Informatics for Integrating Biology and the Bedside

Developer's Guide

i2b2 Workbench

Document Version: *1.4.1*

I2b2 Software Release: *1.4*

Table of Contents

About this Guide 4

1. Prerequisites 5

1.1 Third Party Software 5

1.1.1 Java JDK 5

1.1.2 Eclipse 5

1.2 i2b2 Software 5

1.2.1 i2b2 Workbench Source Code 5

2. Install and Configure 8

2.1 Setting up the i2b2 Workbench within Eclipse 3.5 8

2.1.1 i2b2 Workbench Project Files 8

2.1.2 Importing the i2b2 Workbench projects 9

2.2 Building the i2b2 Workbench within the Eclipse IDE 13

2.2.1 Java Compiler Settings 13

2.2.2 Cleaning and Building the Projects 15

2.3 Running the i2b2 Workbench within the Eclipse IDE 17

2.3.1 Define Run Configurations 17

2.3.1.1 Main Tab 19

2.3.1.2 Arguments Tab 19

2.3.1.3 Plug-ins tab 22

2.3.2 Running the i2b2 Workbench Application 22

2.3.2.1 Reconfiguring the default layout 26

3. Package Structure 27

4. Eclipse (i2b2) Feature 28

4.1 Creating a New "Feature" Project 28

4.2 Feature Manifest (feature.xml) 33

4.2.1 Opening the Feature Manifest Editor 34

4.2.2 Overview Tab 34

4.2.3 Information Tab 35

4.2.3.1 Feature Description 35

4.2.3.2 Copyright Notice 36

4.2.3.3 License Agreement 37

4.2.3.4 Sites to Visit 38

4.2.4 Plug-ins Tab 39

4.2.4.1 Adding and Removing Plug-ins and Fragments 40

4.2.4.2 Update Version of Plug-ins and Fragments 40

4.2.4.3 Update Plug-in Details 43

4.2.5 Included Features Tab 43

4.2.6 Dependencies Tab 44

4.2.6.1 Dependency Properties 45

4.2.7 Installation Tab 45

4.2.8 Build Tab 46

4.2.9 Feature.xml Tab 47

4.2.10 Build.properties Tab 48

License 49

ABOUT THIS GUIDE

Informatics for Integrating Biology and the Bedside (i2b2) is one of the sponsored initiatives of the NIH Roadmap National Centers for Biomedical Computing (http://www.bisti.nih.gov/ncbc/).

One of the goals of i2b2 is to provide clinical investigators broadly with the software tools necessary to collect and manage project-related clinical research data in the genomics age as a cohesive entity; a software suite to construct and manage the modern clinical research chart.

1. PREREQUISITES

1.1 Third Party Software

1.1.1 Java JDK

JDK 6.0 is recommended and can be downloaded from the java website:

http://java.sun.com/products/archive/

1. Install the SDK into a directory of your choice.

   Example: */opt/java/jdk1.6.0* or *YOUR_JAVA_HOME_DIR*

1.1.2 Eclipse

You will need to use version 3.5 of the Eclipse SDK (Galileo), available at: http://www.eclipse.org/downloads/ .

To install Eclipse, extract the downloaded zip file into a directory such as C:\

If you install a new version of Eclipse, be sure to install it in an area separate from any previous Eclipse installations.

Further information regarding the eclipse product can be found on their website (http://www.eclipse.org/documentation/).

1.2 i2b2 Software

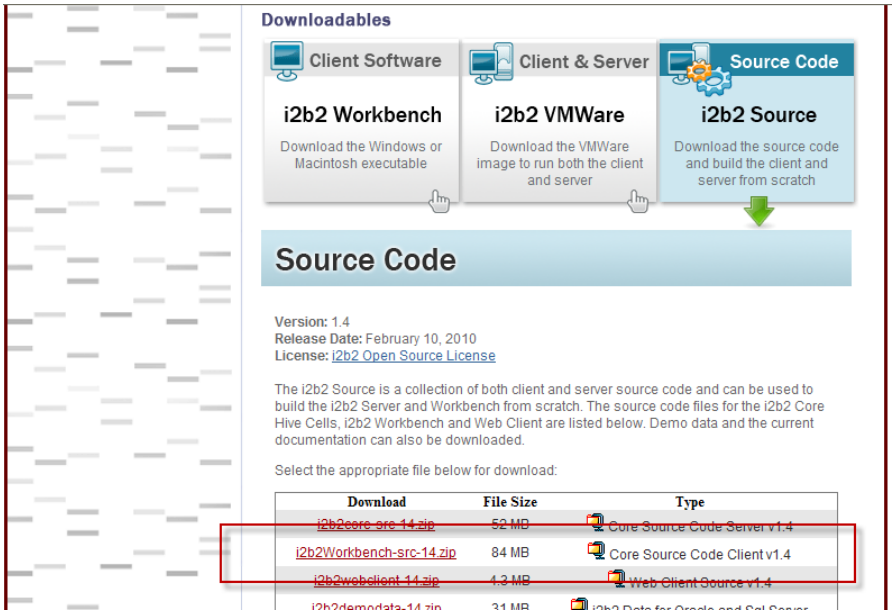
1.2.1 i2b2 Workbench Source Code

The latest version of the source code for the i2b2 Workbench needs to be downloaded.

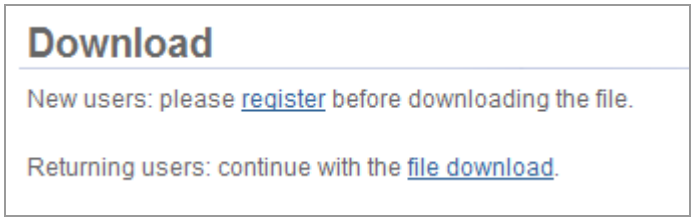
1. Go to the i2b2 website (http://www.i2b2.org)
2. At the top of the page there is a navigation bar, click on "**Software**" to go to the *software page*.

3. Once the software page loads, scroll down to the section called **Downloads**.
4. Click on the **Source Code** box.

5. Information about the latest version and a list of zip files available for downloading will appear.
6. Click on **i2b2Workbench-src-14.zip**.



7. The **Download** page will open. At this point, if you are not already a registered user you will need to do so. If you are a returning user you can click on the "**file download**" link.



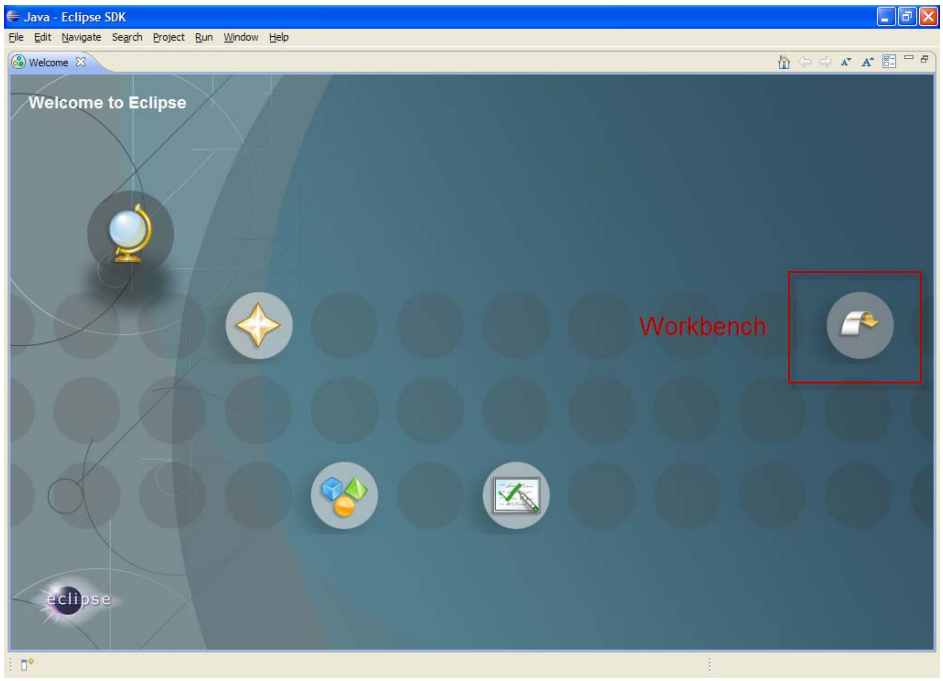
8. The next step is to review the i2b2 license and choose to either accept or decline.
    a. If you choose to **Decline** you will be brought back to the main i2b2 Software page.
    b. If you agree with the license click on **Accept** to continue with the download.

9. Save the zip file in a directory of your choice.

2. INSTALL AND CONFIGURE

2.1 Setting up the i2b2 Workbench within Eclipse 3.5

If launching eclipse for the first time, either via an eclipse application shortcut or *eclipse.exe*, you will be greeted by a Welcome page.



Click on the arrow labeled Workbench to startup the eclipse SDK.

2.1.1 i2b2 Workbench Project Files

The project software is contained in the Eclipse "*Archive*" file called **i2b2Workbench-src-14.zip**, which you downloaded in the previous section. In this file are the following Eclipse projects:

1. BuildAssist
2. edu.harvard.i2b2.common
3. edu.harvard.i2b2.eclipse
4. edu.harvard.i2b2.eclipse.featurs.core
5. edu.harvard.i2b2.eclipse.plugins.analysis

6. edu.harvard.i2b2.eclipse.plugins.commons_apache
7. edu.harvard.i2b2.eclipse.plugins.explorer
8. edu.harvard.i2b2.eclipse.plugins.jdnc_support
9. edu.harvard.i2b2.eclipse.plugins.log
10. edu.harvard.i2b2.eclipse.plugins.ontology
11. edu.harvard.i2b2.eclipse.plugins.previousQuery
12. edu.harvard.i2b2.eclipse.plugins.query
13. edu.harvard.i2b2.eclipse.plugins.webservicesAxis2
14. edu.harvard.i2b2.eclipse.plugins.workplace
15. edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb
16. edu.harvard.i2b2.eclipse.plugins.xml_support_jdom
17. edu.harvard.i2b2.xml

For additional information on the above projects please see the section called "Package Structure".

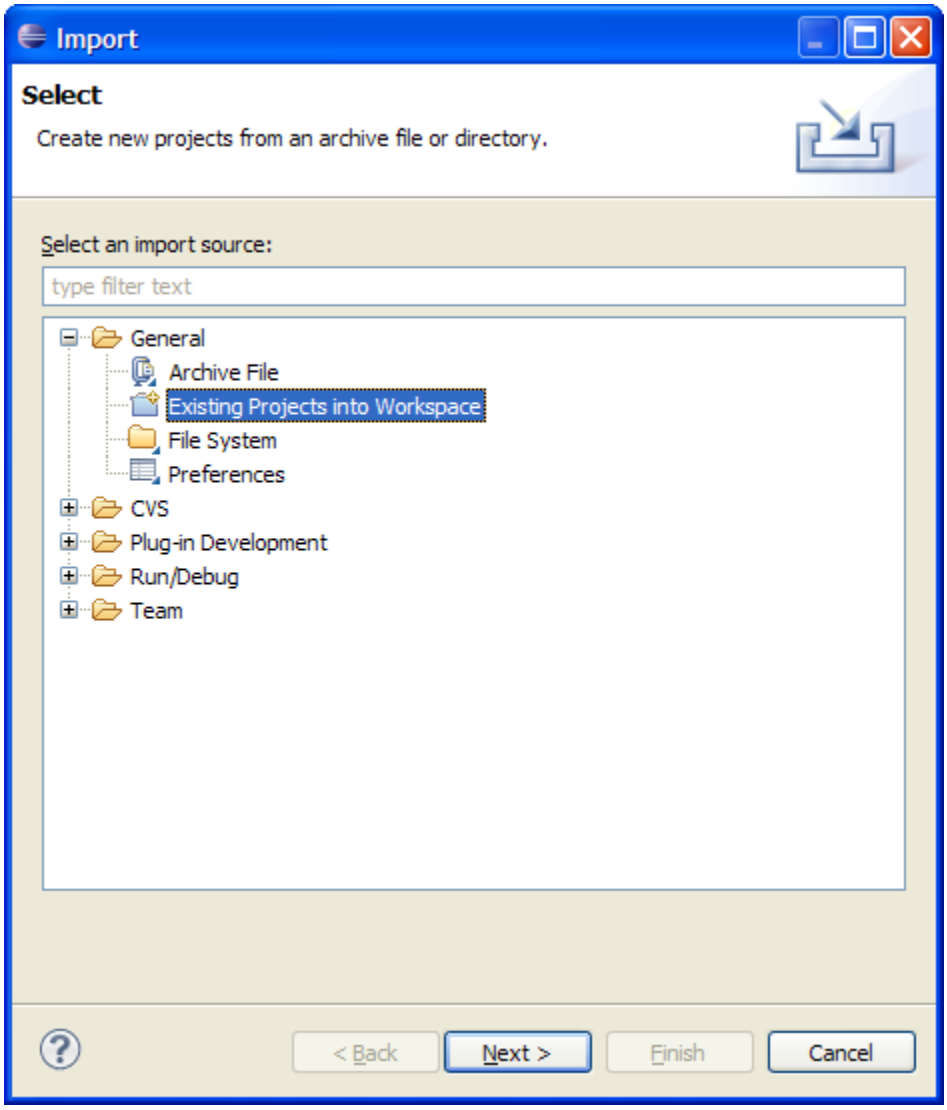
2.1.2 Importing the i2b2 Workbench projects

