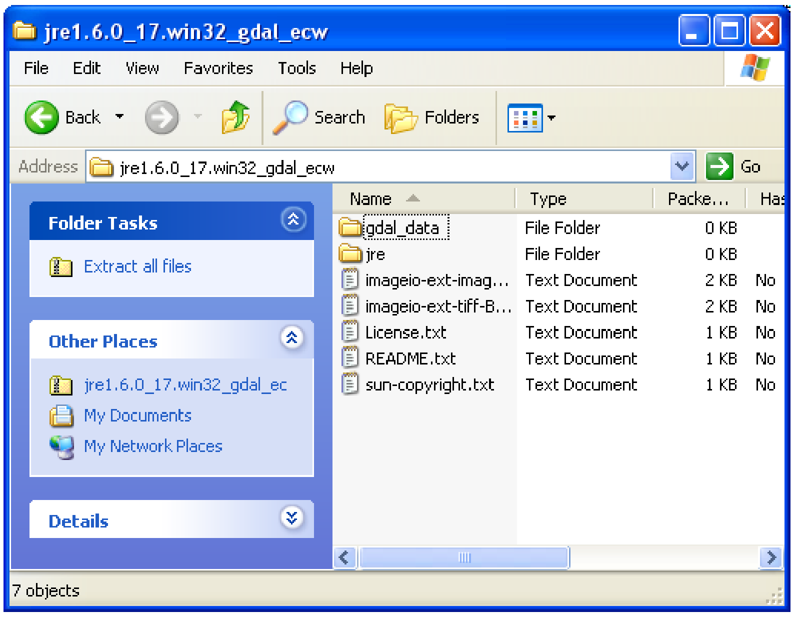
If you need a good program to unzip archive files try: www.7-zip.org

2. Unzip the downloaded **eclipse-modeling-helios-SR1-incubation-win32.zip** file to your C:\java directory – the folder **C:\java\eclipse** will be created



```
configuration
dropins
features
p2
plugins
readme
.eclipseproduct
artifacts.xml
eclipse.exe
eclipse.ini
eclipsec.exe
epl-v10.html
notice.html
```

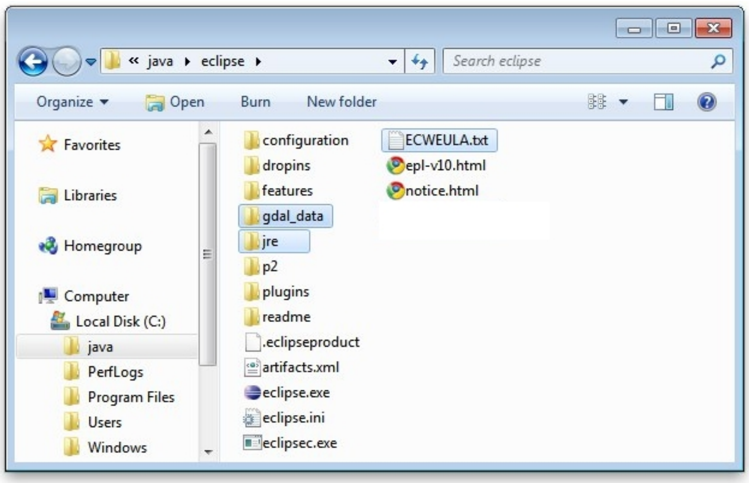
3. Open up the jre zip file Drag from the 7zip window.



Name	Type	Packe…	Has
gdal_data	File Folder	0 KB	
jre	File Folder	0 KB	
imageio-ext-imag…	Text Document	2 KB	No
imageio-ext-tiff-B…	Text Document	2 KB	No
License.txt	Text Document	1 KB	No
README.txt	Text Document	1 KB	No
sun-copyright.txt	Text Document	1 KB	No

The folder must be called "jre" for the eclipse.exe to recognize it

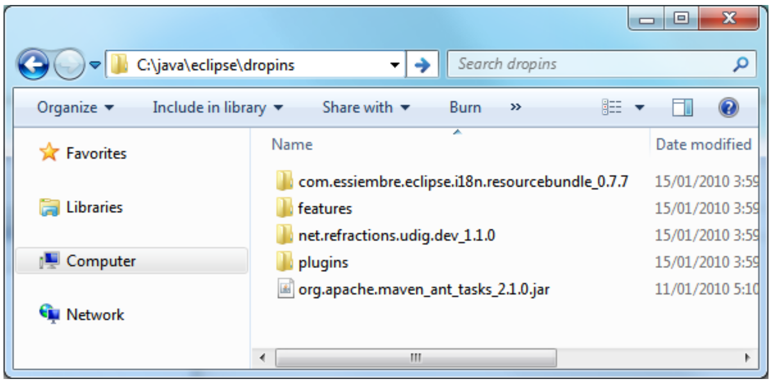
4. Drop into your C:\java\eclipse



configuration
dropins
features
gdal_data
jre
p2
plugins
readme
.eclipseproduct
artifacts.xml
eclipse.exe
eclipse.ini
eclipsec.exe
ECWEULA.txt
epl-v10.html
notice.html

Be careful when extracting this zip not to get something silly like dropins/dropins

5. Unzip the **dropins-3.6.1.zip** file to your eclipse/dropins folder. The download will add additional plug-ins and features to to your eclipse installation without getting mixed up with the bare-bones eclipse.



Name	Date modified
com.essiembre.eclipse.i18n.resourcebundle_0.7.7	15/01/2010 3:59
features	15/01/2010 3:59
net.refractions.udig.dev_1.1.0	15/01/2010 3:59
plugins	15/01/2010 3:59
org.apache.maven_ant_tasks_2.1.0.jar	11/01/2010 5:10

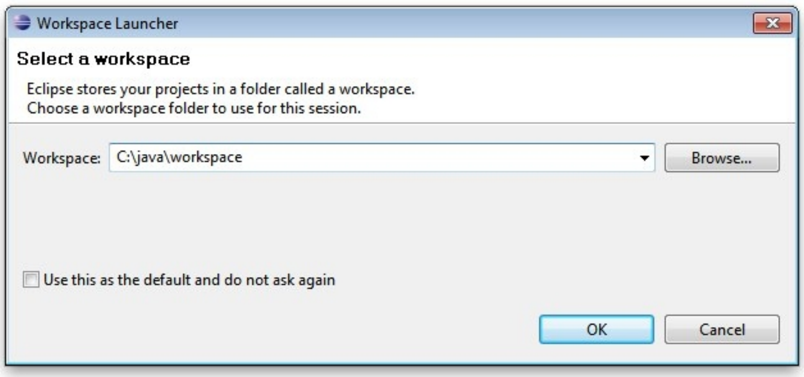
You may want to modify the "eclipse.ini" file to specify additional command line options such as more memory.