1. Run eclipse and select a workspace whose *pathname* does not contain spaces.

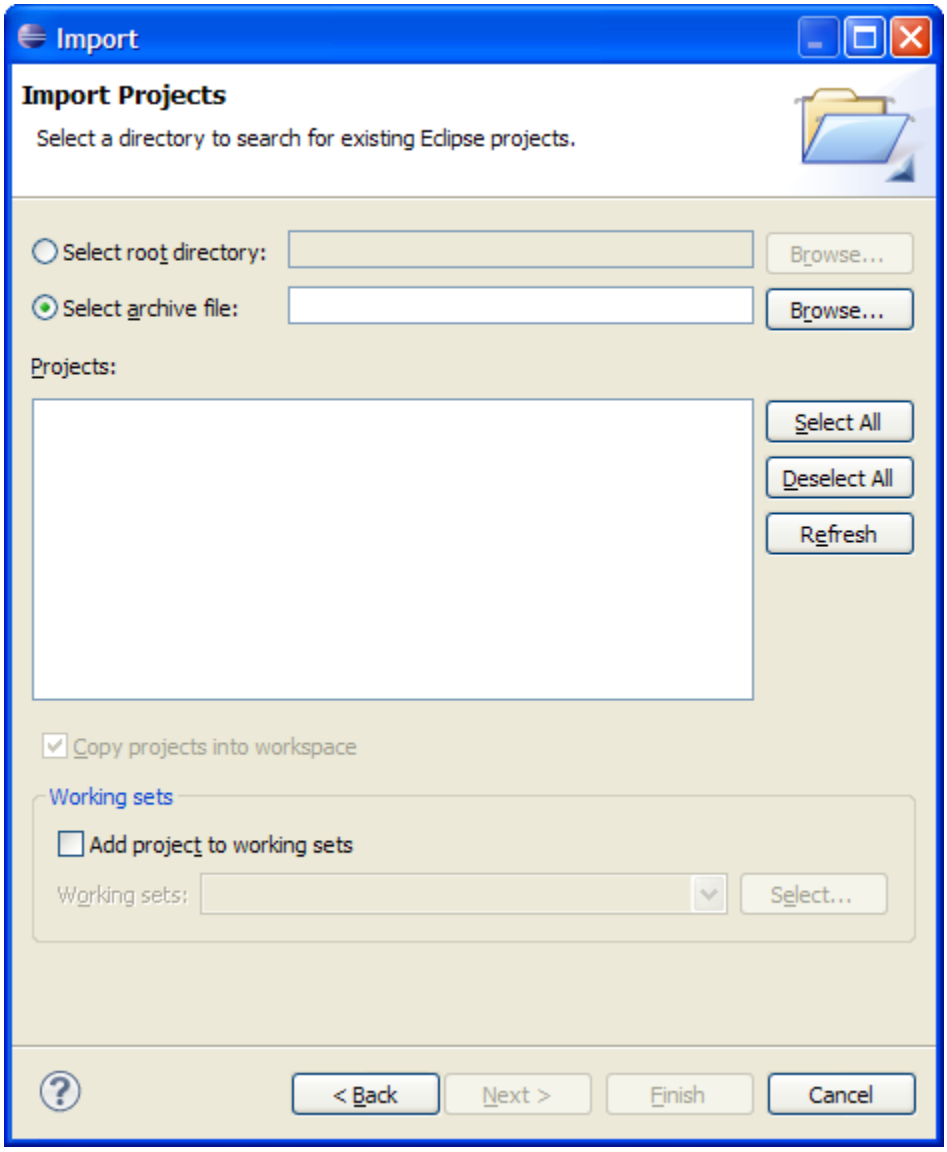
   Example (correct): c:\workspace\i2b2workbench

   Example (incorrect): c:\eclipse workspace\i2b2workbench

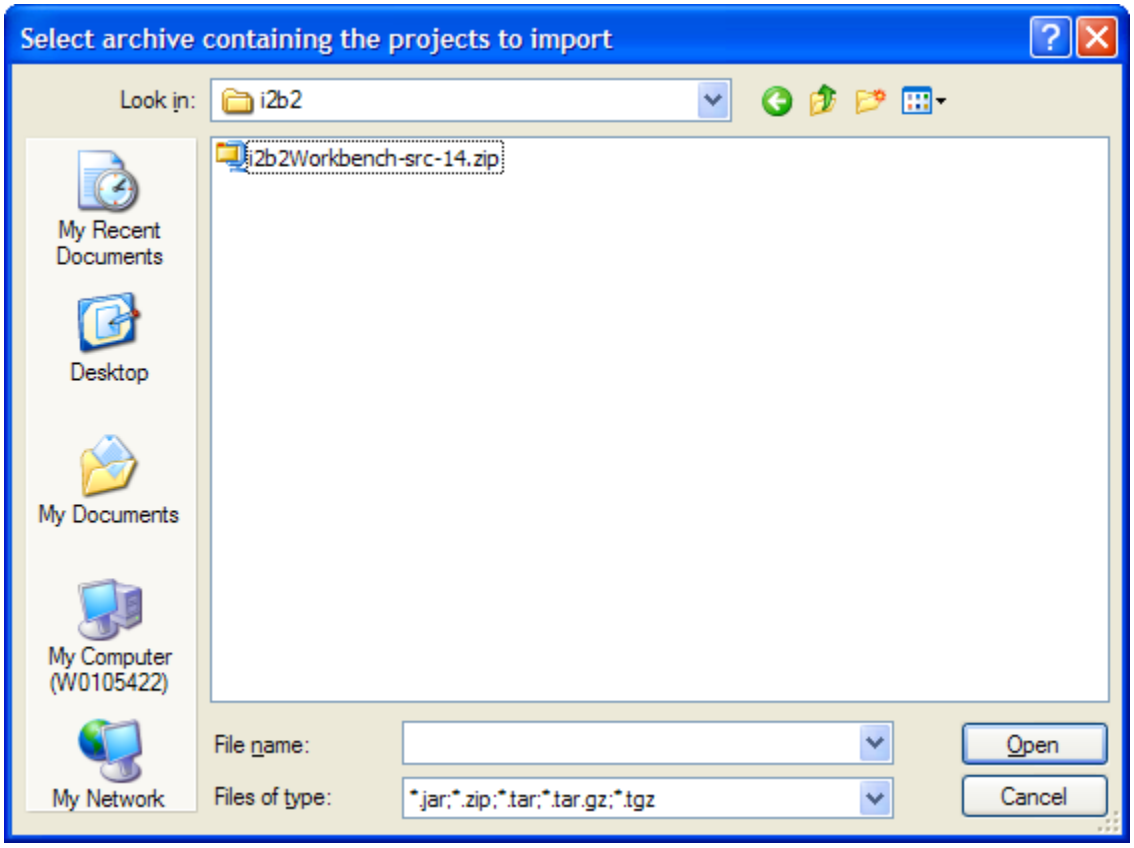
2. Select **File** from the toolbar in Eclipse and then select **Import** from the drop down menu.
3. The *Import dialog box* will open.
4. Double click on the folder called **General** and then select **Existing Projects into Workspace**.



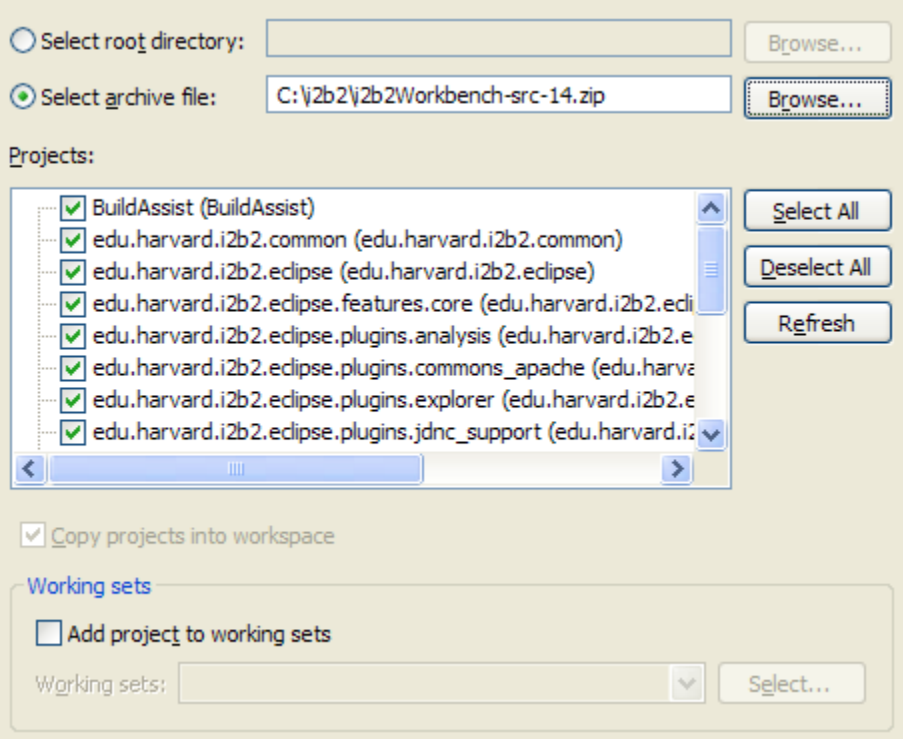
5. Click on the **Next** button.
6. The *Import Projects page* of the import wizard will open.
7. Click on the radial button next to **Select archive file**.



8. Enter the path of the archive file or browse for the location where you downloaded the zip file (***i2b2Workbench-src-14.zip***).



9. Once you open the zip file, the projects will appear in the *Import dialog box*.



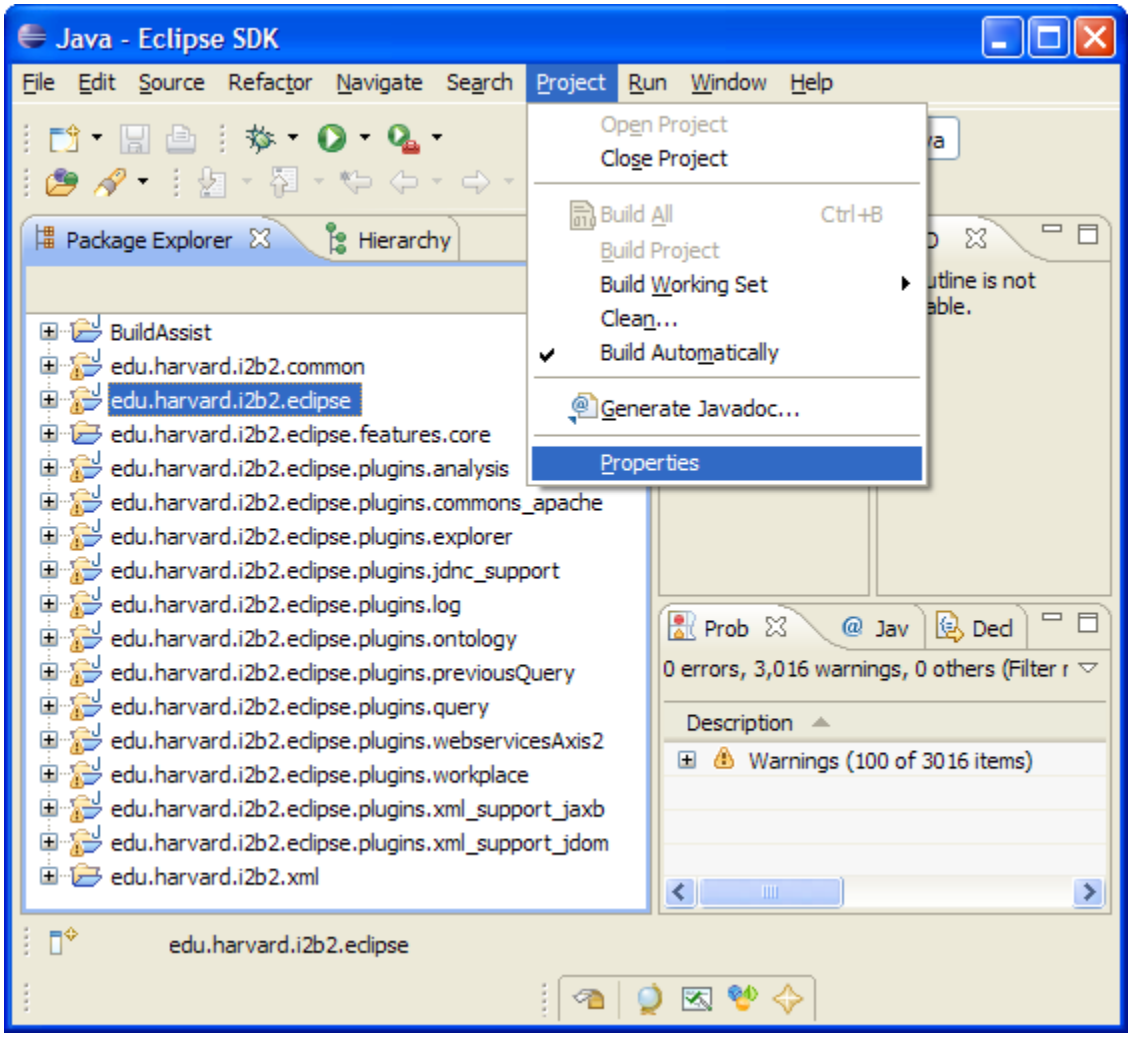
10. Verify all the projects are checked off
11. Click on the **Finish** button.
12. Once the projects have finished importing they will appear in your Eclipse Workbench.

2.2 Building the i2b2 Workbench within the Eclipse IDE

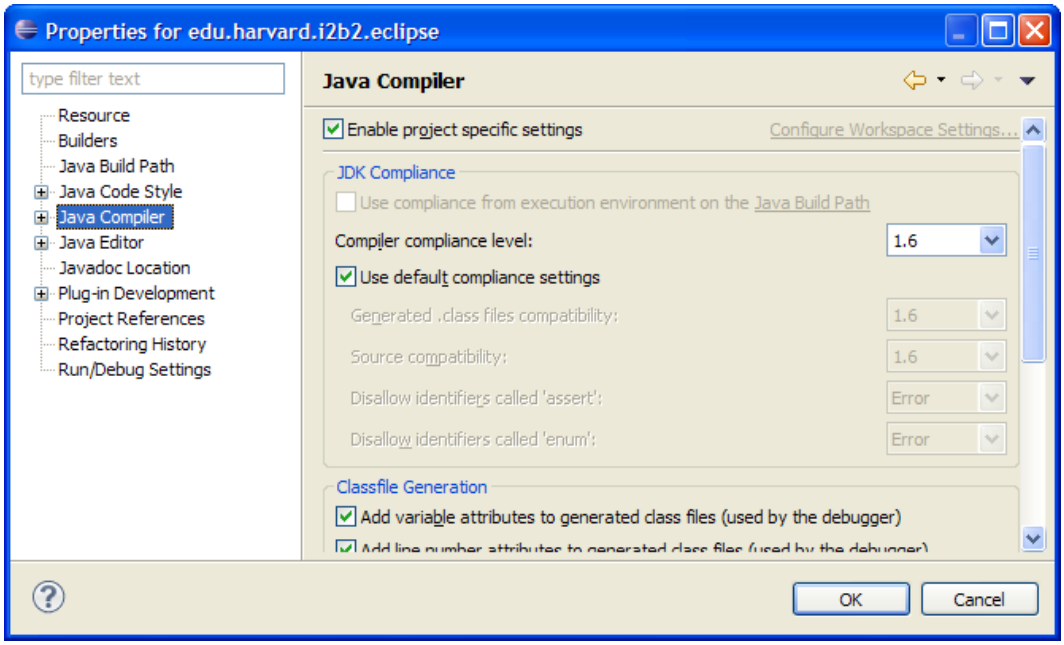
2.2.1 Java Compiler Settings

Check your Java Compiler settings by clicking on **edu.harvard.i2b2.eclipse** and then by clicking on *Projects* located on the Eclipse toolbar. Select Properties from the drop down menu.

1. In the Eclipse workbench, click on the **edu.harvard.i2b2.eclipse** project.
2. On the Eclipse menu bar, click on **Project** and then select **Properties** from the drop down menu.



3. In the left navigation bar of the Properties window click on Java Compiler.

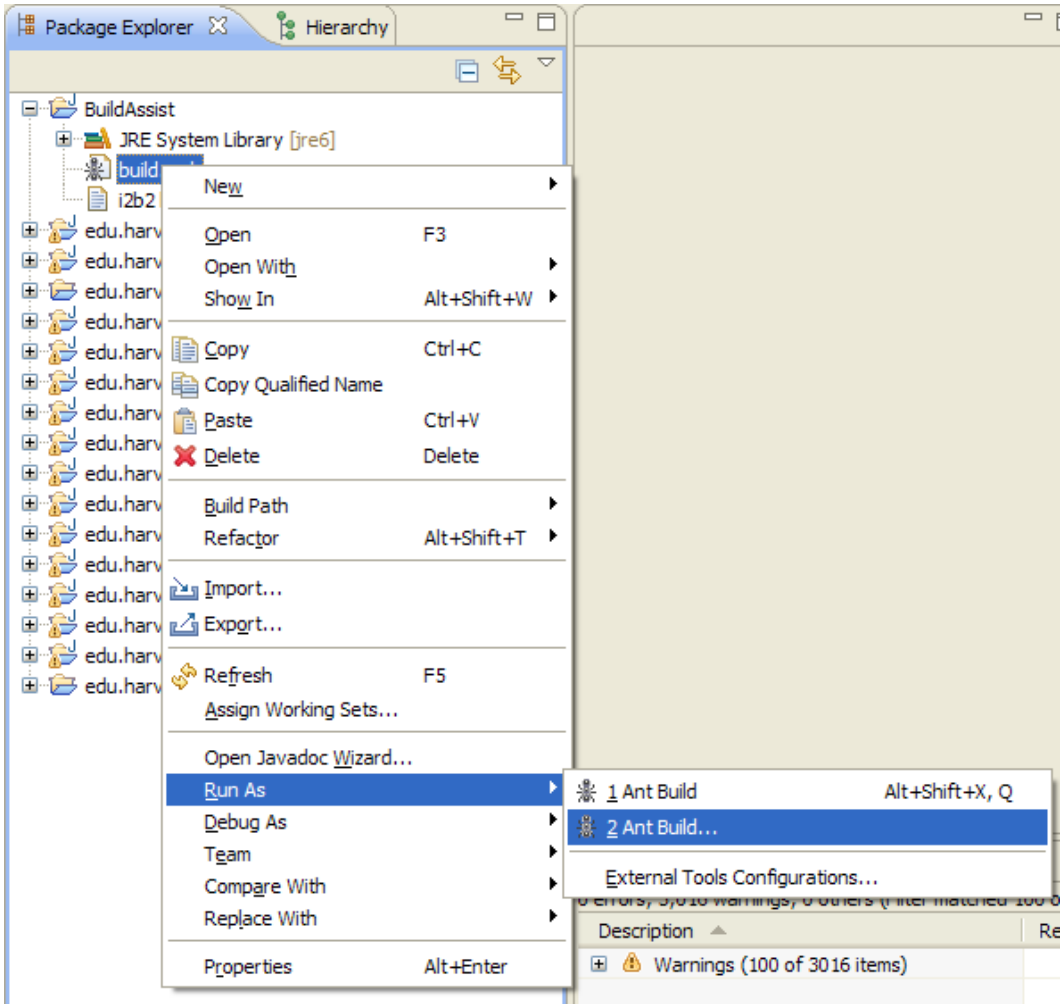


4. Verify the Compiler compliance level is set to 1.6.

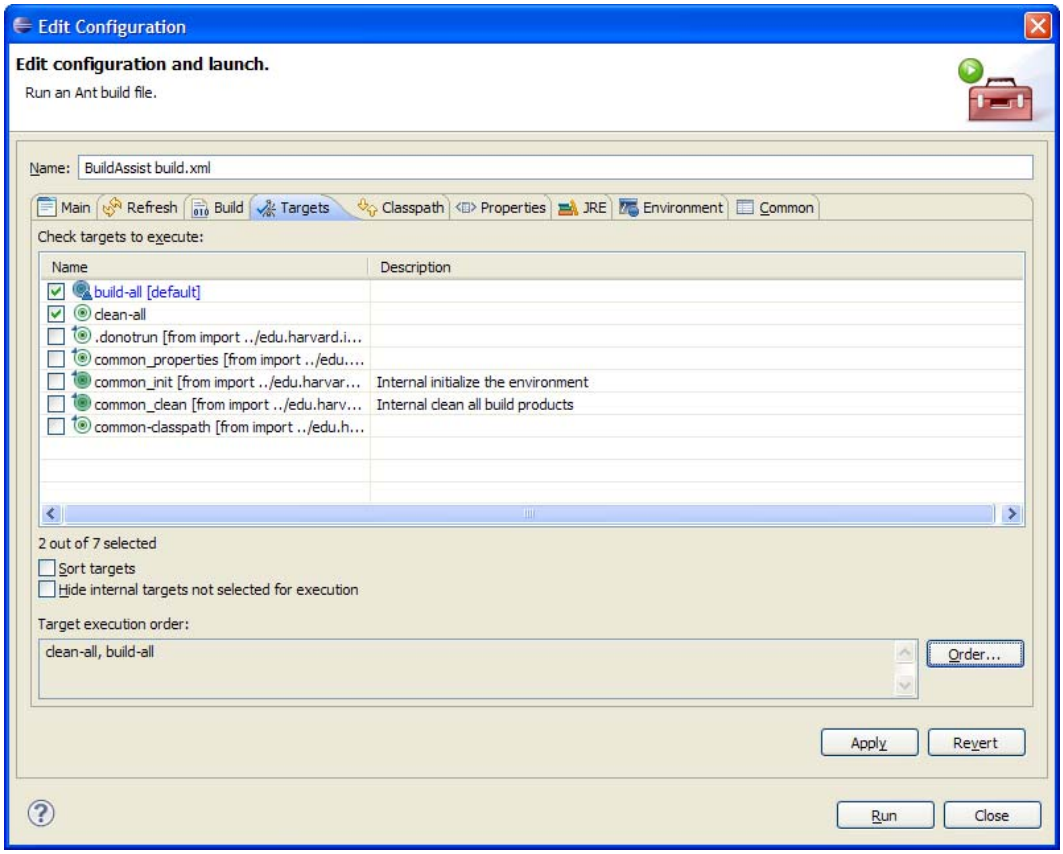
2.2.2 Cleaning and Building the Projects

The following steps outline the process of cleaning and building the projects.

1. Double click on the **BuildAssist** project to open the folder.
2. Right click on the **build.xml** file and select **Run As** from the menu.
3. On the sub-menu there are two **Ant-Build** options, select the second occurrence.



4. The build *Configuration* dialog box will open. Verify the following settings:
    a. The **clean-all** and **build-all** boxes are checked.
    b. The ***target execution*** order is: “clean-all, build-all”



5. Click on the **Run** button to execute.
6. The last step in the build process is to **refresh** all the projects.
    a. Select all the projects.
    b. Right click on the projects and select Refresh from the menu.

Eclipse will automatically build your workbench once the refresh has completed.

2.3 Running the i2b2 Workbench within the Eclipse IDE

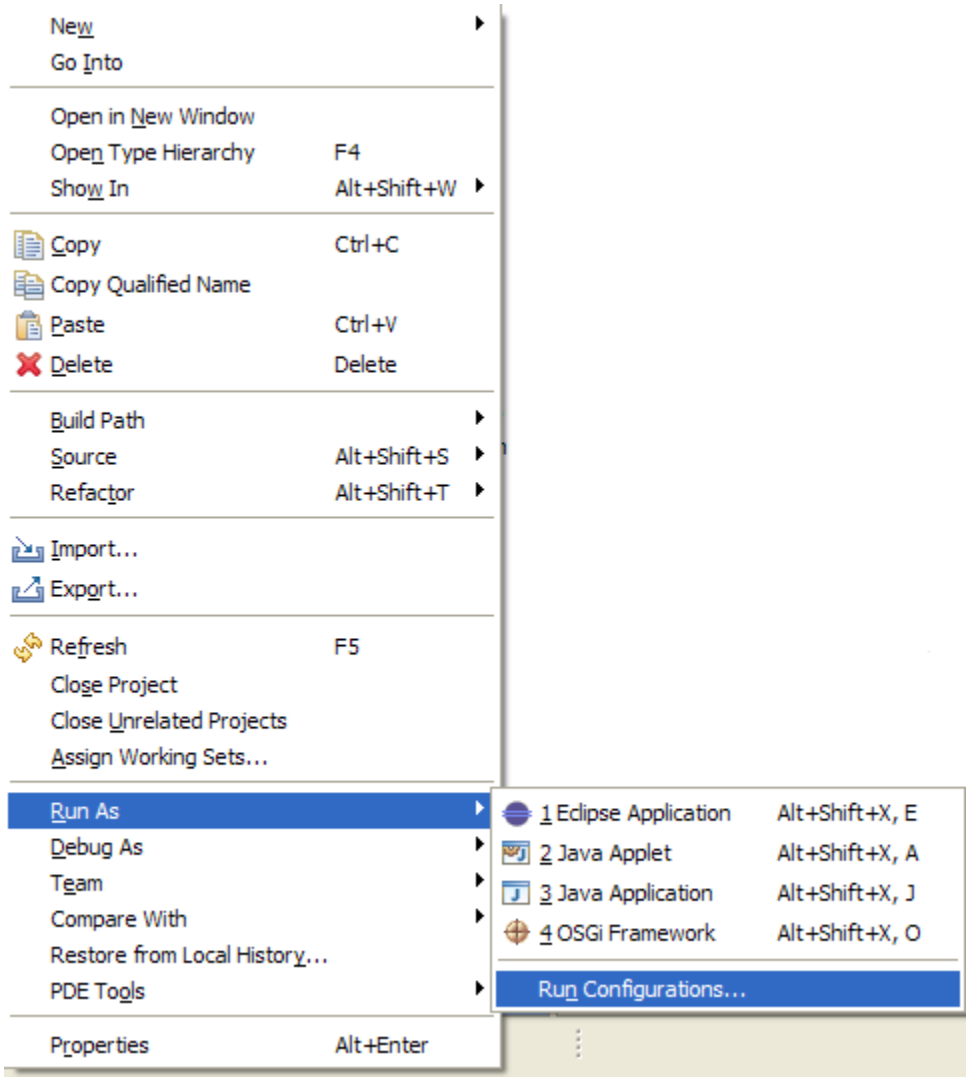
2.3.1 Define Run Configurations

In order to run the i2b2 Workbench from within Eclipse you need to define the run configurations.

- ***You only need to set up the run configuration once.***

Steps to access the Run Configurations window:

1. Right click the **edu.harvard.i2b2.eclipse** project.
2. Select **Run As** from the pop-up menu and **Run Configurations** from the sub-menu.



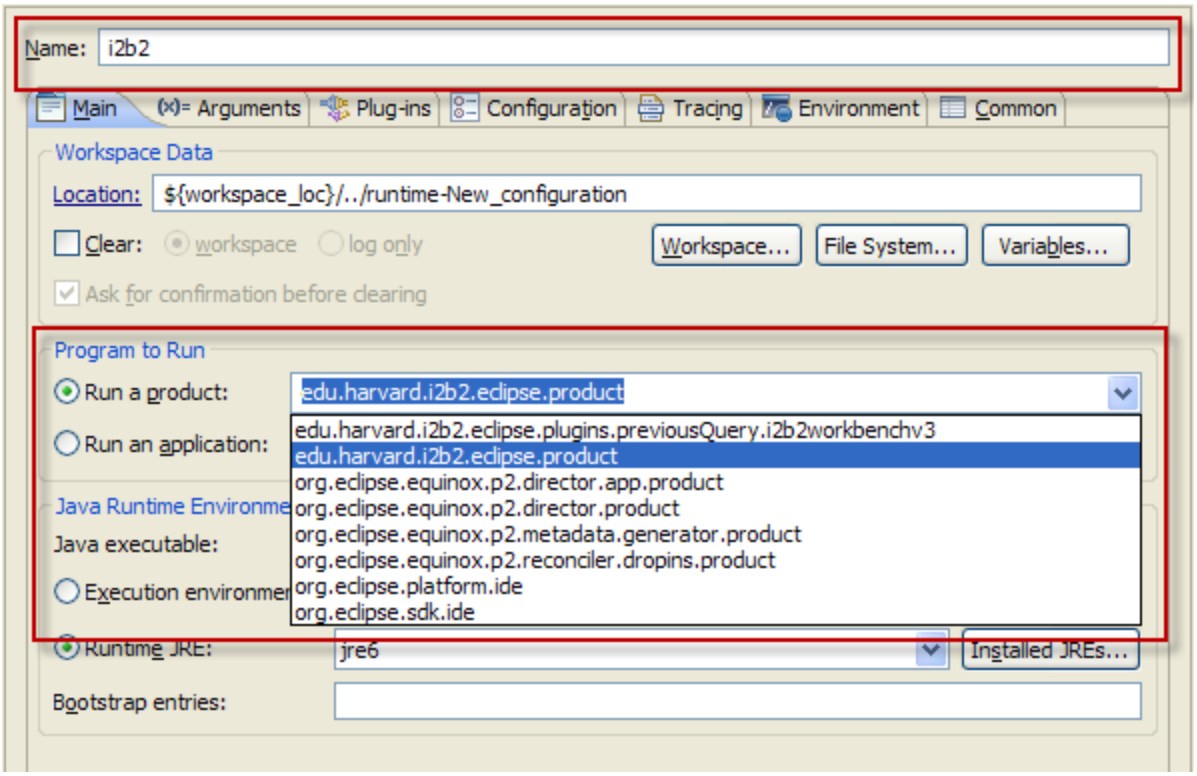
The next three sections outline the process of setting up the configuration information on the *Main*, *Arguments*, and *Plug-ins* tabs.