6. Navigate to **C:\java\eclipse** and right-click on the **eclipse.exe** file and select **Send To->Desktop (create shortcut)**.

4 Eclipse Workspace

The Eclipse IDE keeps track of what you are doing in a workspace. You can have several workspaces (often for different projects) each with its own configuration.

1. Double click on your desktop short cut to start up eclipse. When you start up eclipse for the first time it prompt you for a workspace.
2. Choose a workspace for your sdk development:
   C:\java\workspace



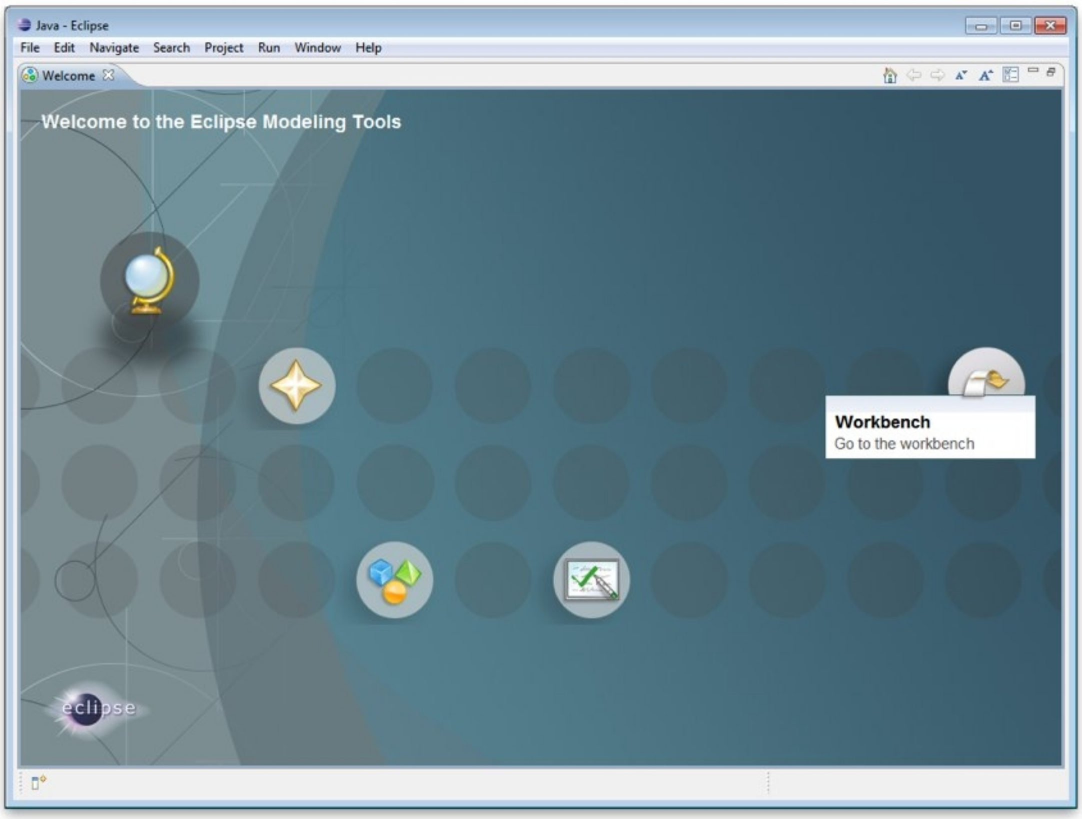
Workspace Launcher

Select a workspace

Eclipse stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: C:\java\workspace

3. Wait a few moments while eclipse starts up.
4. On the Welcome view press the **Workbench** button along the right hand side.



Java - Eclipse
File Edit Navigate Search Project Run Window Help
Welcome
Welcome to the Eclipse Modeling Tools
Workbench
Go to the workbench
eclipse

5 Eclipse Preferences

We have a few global settings to configure before we can proceed.

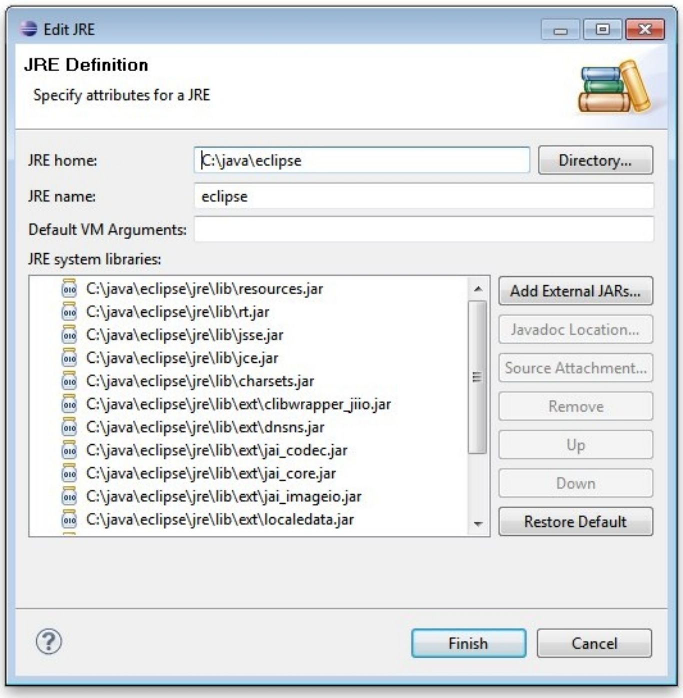
We are waiting for a Mac OSX JRE to be available before using Java 6.

1. Open up **Window > Preferences** from the menu bar.
2. Navigate to the **Java > Compiler** page and change:
   - Compiler compliance level: 1.5
3. Check the **Java > Installed JREs** page:
   - Should have Location: C:\java\eclipse

     If the setting is not correct you can use the **Add..** button and create a JRE entry for C:\java\eclipse
   - You can press **Edit** to look at the installed JRE.

     Regardless of platform we are interested in making sure **jai_core.jar, jai_imageio.jar** and **jai_codec.jar** are available.

On OSX we are not able to provide a JRE for you to download – uDig will install JAI and ImageIO when run for the first time.



JRE home: C:\java\eclipse

JRE name: eclipse

Default VM Arguments:

JRE system libraries:

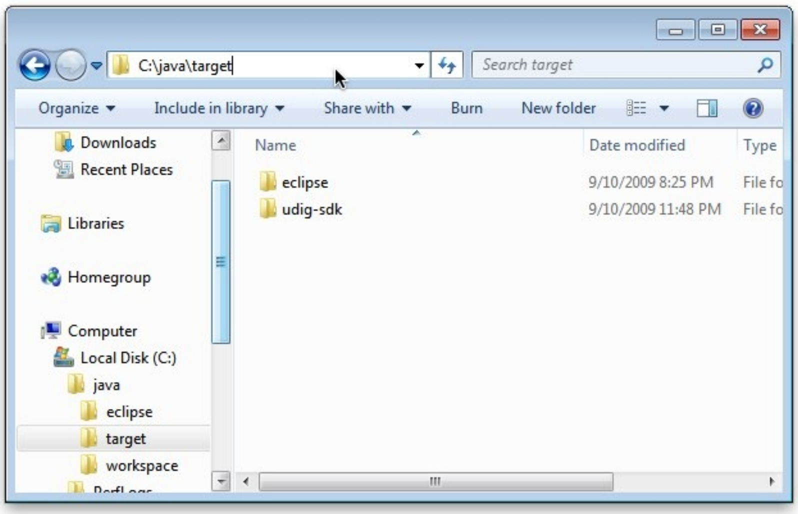
```
C:\java\eclipse\jre\lib\resources.jar
C:\java\eclipse\jre\lib\rt.jar
C:\java\eclipse\jre\lib\jsse.jar
C:\java\eclipse\jre\lib\jce.jar
C:\java\eclipse\jre\lib\charsets.jar
C:\java\eclipse\jre\lib\ext\clibwrapper_jiio.jar
C:\java\eclipse\jre\lib\ext\dnsns.jar
C:\java\eclipse\jre\lib\ext\jai_codec.jar
C:\java\eclipse\jre\lib\ext\jai_core.jar
C:\java\eclipse\jre\lib\ext\jai_imageio.jar
C:\java\eclipse\jre\lib\ext\localedata.jar
```

   - As a result of these changes, Eclipse may prompt you to perform a clean build. Simply accept this request; it won't take long since we don't yet have any source code.

6 uDIG SDK

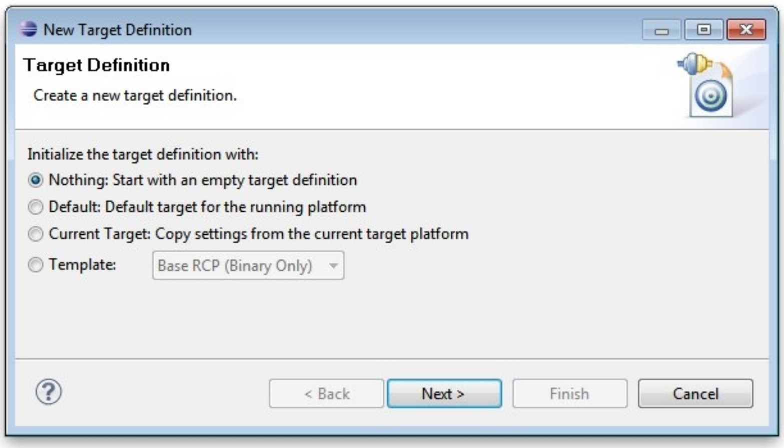
We are now going to unpack the udig sdk and use it as our plug-in target platform.

1. Extract the udig sdk download into C:\java\target
2. Extract the eclipse-delta-pack download into C:\java\target
3. You will be left with the following directory structure:



Name	Date modified	Type
eclipse	9/10/2009 8:25 PM	File fo
udig-sdk	9/10/2009 11:48 PM	File fo

4. Go back to eclipse and open **Window>Preferences**.
5. Select the **Plugin Development > Target Platform** page.
6. Press the **Add** button to start a new target definition
7. Choose **Nothing** as we are going to define a target from scratch



Target Definition

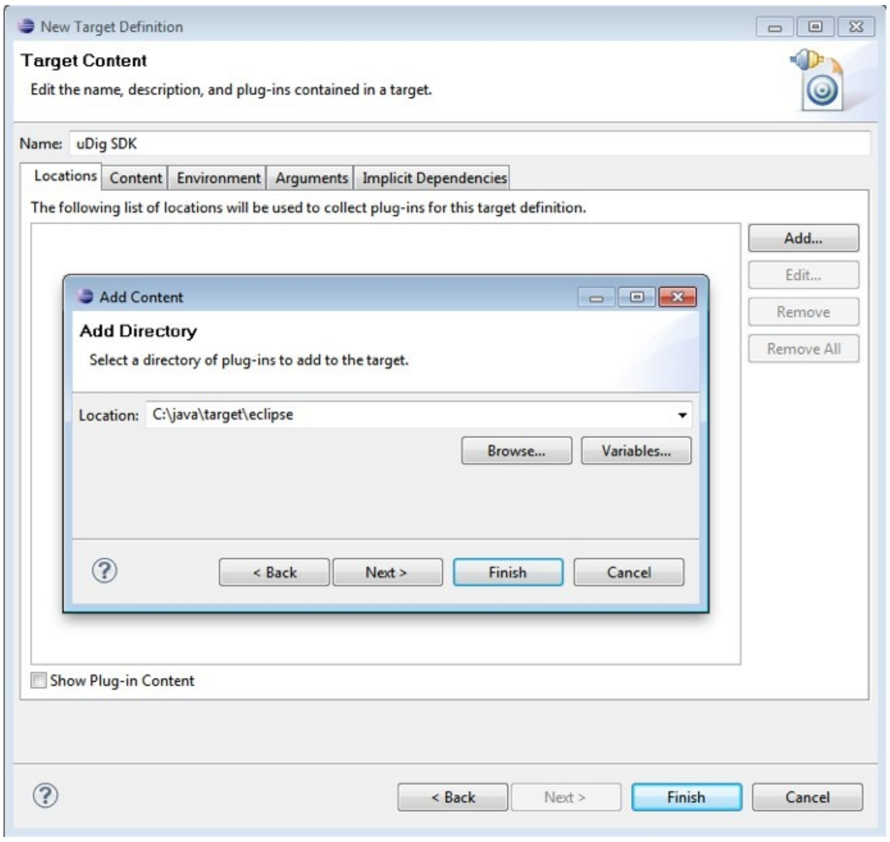
Create a new target definition.

Initialize the target definition with:

- [✓ Nothing: Start with an empty target definition]
- Default: Default target for the running platform
- Current Target: Copy settings from the current target platform
- Template: Base RCP (Binary Only)

8. Fill in the name “uDig SDK” for this target

9. Press the **Add** button and add a Directory. Choose the C:\java\target\eclipse directory where you unpacked the eclipse-delta-pack


New Target Definition

Target Content

Edit the name, description, and plug-ins contained in a target.

Name: uDig SDK

The following list of locations will be used to collect plug-ins for this target definition.