2.3.1.1 MAIN TAB

1. Double click on "**Eclipse Application**" to create a new configuration.
2. Enter a meaningful name at the name field

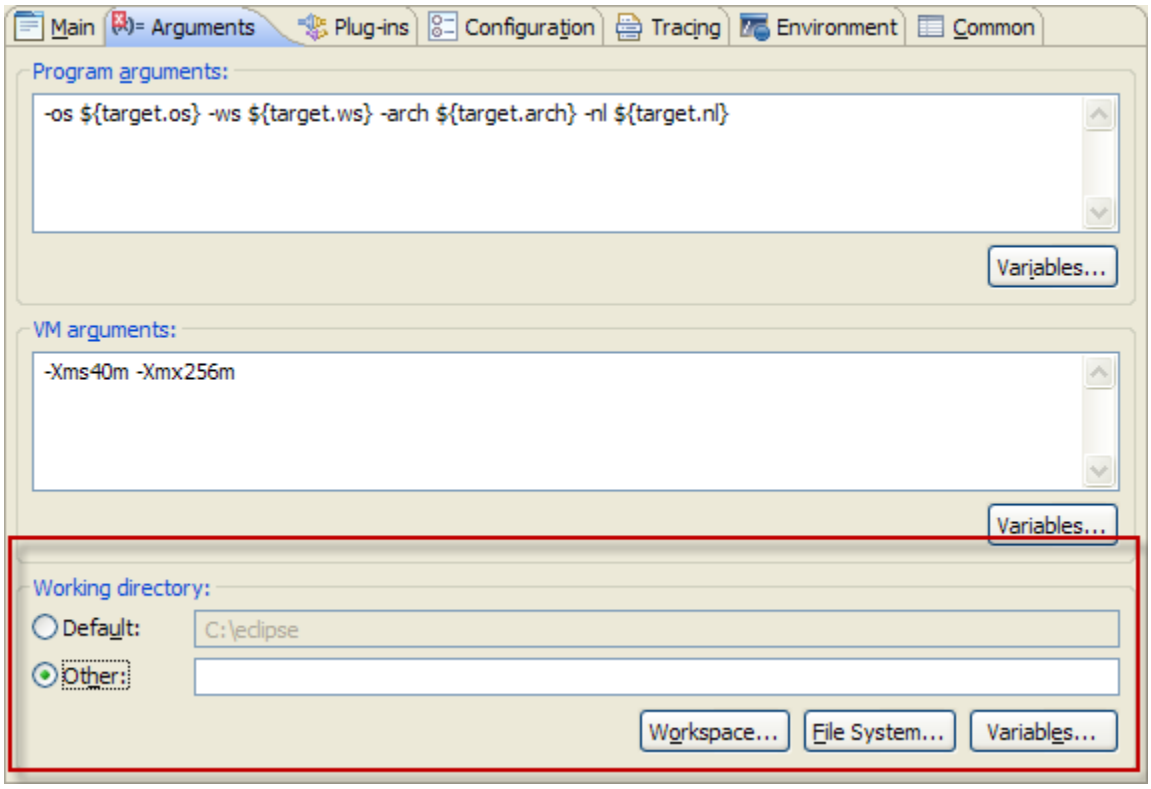
   Example: "i2b2"

3. Verify the dial next to "**Run a product:**" is selected.
4. Click on the arrow to display the list of products and select **edu.harvard.i2b2.eclipse.product** from the drop down list.

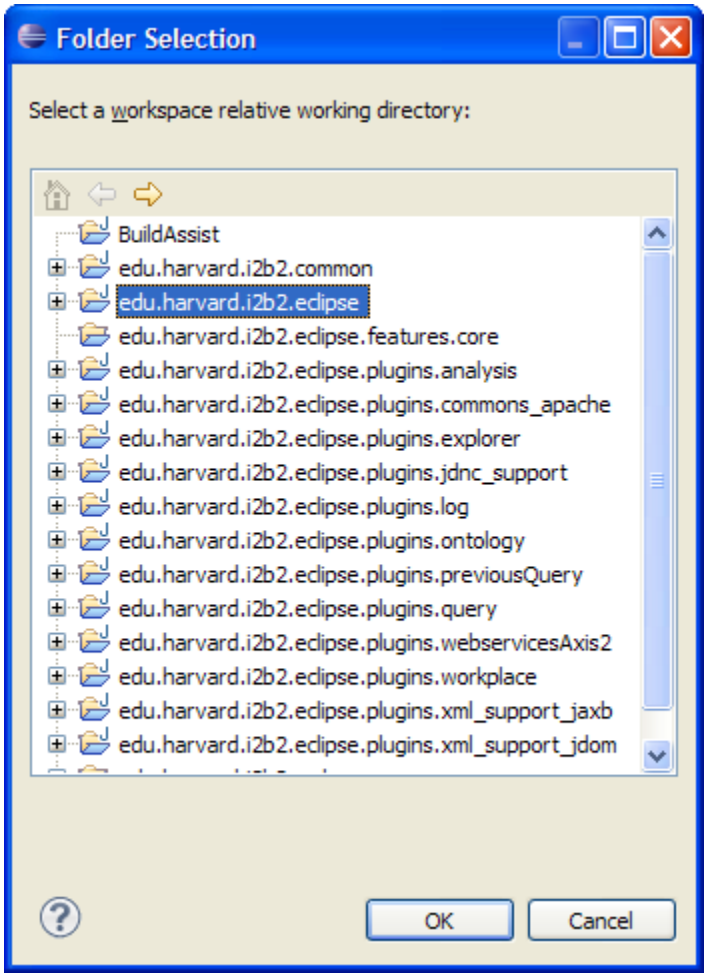


2.3.1.2 ARGUMENTS TAB

1. Click on the “**Arguments**” tab.
2. In the “**Working directory:**” section click on “**Other**”

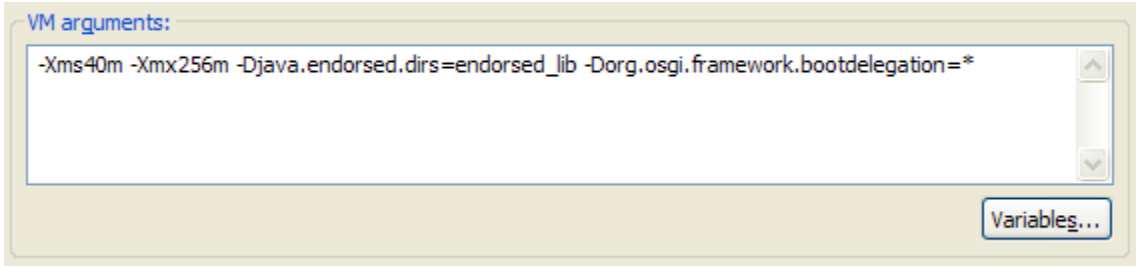


3. Click on the “**Workspace**” button.
4. The *Folder Selection* window will open.
5. Select **edu.harvard.i2b2.ecplise** and click on the “**OK**” button.



6. In the "**VM arguments:**" section add the following:

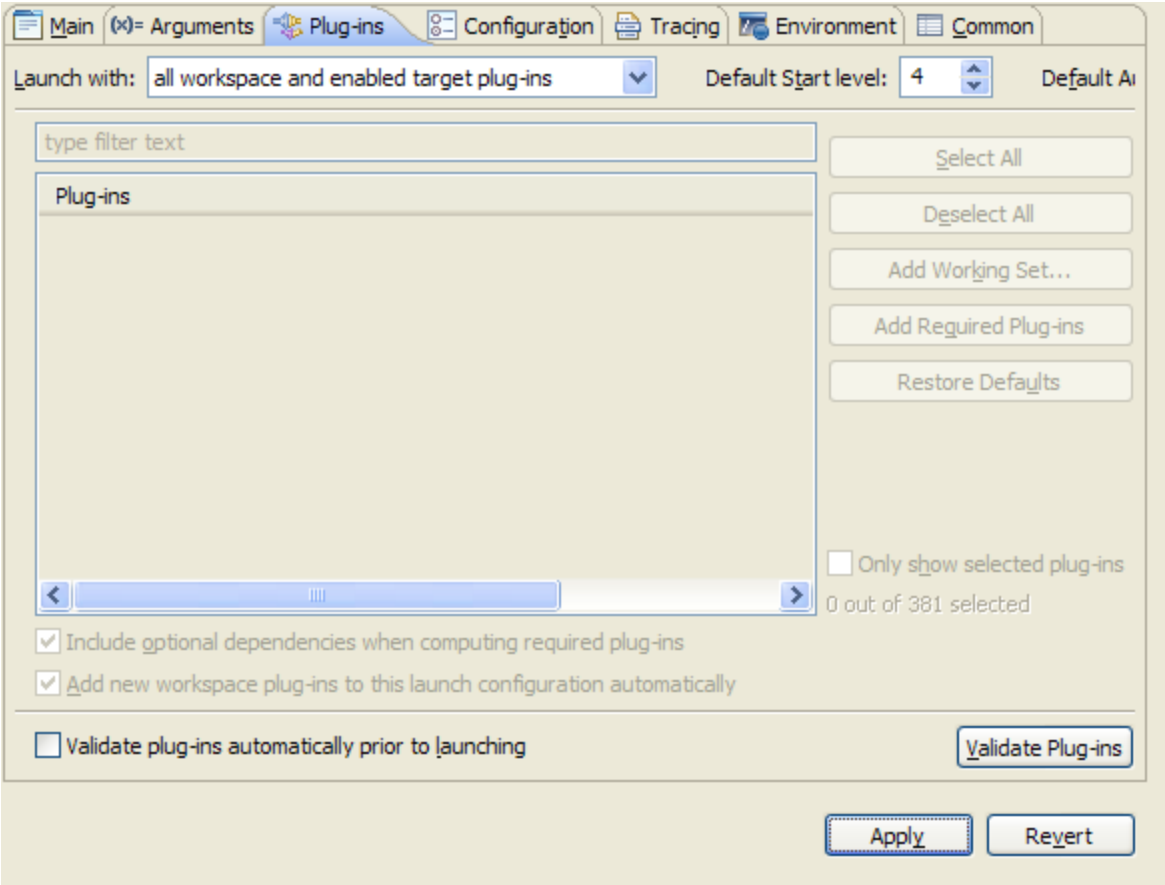
```
-Djava.endorsed.dirs=endorsed_lib
-Dorg.osgi.framework.bootdelegation=*
```



- ***-Xms40m and –Xmx256m should already default. If not then these also need to be added.***

2.3.1.3 PLUG-INS TAB

You can configure any combination of workspace plug-ins and target platform plug-ins as your IDE configuration. The default is that all of the workspace and target platform plug-ins are selected.



At this time we recommend that you click on the "Apply" button to save your settings.

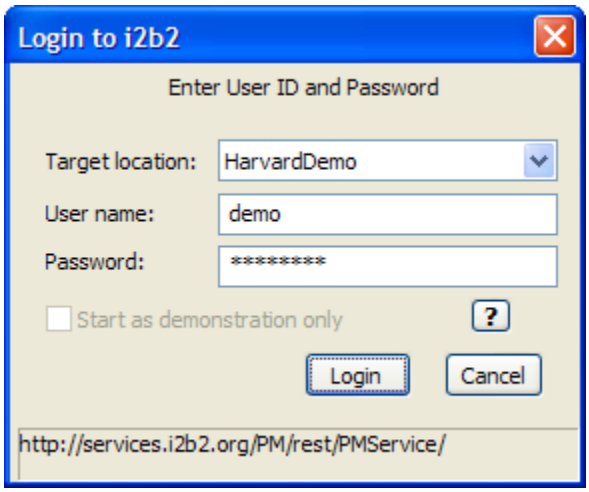
2.3.2 Running the i2b2 Workbench Application

The following steps outline the process of running and logging into the i2b2 Workbench.

The below steps assume you are starting from within the* Run Configurations *window. Please see the above section if you need to know how to access this window.

1. Click on the "**Run**" button to run the i2b2 Workbench under control of the Eclipse IDE.
2. The i2b2 Workbench splash screen will appear.