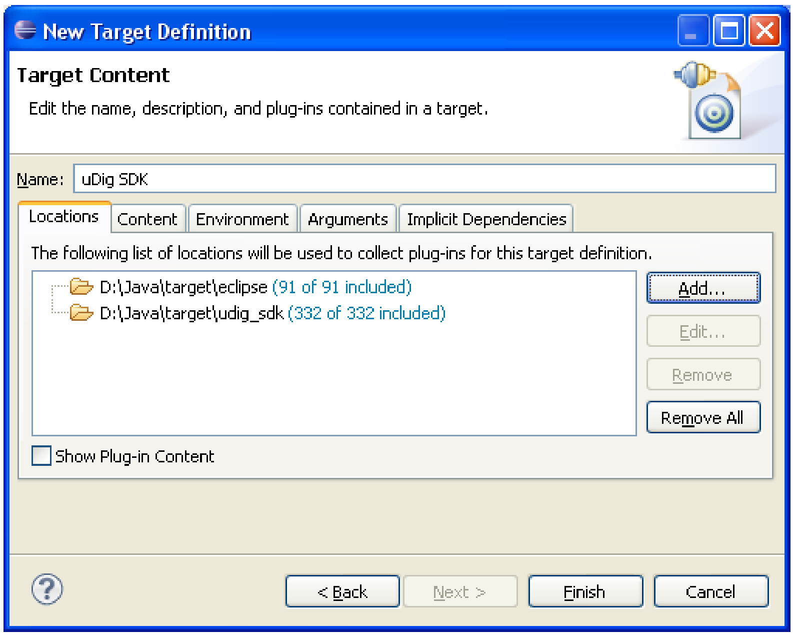
Add Content

Add Directory

Select a directory of plug-ins to add to the target.

Location: C:\java\target\eclipse

10. Press **Add** again and add the C:\java\target\udig-sdk directory.


New Target Definition

Target Content

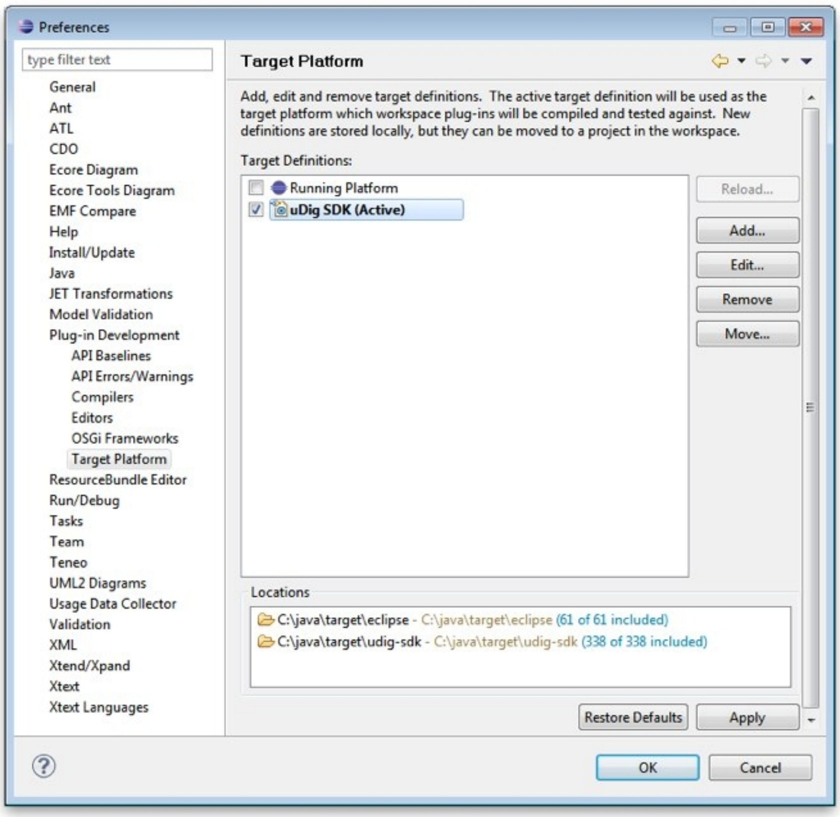
Edit the name, description, and plug-ins contained in a target.

Name: uDig SDK

The following list of locations will be used to collect plug-ins for this target definition.

D:\Java\target\eclipse (91 of 91 included)
D:\Java\target\udig_sdk (332 of 332 included)

11. Press Finish to complete the “uDig SDK” target platform. Please tick the checkbox next to the “uDig SDK” so it will be the Active target platform.



Add, edit and remove target definitions. The active target definition will be used as the target platform which workspace plug-ins will be compiled and tested against. New definitions are stored locally, but they can be moved to a project in the workspace.

Target Definitions:

☐ Running Platform
☑ uDig SDK (Active)

Locations

C:\java\target\eclipse - C:\java\target\eclipse (61 of 61 included)
C:\java\target\udig-sdk - C:\java\target\udig-sdk (338 of 338 included)

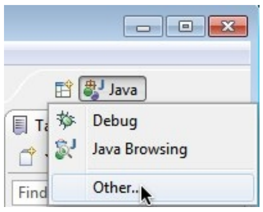
12. Press **OK**

At this point all the source code for the Eclipse and uDig plug-ins are available. We can now start working on uDig plug-ins, but before we do that lets try running the application.

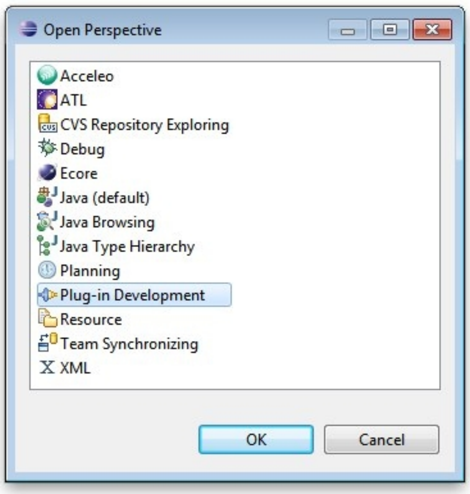
7 Running uDig

With all this in place we can now run the uDig application from your development environment. This is a good way to test that everything is installed correctly.

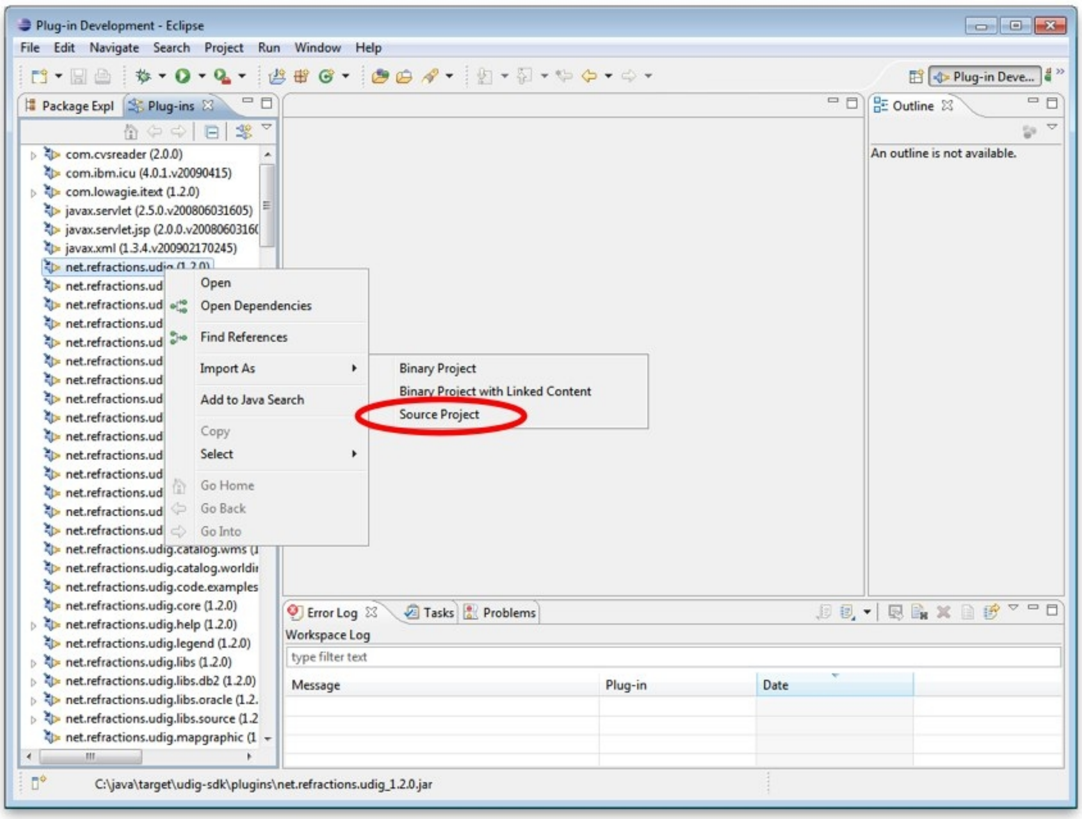
1. To start out with we will switch to the **Plug-in Development** perspective; in the top right of the toolbar you can choose **Other** to open the **Open Perspective** dialog



2. Chose **Plugin-Development** from the list and press **OK**

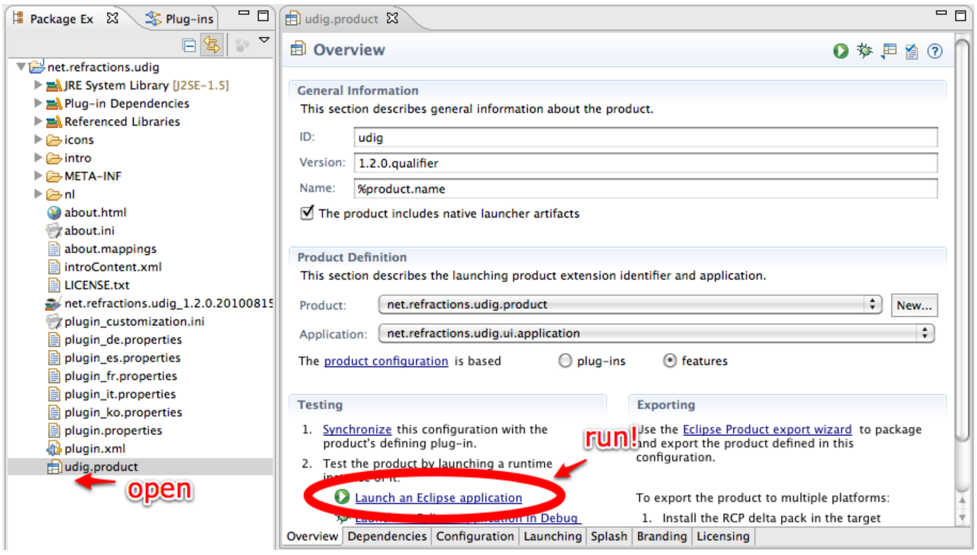


3. Click on the **Plugins** view and right click on **net.refractions.udig** plug-in the list. Select **Import As > Source Project** to copy the plugin into your workspace.