3. The **Login** dialog box will open.



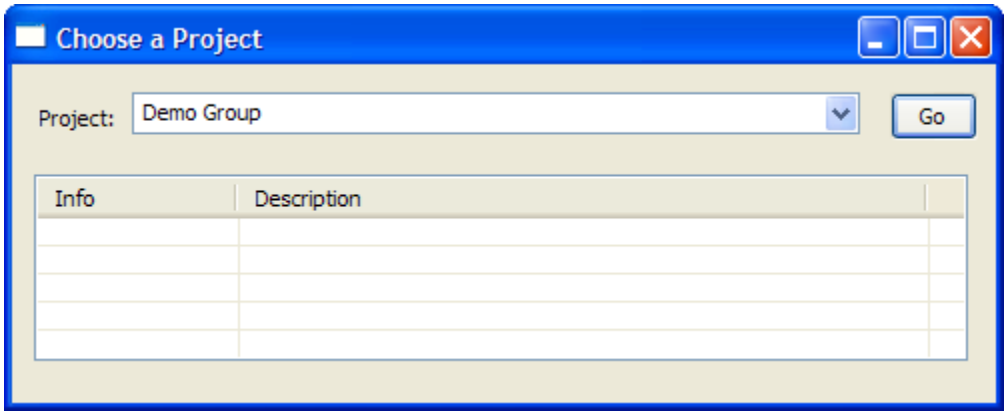
4. Enter the following information:

   User name: demo

   Password: demouser

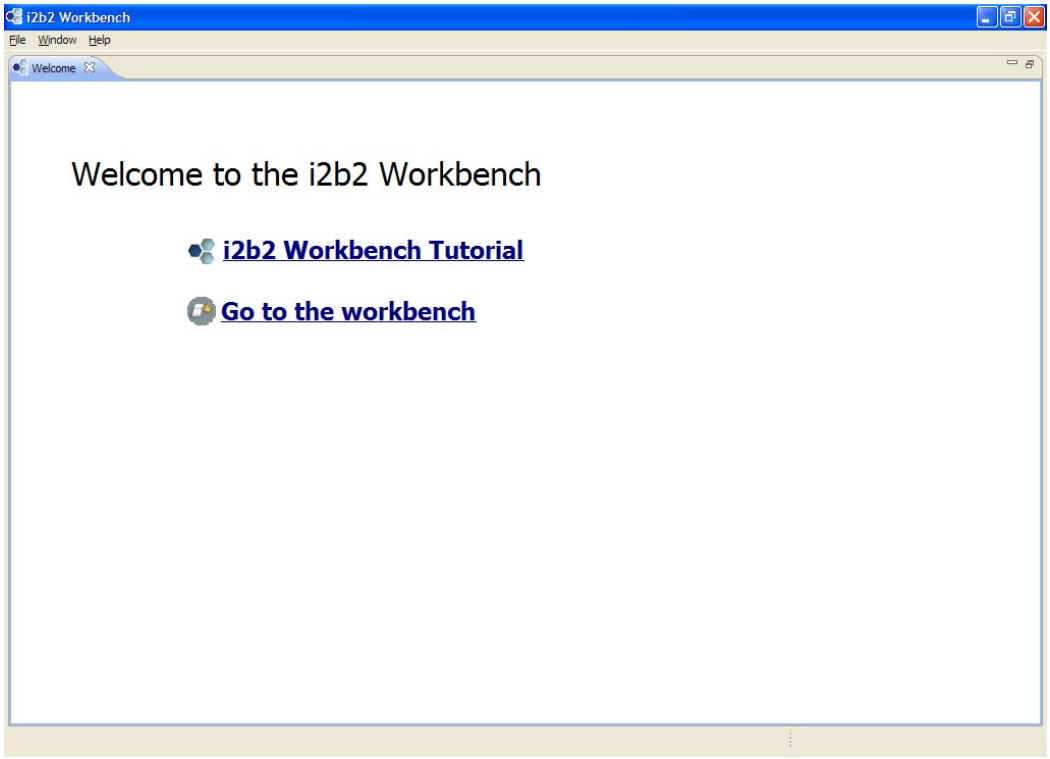
5. Click on the "**Login**" button.
6. The **Choose a Project** dialog box will open.

The project dialog box will only appear if you have access to multiple projects. If you only have access to one project you will be brought to the Welcome page.

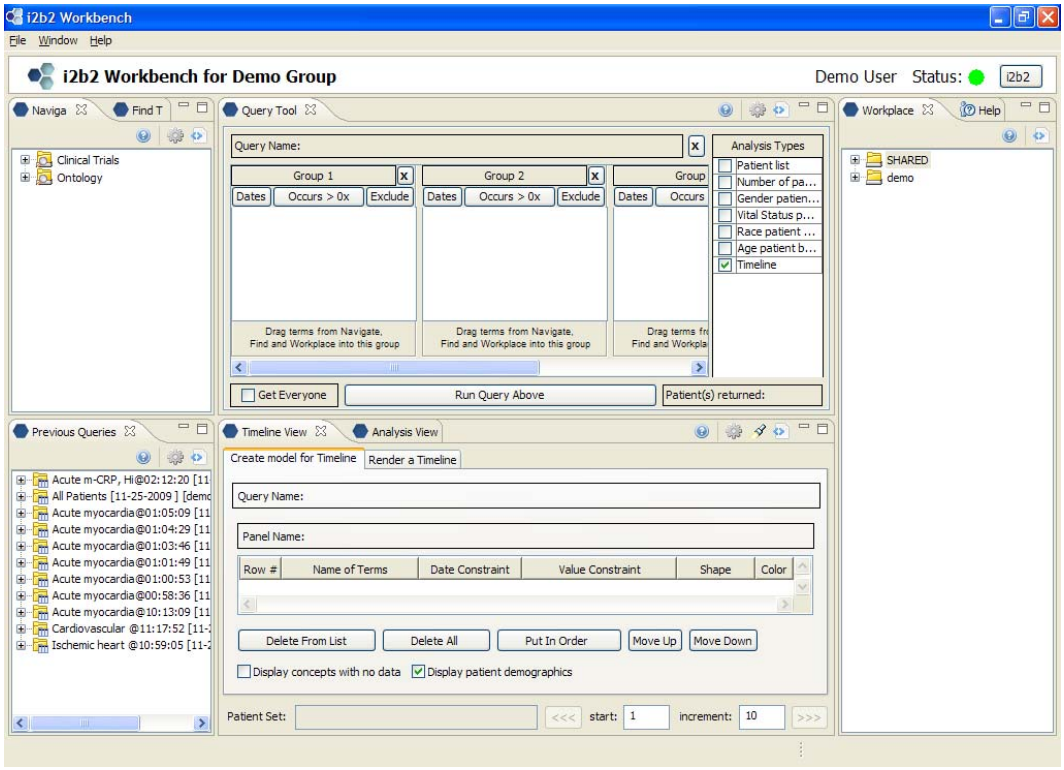


7. Accept the default project by simply clicking on the "**Go**" button.
8. The **Welcome** page will open.

The welcome page will open the first time you login. All subsequent logins will bring you directly to the i2b2 desktop. You can access the welcome page anytime by selecting it from the Help menu on the toolbar.



9. Click on **Go to the workbench**.
10. The **desktop** of the i2b2 Workbench will open.



2.3.2.1 RECONFIGURING THE DEFAULT LAYOUT

The default layout can be reconfigured to the users needs.

a. On the toolbar, click on **Window** and select "**Show View**" from the drop down menu.
b. Click on the views you would like to add.
c. Move the position of any open view by clicking on its tab and dragging it to another position.

3. PACKAGE STRUCTURE

The ***archive file*** contains the following projects. The project/directory **edu.harvard.2b2.eclipse** is considered the *base plug-in* for the project and contains project properties files and configuration files.

Directory	Description
BuildAssist	Files to build the workbench
edu.harvard.i2b2.common	Common core exception, jaxb, and utilities used by i2b2 hive
edu.harvard.i2b2.eclipse	The base plug-in for our development (namely the UI for the top panel)
edu.harvard.i2b2.eclipse.features.core	The base features plug-in for the i2b2 core views.
edu.harvard.i2b2.eclipse.plugins.analysis	UI for the Analysis view.
edu.harvard.i2b2.eclipse.plugins.commons_apache	Third party jars needed for common functions
edu.harvard.i2b2.eclipse.plugins.explorer	UI for the Timeline view
edu.harvard.i2b2.eclipse.plugins.jdnc_support	Third party jars needed for UI support
edu.harvard.i2b2.eclipse.plugins.log	Third party jars needed for logging
edu.harvard.i2b2.eclipse.plugins.ontology	UI for the Ontology view
edu.harvard.i2b2.eclipse.plugins.previousQuery	UI for the Previous Query view
edu.harvard.i2b2.eclipse.plugins.query	UI for the Query view
edu.harvard.i2b2.eclipse.plugins.webservicesAxis2	Third party jars needed for web service support
edu.harvard.i2b2.eclipse.plugins.workplace	UI for the Workplace view
edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb	Third party jars needed for jaxb/message processing
edu.harvard.i2b2.eclipse.plugins.xml_support_jdom	Third party jars needed for jdom/message processing
edu.harvard.i2b2.xml	XSD for the hive

4. ECLIPSE (I2B2) FEATURE

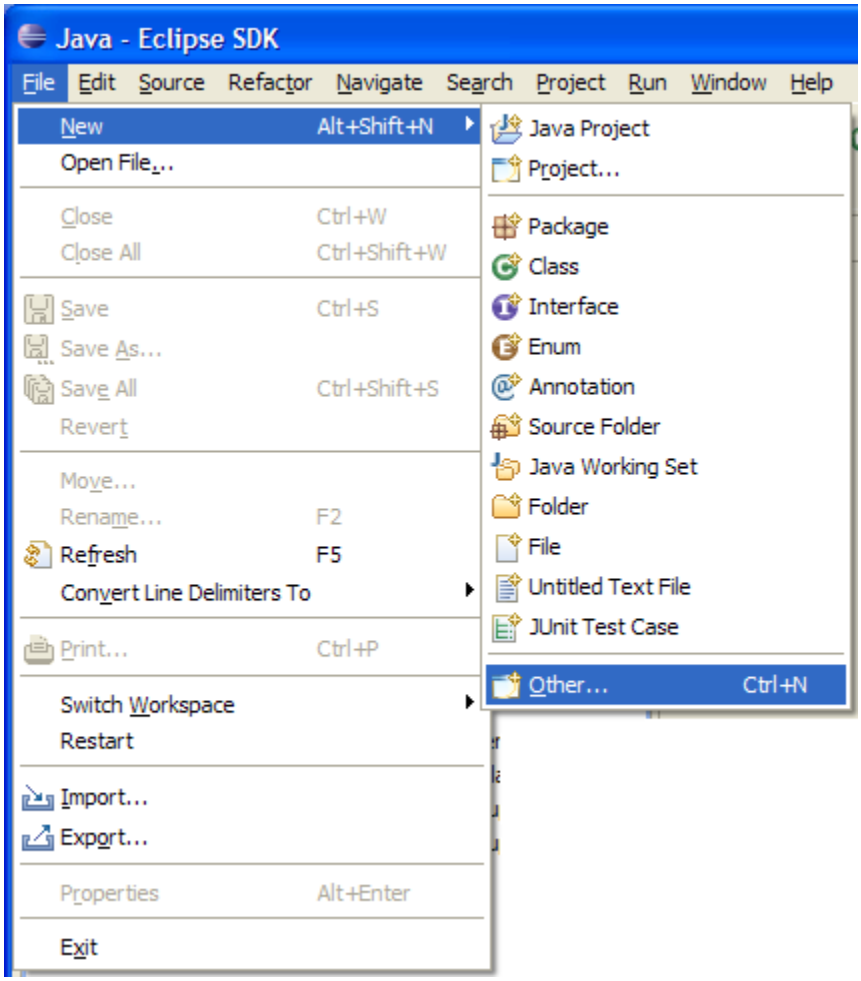
In Eclipse the plug-ins can be packaged together as an Eclipse Feature. The edu.harvard.i2b2.eclipse.features.core plug-in is the baseline Eclipse Feature for the i2b2 Workbench. Included in this feature are the core i2b2 Views and required plug-ins to run the i2b2 Workbench. As optional views are developed they will be packaged up into a feature and added to an update site where users can easily install the new view from within the i2b2 Workbench.

The following sections provide a general overview. For more detailed information about Eclipse Features please go to the Eclipse website.

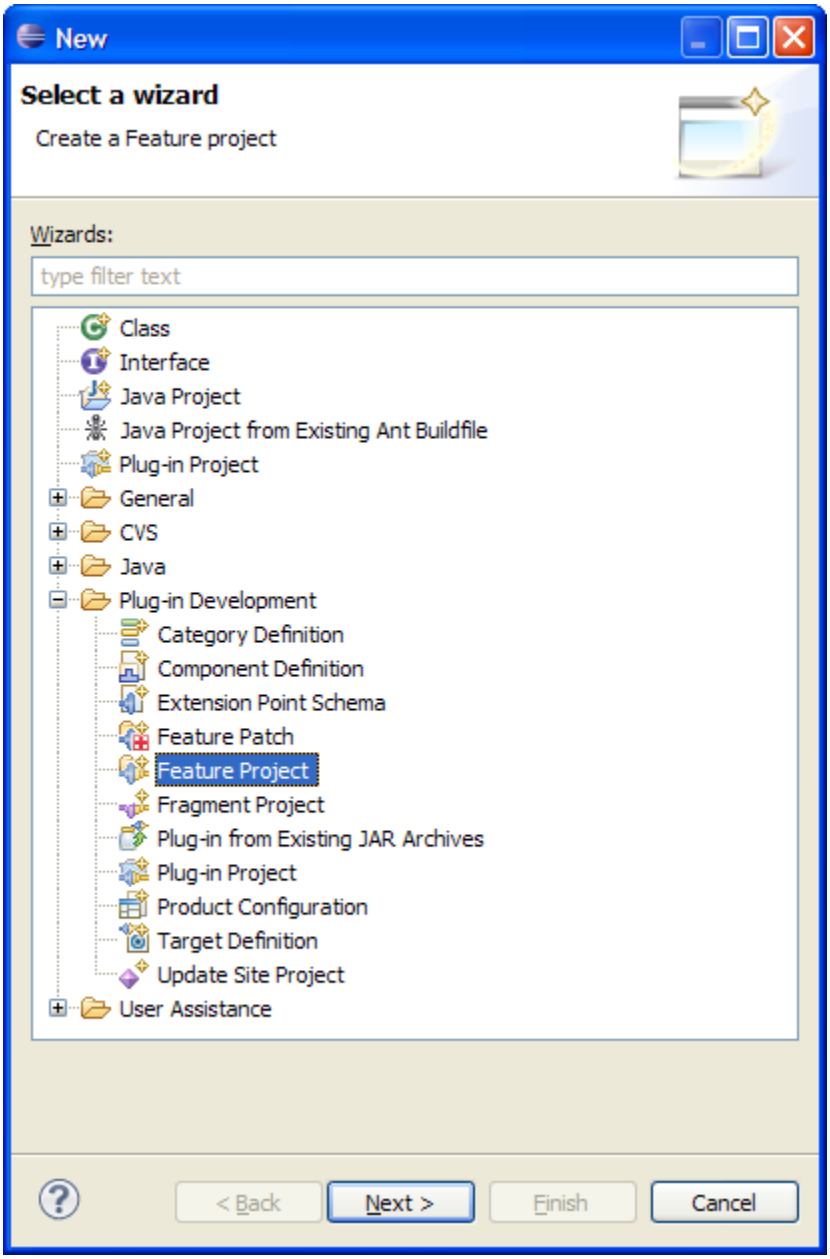
4.1 Creating a New "Feature" Project

The following steps outline how to create a new Feature Project.