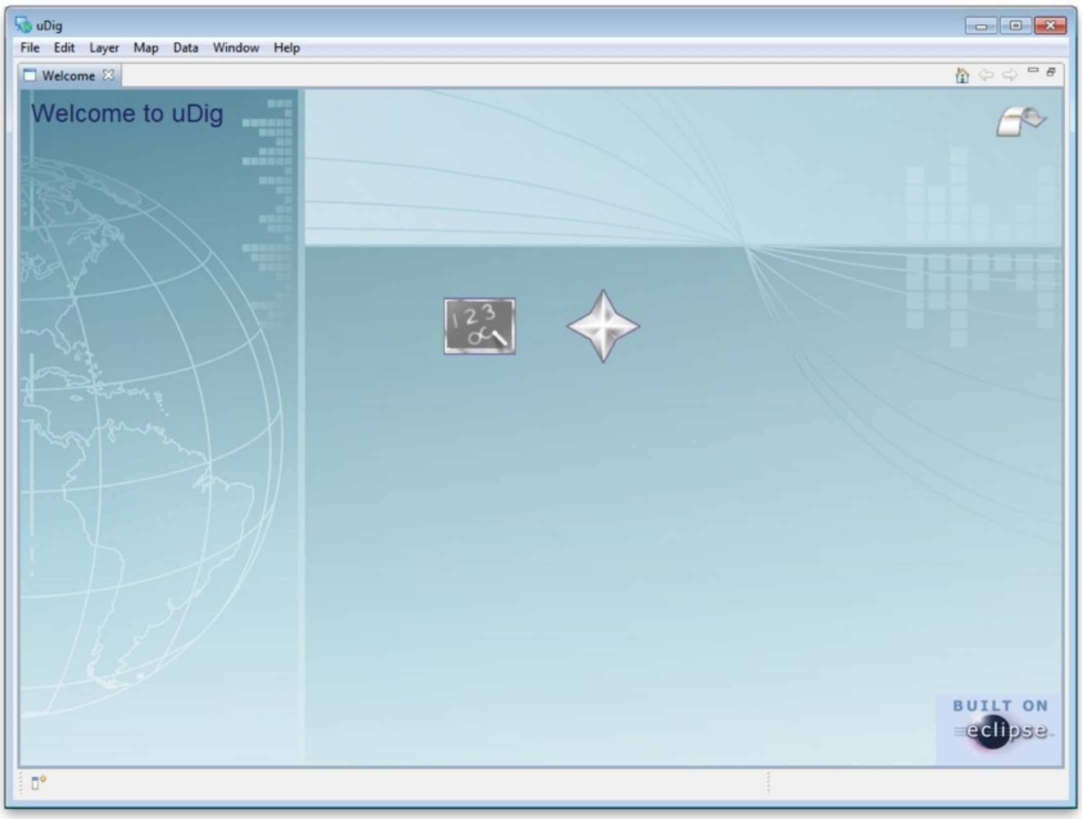

Plug-in Development - Eclipse
File Edit Navigate Search Project Run Window Help
Package Expl
Plug-ins
Outline
An outline is not available.
com.cvsreader (2.0.0)
com.ibm.icu (4.0.1.v20090415)
com.lowagie.itext (1.2.0)
javax.servlet (2.5.0.v200806031605)
javax.servlet.jsp (2.0.0.v200806031607)
javax.xml (1.3.4.v200902170245)
net.refractions.udig (1.2.0)
net.refractions.udig.catalog.wms
net.refractions.udig.catalog.worldimage
net.refractions.udig.code.examples
net.refractions.udig.core (1.2.0)
net.refractions.udig.help (1.2.0)
net.refractions.udig.legend (1.2.0)
net.refractions.udig.libs (1.2.0)
net.refractions.udig.libs.db2 (1.2.0)
net.refractions.udig.libs.oracle (1.2.0)
net.refractions.udig.libs.source (1.2.0)
net.refractions.udig.mapgraphic (1.2.0)
Open
Open Dependencies
Find References
Import As
Add to Java Search
Copy
Select
Go Home
Go Back
Go Into
Binary Project
Binary Project with Linked Content
Source Project
Error Log
Tasks
Problems
Workspace Log
type filter text
Message
Plug-in
Date
C:\java\target\udig-sdk\plugins\net.refractions.udig_1.2.0.jar

4. Change to the **Package Explorer** view and open up the net.refractions.udig plugin and double click to open **udig.product**.

5. From the overview tab of udig.product click on the **Launch Eclipse Application** link (it is located under testing as shown below).


Package Ex
Plug-ins
net.refractions.udig
JRE System Library [J2SE-1.5]
Plug-in Dependencies
Referenced Libraries
icons
intro
META-INF
nl
about.html
about.ini
about.mappings
introContent.xml
LICENSE.txt
net.refractions.udig_1.2.0.20100815
plugin_customization.ini
plugin_de.properties
plugin_es.properties
plugin_fr.properties
plugin_it.properties
plugin_ko.properties
plugin.properties
plugin.xml
udig.product
open
udig.product
Overview
General Information
This section describes general information about the product.
ID: udig
Version: 1.2.0.qualifier
Name: %product.name
The product includes native launcher artifacts
Product Definition
This section describes the launching product extension identifier and application.
Product: net.refractions.udig.product
New...
Application: net.refractions.udig.ui.application
The product configuration is based ( ) plug-ins (•) features
Testing
1. Synchronize this configuration with the product's defining plug-in.
2. Test the product by launching a runtime instance of it.
Launch an Eclipse application
Launch an Eclipse application in Debug
run!
Exporting
Use the Eclipse Product export wizard to package and export the product defined in this configuration.
To export the product to multiple platforms:
1. Install the RCP delta pack in the target
Overview Dependencies Configuration Launching Splash Branding Licensing

6. The application will now start!

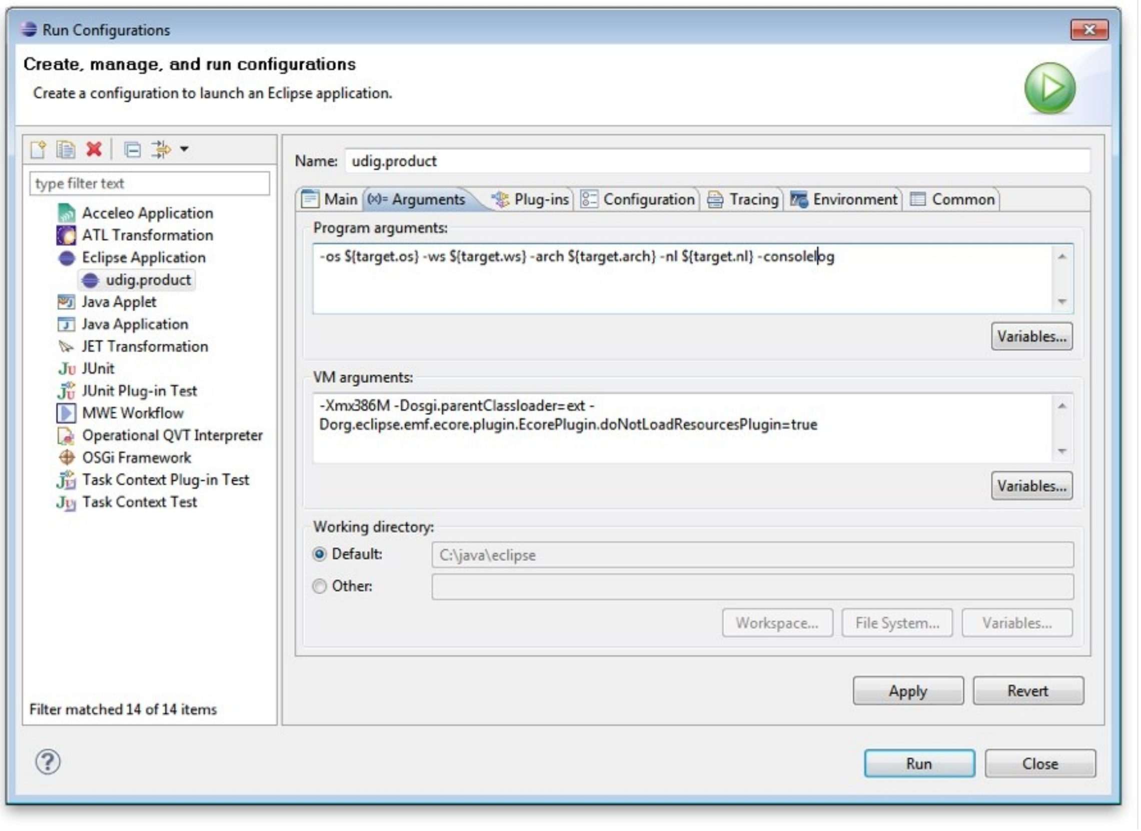


8 What to Do Next

Here are some additional things to try when running uDig.

- From Eclipse open up **Run > Run Configurations** to examine or customize configuration of uDig you are running. Many of these fields were filled in for you by the udig.product.
- The number one tip is to go to the Arguments tab and enable console logging. To send log information to the console as udig runs add **-consolelog** to your “program arguments”.

For documentation on these command line parameters check the eclipse help menu.