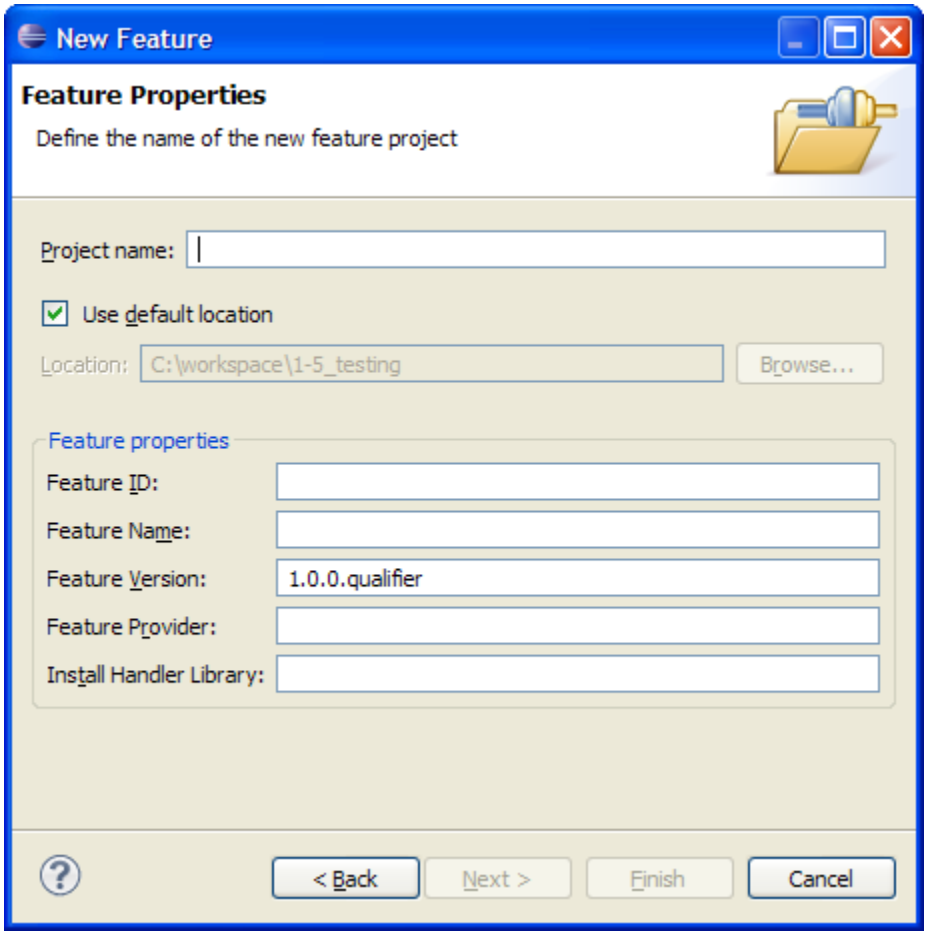
1. On the Eclipse menu bar click on **File**.
2. Select **New** from the drop down menu and **Other** from the sub-menu.



3. The **New** dialog box will open to the **Select Wizard** page.
4. Double click on **Plug-in Development** to open the folder.
5. Click on **Feature Project**.



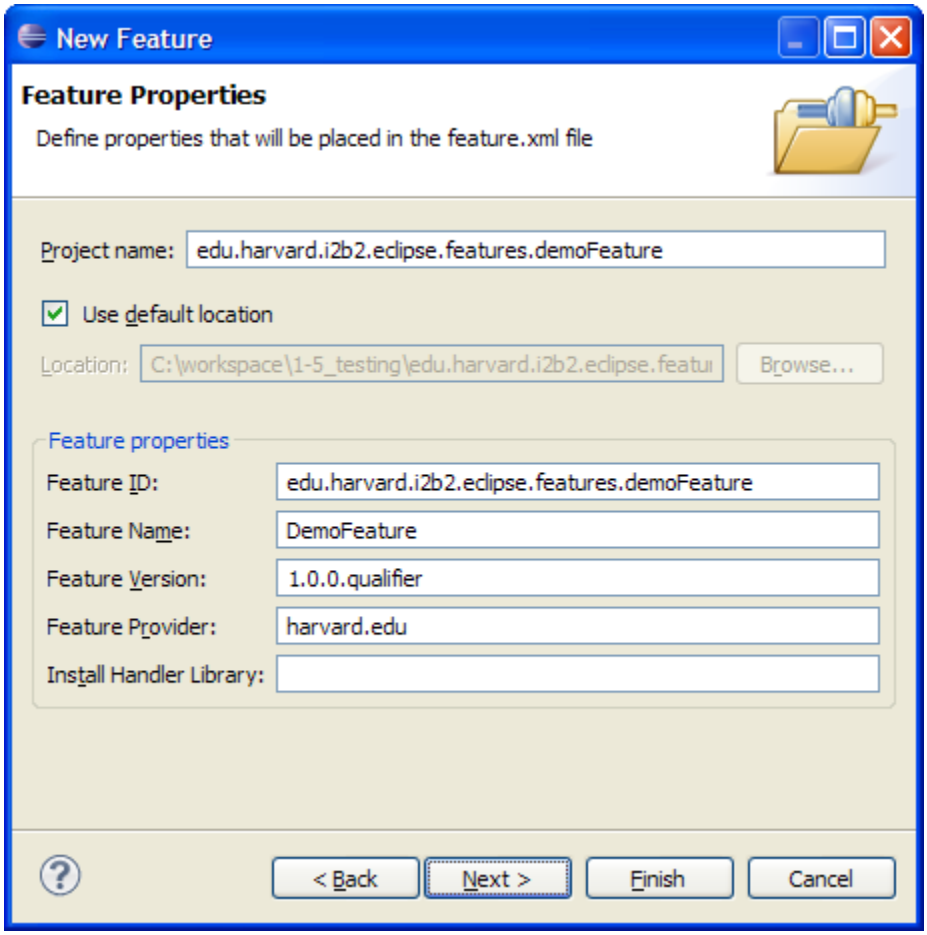
6. Click on the **Next** button.
7. The **New Feature** dialog box will open to the **Feature Properties** page.



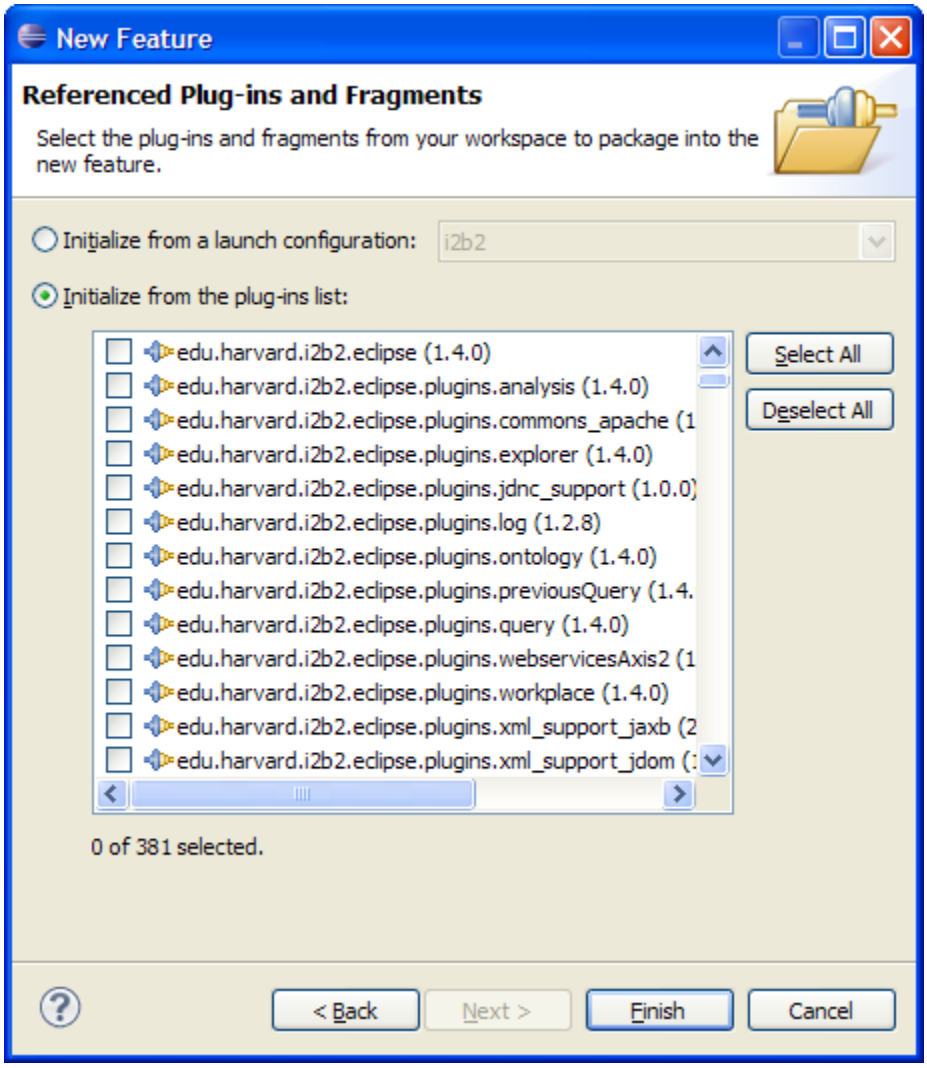
8. At the **Project Name** field enter the name of the new feature project.

As you enter the Project Name the Feature ID and Feature Name fields will automatically be populated.

9. Enter the provider of this feature at the **Feature Provider** field.



10. Click on the **Next** button.
11. The **Referenced Plug-ins and Fragments** page will open.
12. Select the plug-in(s) you want to include in the feature.



13. Click on the **Finish** button.
14. The feature project is created and now appears in your workspace.

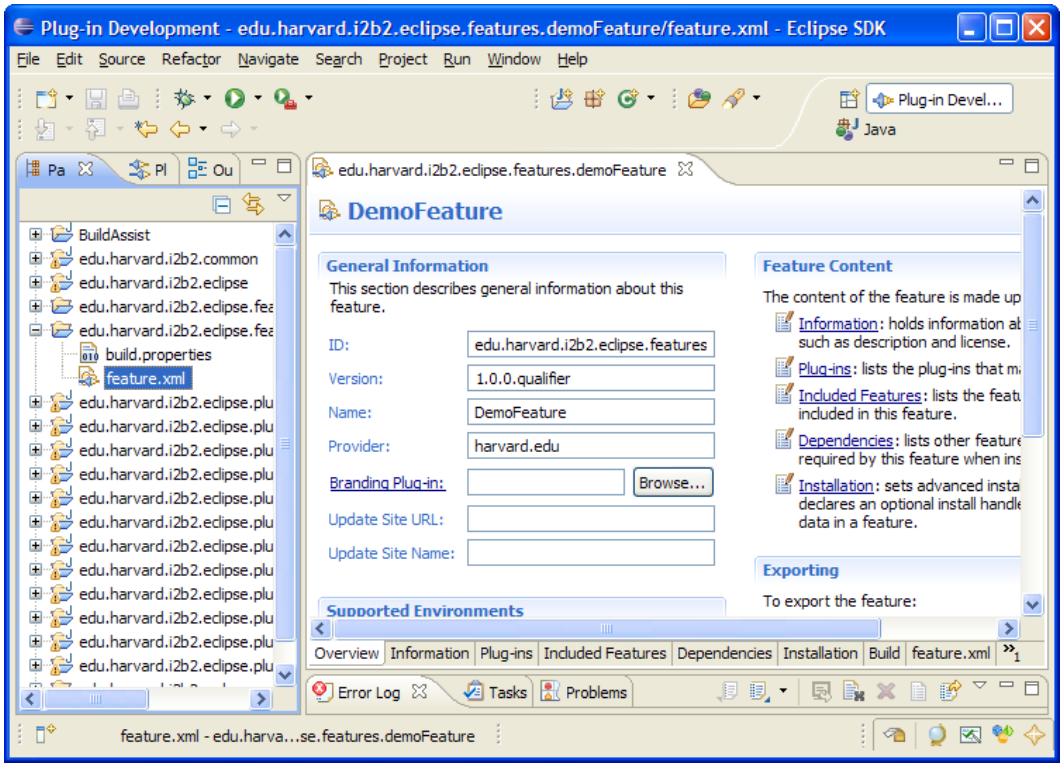
4.2 Feature Manifest (feature.xml)

Eclipse automatically creates a manifest or central file that contains all the information needed for the feature to work. This file is called the feature.xml and contains license and copyright information as well as the required plug-ins, dependencies and builds information.

4.2.1 Opening the Feature Manifest Editor

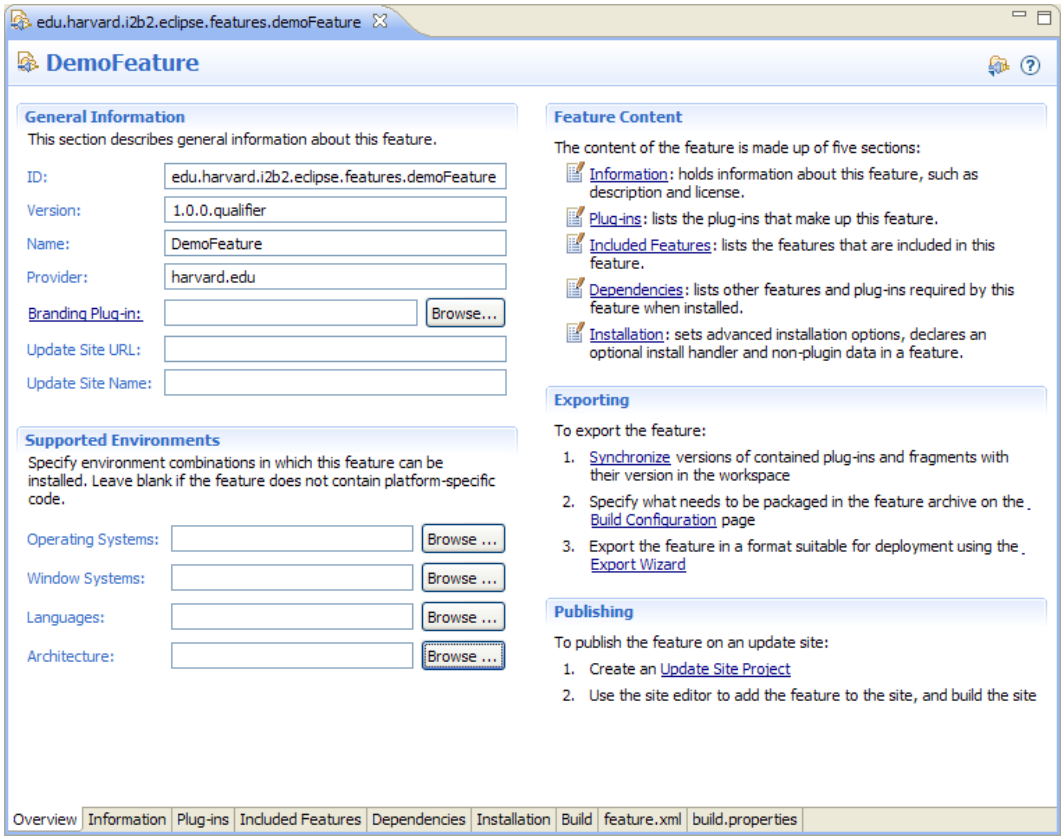
The following outlines how to open the **Feature Manifest Editor**.