Create, manage, and run configurations

Create a configuration to launch an Eclipse application.

type filter text

Filter matched 14 of 14 items

Name: udig.product

Program arguments:

```
-os ${target.os} -ws ${target.ws} -arch ${target.arch} -nl ${target.nl} -consolelog
```

VM arguments:

```
-Xmx386M -Dosgi.parentClassloader=ext -
Dorg.eclipse.emf.ecore.plugin.EcorePlugin.doNotLoadResourcesPlugin=true
```

Working directory:

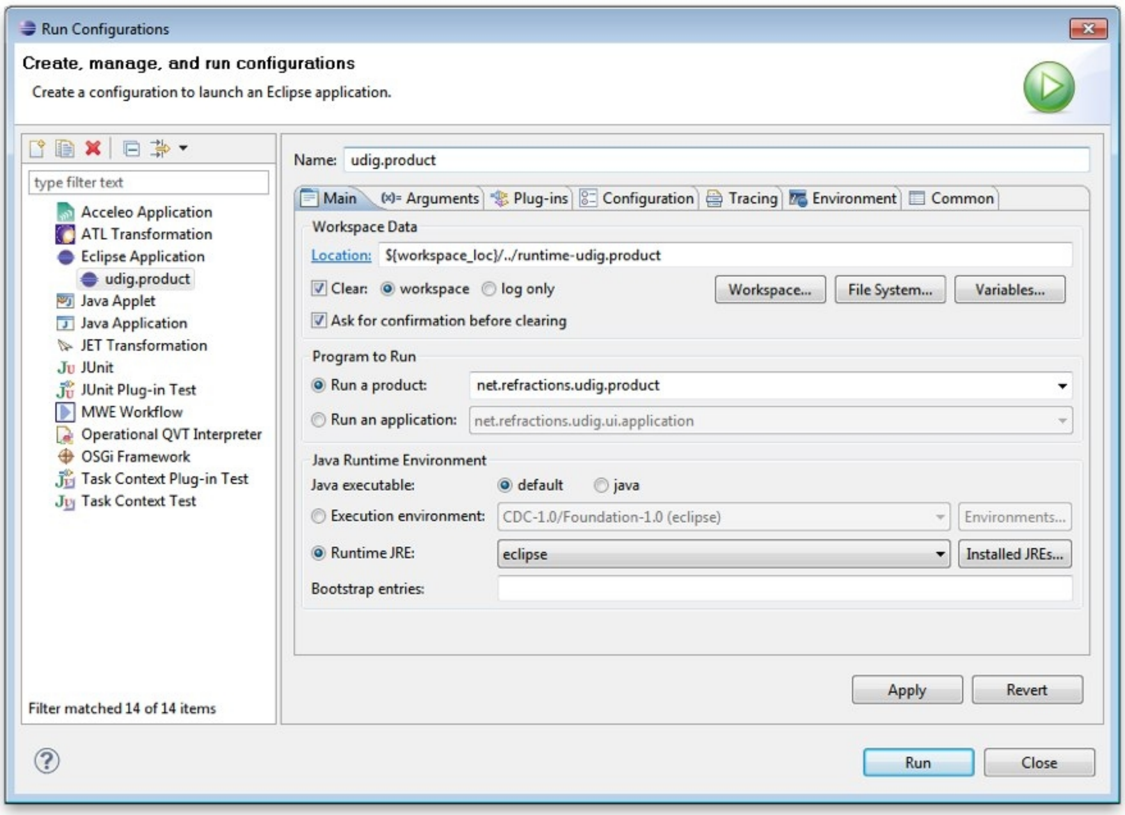
[✓ Default:] C:\java\eclipse

Other:

You can also review the VM arguments; including changing the amount of memory available to your uDig application **-Xmx512m** may be useful when working with large images?

The **-Dosgi.parentClassload=ext** setting allows uDig to find JRE extensions such as Java Advanced Imaging and ImageIO.

- The running uDig application makes use of the “Workspace Data” folder defined in the Run dialog. Try checking **clear** and workspace in order to simulate starting uDig from a fresh install.



Run Configurations

Create, manage, and run configurations

Create a configuration to launch an Eclipse application.

Name: udig.product

Workspace Data

Location: ${workspace_loc}/../runtime-udig.product

[✓] Clear: [● workspace] ○ log only

[✓] Ask for confirmation before clearing

Program to Run

● Run a product: net.refractions.udig.product

○ Run an application: net.refractions.udig.ui.application

Java Runtime Environment

Java executable: [● default] ○ java

○ Execution environment: CDC-1.0/Foundation-1.0 (eclipse)

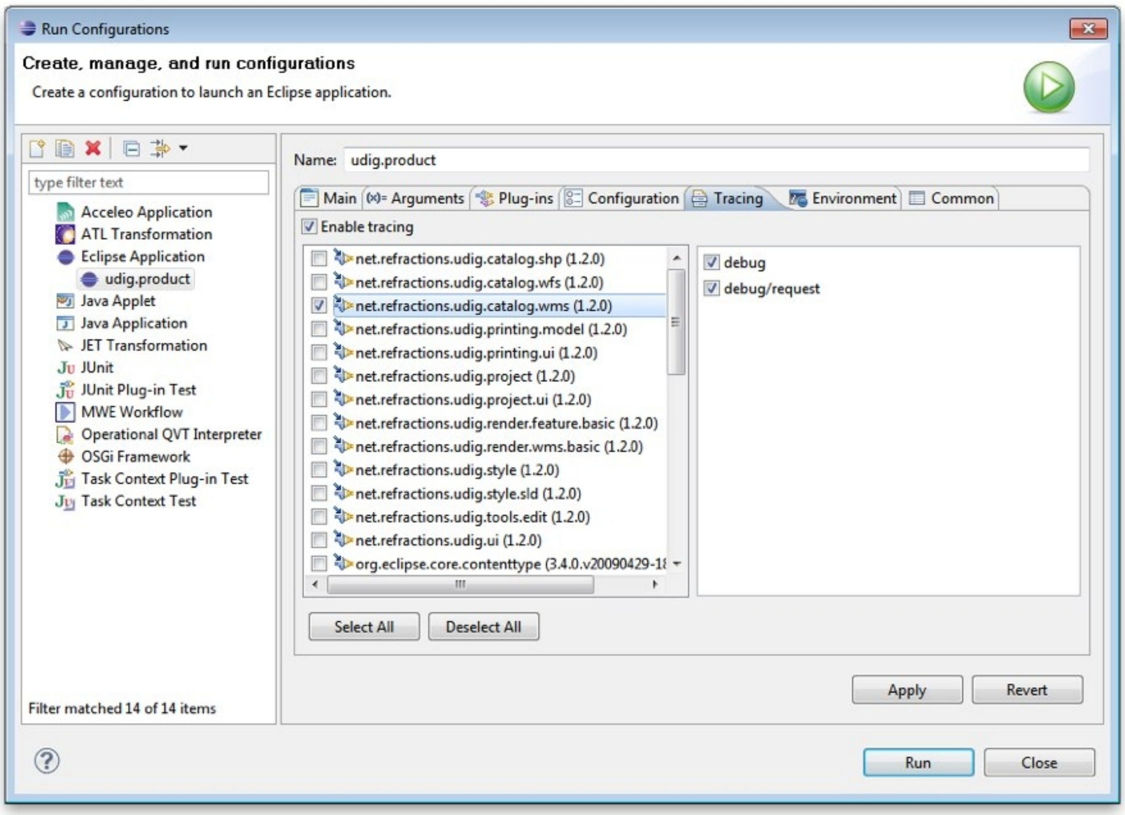
● Runtime JRE: eclipse

Bootstrap entries:

Filter matched 14 of 14 items

- Have a look on the Tracing tab of the Run dialog; you can control the amount of logging information produced (for example WMS logging is turned on below).

Tracing is especially useful when the -consolelog program argument is used.



Run Configurations

Create, manage, and run configurations

Create a configuration to launch an Eclipse application.

Name: udig.product

[✓] Enable tracing

- [ ] net.refractions.udig.catalog.shp (1.2.0)
- [ ] net.refractions.udig.catalog.wfs (1.2.0)
- [✓] net.refractions.udig.catalog.wms (1.2.0)
- [ ] net.refractions.udig.printing.model (1.2.0)
- [ ] net.refractions.udig.printing.ui (1.2.0)
- [ ] net.refractions.udig.project (1.2.0)
- [ ] net.refractions.udig.project.ui (1.2.0)
- [ ] net.refractions.udig.render.feature.basic (1.2.0)
- [ ] net.refractions.udig.render.wms.basic (1.2.0)
- [ ] net.refractions.udig.style (1.2.0)
- [ ] net.refractions.udig.style.sld (1.2.0)
- [ ] net.refractions.udig.tools.edit (1.2.0)
- [ ] net.refractions.udig.ui (1.2.0)
- [ ] org.eclipse.core.contenttype (3.4.0.v20090429-1

[✓] debug

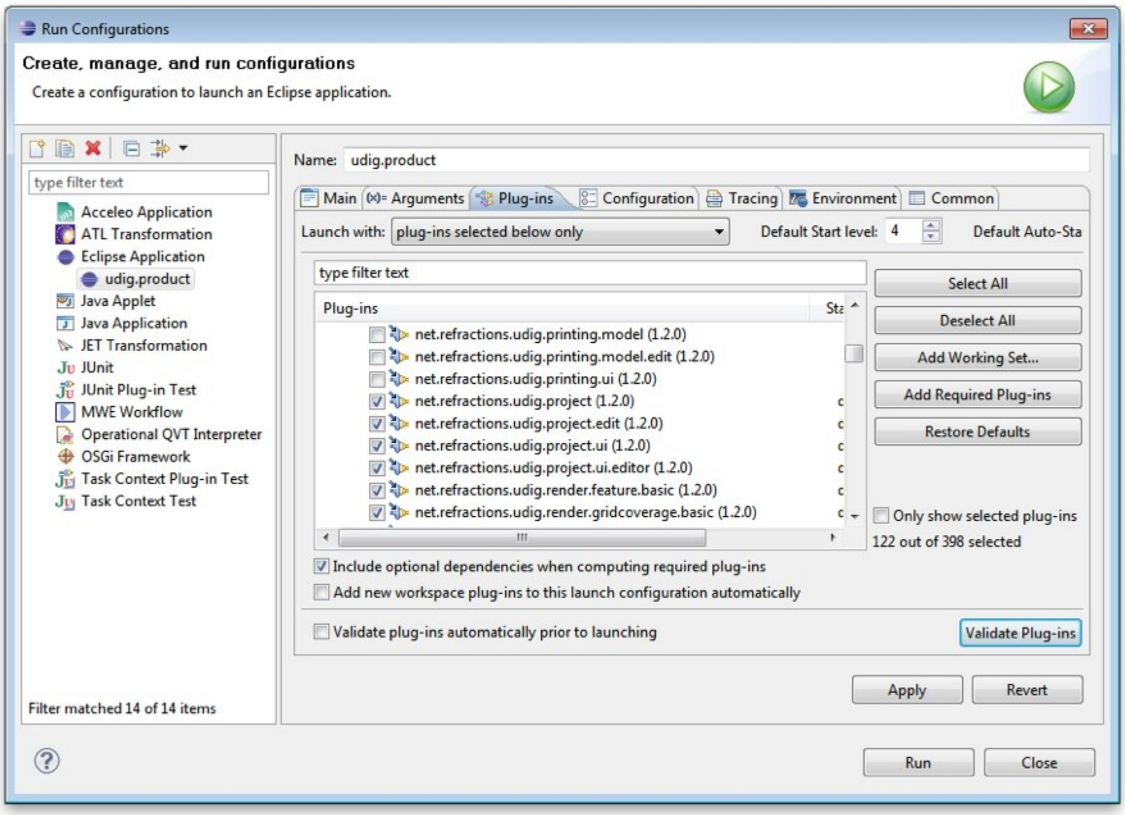
[✓] debug/request

Filter matched 14 of 14 items

-

Normally the plugins included by an application are organized into features – we will cover how to do this in the custom application tutorial.

- Have a look at the plug-ins tab and see if you can turn off: printing support.



Run Configurations

Create, manage, and run configurations

Create a configuration to launch an Eclipse application.

Name: udig.product

Launch with: plug-ins selected below only

Default Start level: 4

Plug-ins:

- [ ] net.refractions.udig.printing.model (1.2.0)
- [ ] net.refractions.udig.printing.model.edit (1.2.0)
- [ ] net.refractions.udig.printing.ui (1.2.0)
- [✓] net.refractions.udig.project (1.2.0)
- [✓] net.refractions.udig.project.edit (1.2.0)
- [✓] net.refractions.udig.project.ui (1.2.0)
- [✓] net.refractions.udig.project.ui.editor (1.2.0)
- [✓] net.refractions.udig.render.feature.basic (1.2.0)
- [✓] net.refractions.udig.render.gridcoverage.basic (1.2.0)

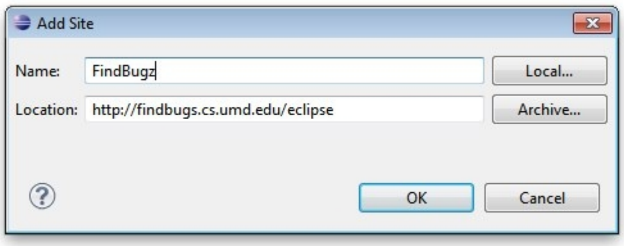
122 out of 398 selected

[✓] Include optional dependencies when computing required plug-ins

[ ] Add new workspace plug-ins to this launch configuration automatically

[ ] Validate plug-ins automatically prior to launching

- Advanced: When working on the uDig project itself we use the FindBugz tool to check for obvious mistakes prior to committing. You can add FindBugz to your environment using the following update site:
  http://findbugs.cs.umd.edu/eclipse



Add Site

Name: FindBugz

Location: http://findbugs.cs.umd.edu/eclipse