1. In the list of projects, double click on the name of the feature you created in the earlier section called '*Creating a New "Feature" Plug-in*'.
2. Double click on the **feature.xml** file.
3. The **Feature Manifest Editor** will open to the **Overview** page.



4.2.2 Overview Tab

The **Overview** page contains basic information about the feature.

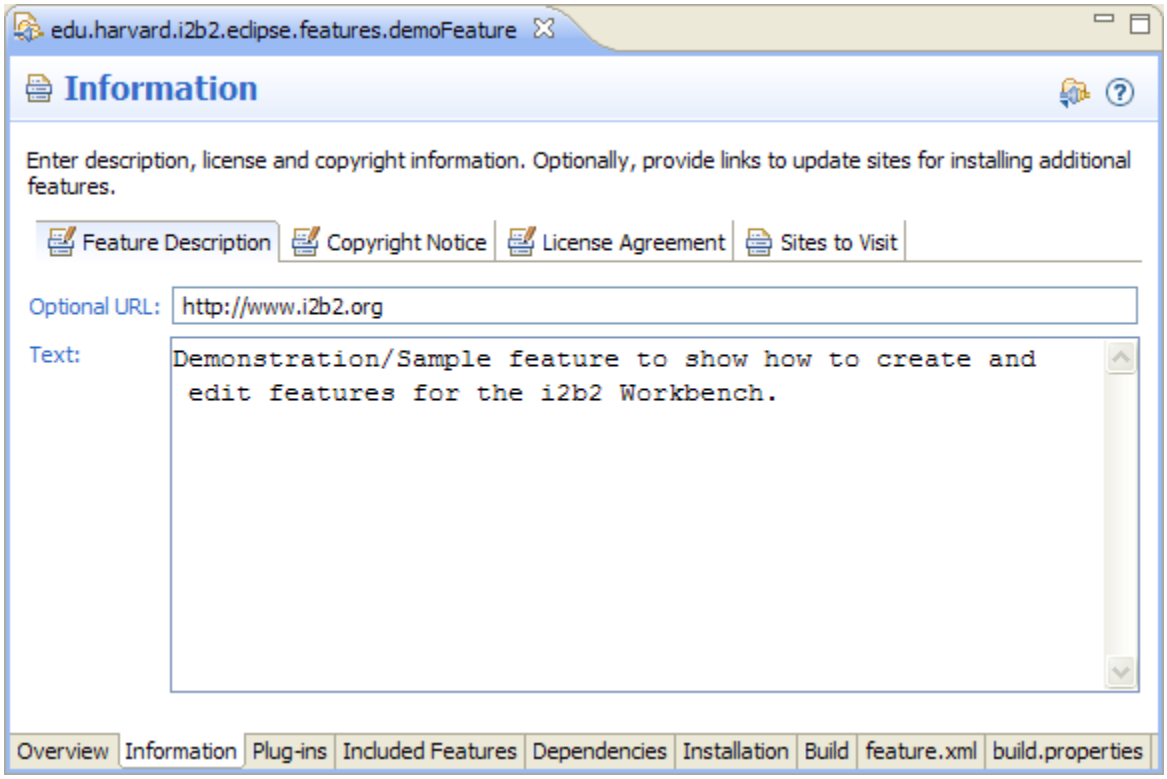


4.2.3 Information Tab

The **Information** page contains several pages for entering a *description*, *copyright notice*, *license agreement* and *sites to visit*. All of this information is visible to the user when they install the feature from within the i2b2 Workbench.

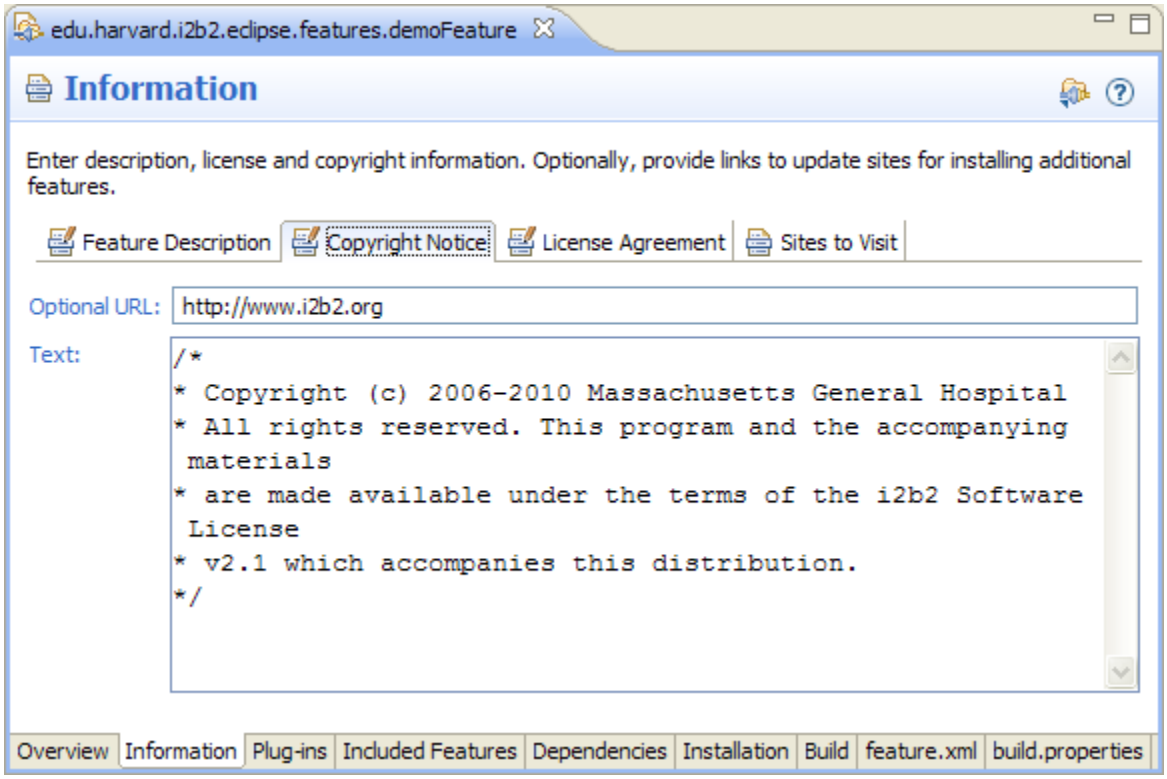
4.2.3.1 FEATURE DESCRIPTION

The first page to appear when selecting the *Information* page is the **Feature Description**. This page allows you to enter a brief description that users will see when installing this feature. You can also enter a URL address of where the user can go to see additional information.



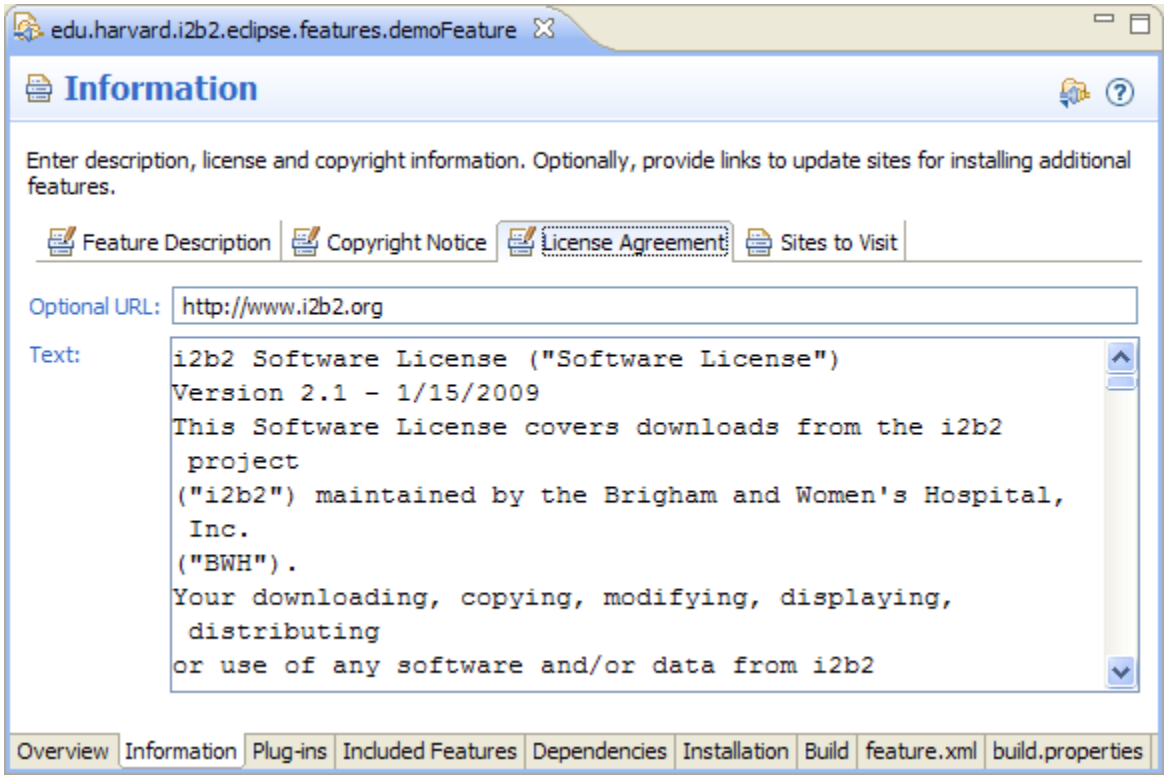
4.2.3.2 COPYRIGHT NOTICE

Clicking on the tab for the **Copyright Notice** page will allow you to enter any copyright information that is required for this feature.



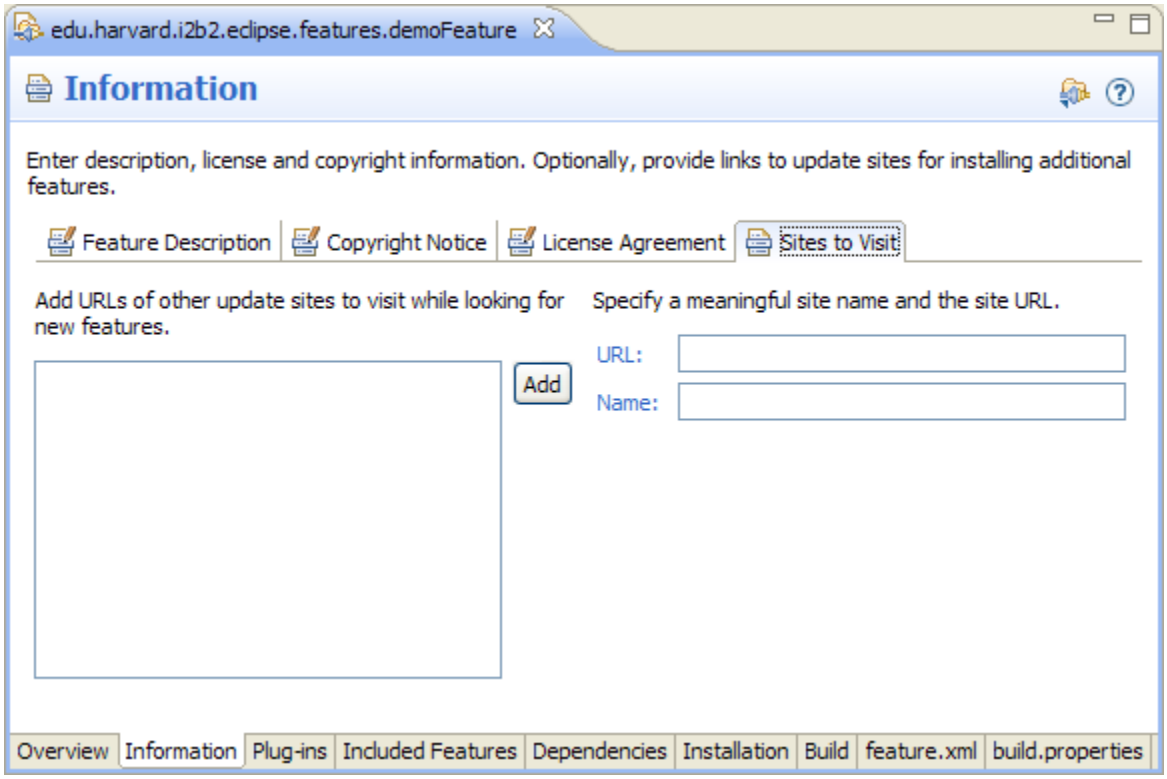
4.2.3.3 LICENSE AGREEMENT

The information you enter on the **License Agreement** page will appear during the installation of the feature in the i2b2 Workbench. Users will have to accept the license in order to continue with the installation process. If they do not want to accept the license then they will not be able to install the feature.



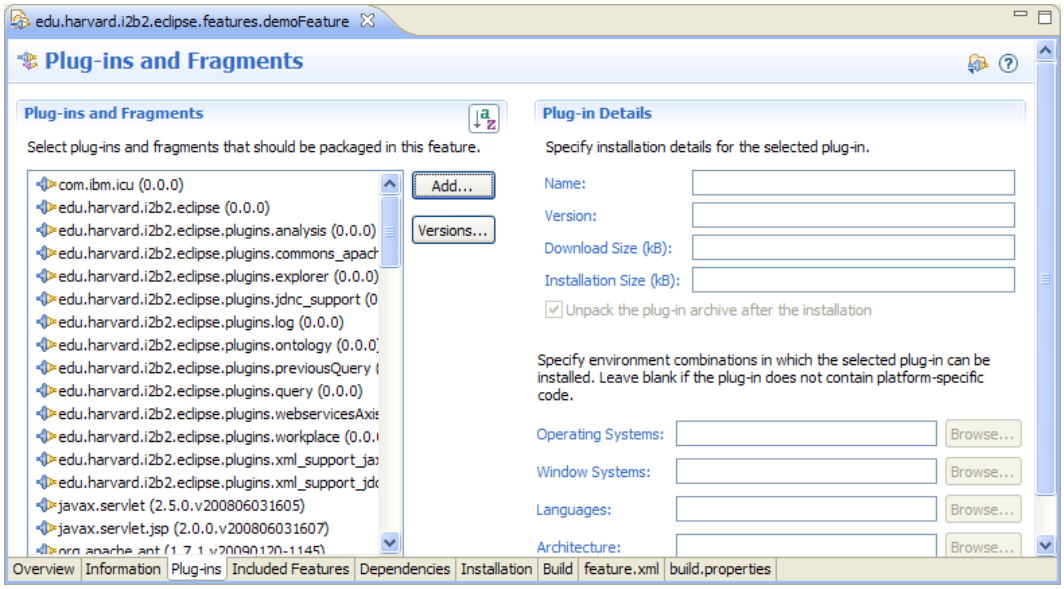
4.2.3.4 SITES TO VISIT

The **Sites to Visit** page is an optional page that you can use to provide your users with a list of additional update sites.



4.2.4 Plug-ins Tab

The plug-ins and fragments that are included in the feature are listed on the **Plug-ins and Fragments** page. The list is populated with those plug-ins selected when the feature was created.



4.2.4.1 ADDING AND REMOVING PLUG-INS AND FRAGMENTS

Add Plug-in/Fragment

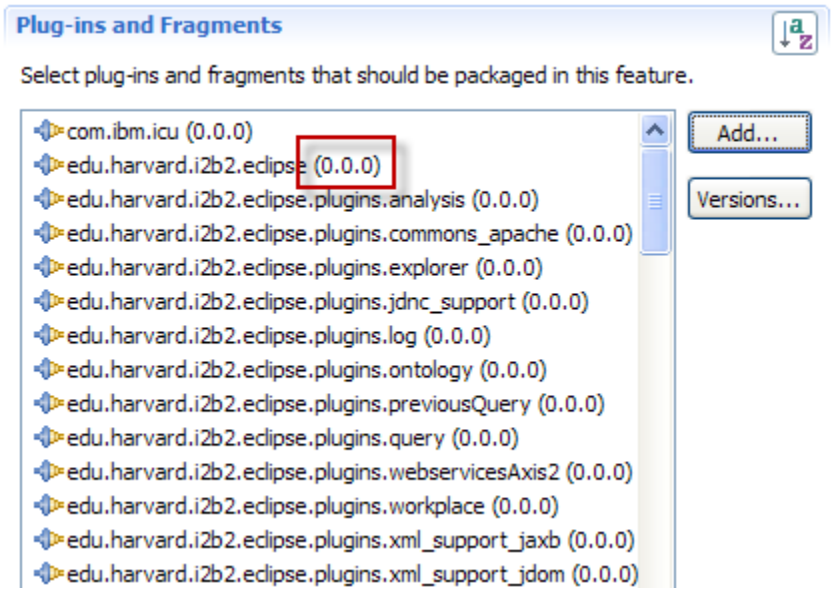
1. Click on the **Add** button and the **Plug-in Selection** dialog box will open.
2. Select the plug-in and/or fragment you want to add.
3. Click on the **OK** button.

Delete Plug-in/Fragment

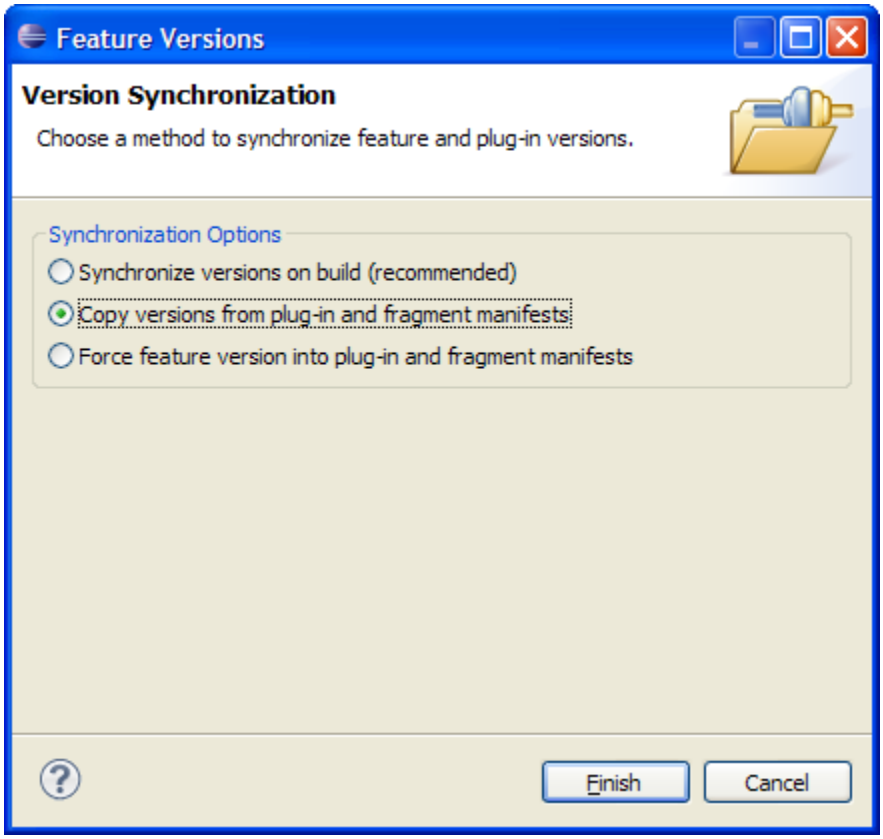
1. *Right click* on the name of the plug-in or fragment to be deleted.
2. Select **Delete** from the pop-up menu.

4.2.4.2 UPDATE VERSION OF PLUG-INS AND FRAGMENTS

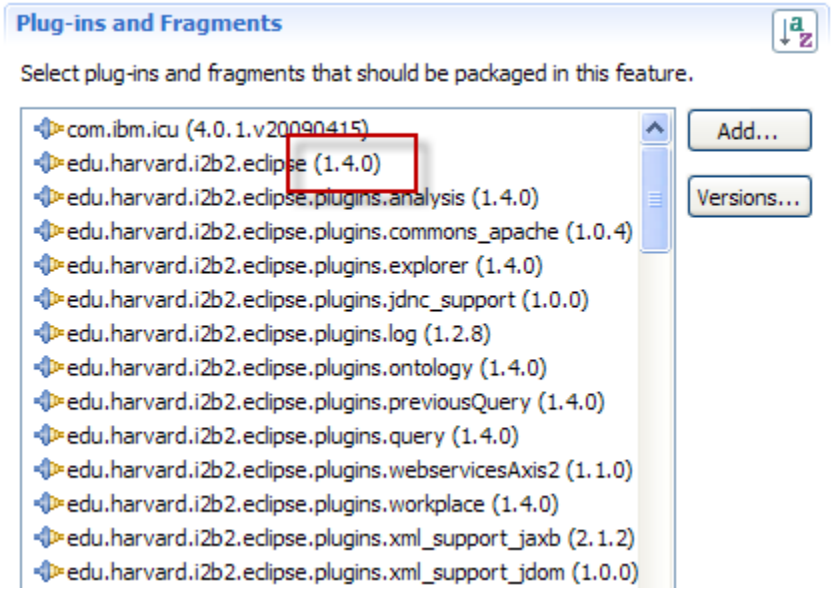
When the feature is first created the version of the plug-ins/fragments to be included is set to 0.0.0. Therefore, it is important that you update them to reflect the correct version information.



1. Select all the plug-ins/fragments in the list.
2. Click on the **Versions…** button.
3. The **Features Version** dialog box will open.



4. Select **Copy versions from plug-in and fragment manifests**.
5. Click on the **Finish** button.
6. Eclipse will update the plug-ins/fragments with the version information in the each of their respective manifests.



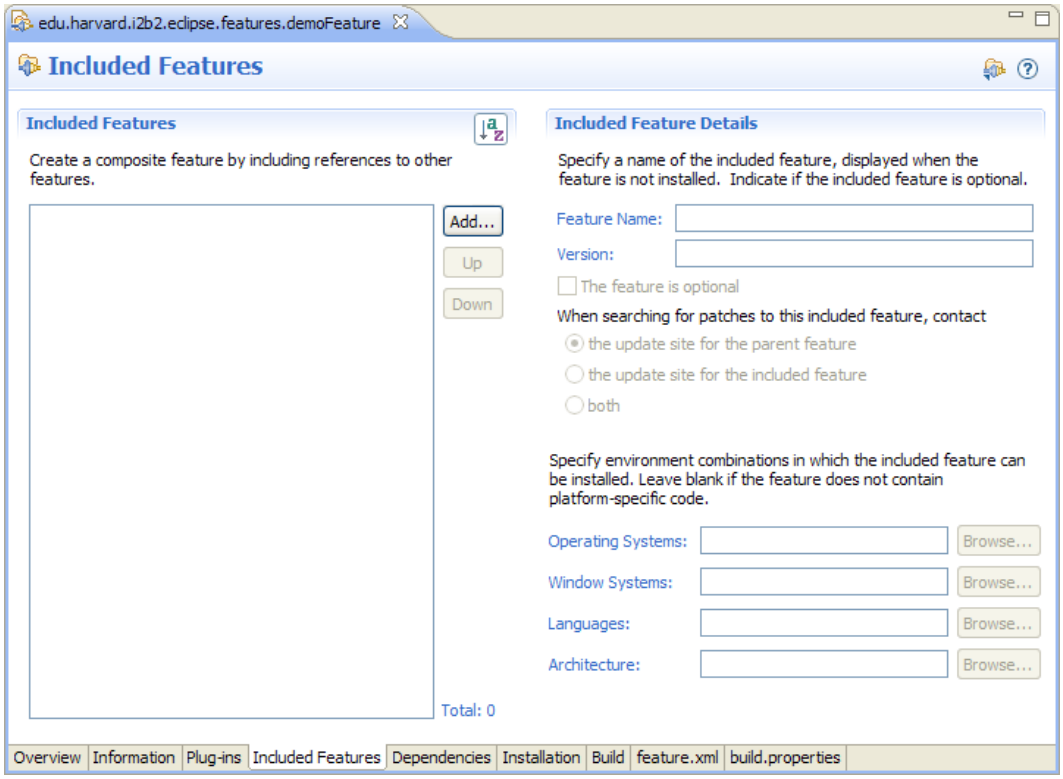
4.2.4.3 UPDATE PLUG-IN DETAILS

On the top right corner of the Plug-ins page is the **Plug-in Details** section. The information that appears is dependent upon which plug-in is highlighted in the list on the left of the page.

The *name* and *version* information defaults from the plug-in. You can add *download* and *installation sizes* as well as any *environment information* that is specific to the plug-in. All of this information is visible to the user when they are installing the feature from within the i2b2 Workbench.

4.2.5 Included Features Tab

A list of sub-features to be included with the feature is defined on the **Included Features** page.



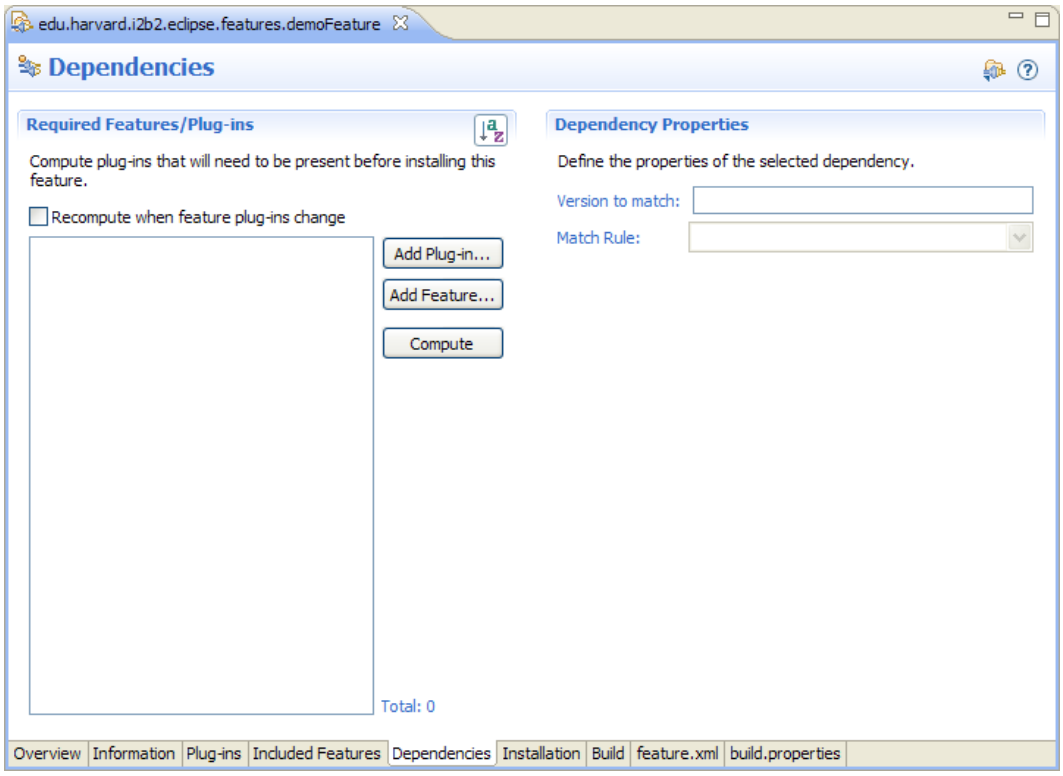
4.2.6 Dependencies Tab

All the plug-ins and other features that are required for the feature to work are listed on the **Dependencies** page. If an item on the list is missing then the feature will not load when the user tries to install it via the i2b2 Workbench.

The dependencies page is not populated when you first create the feature. Therefore, it is important that you add them before building and deploying it.

There are two ways to add dependencies:

1. ***Manually***: You select which plug-ins or features to include by clicking on the **Add Plug-in** or **Add Feature** buttons.
2. ***Compute***: This method will populate the list of dependencies based on the requirements defined by the plug-ins listed on the Plug-ins page. Simply click on the **Compute** to update the list.

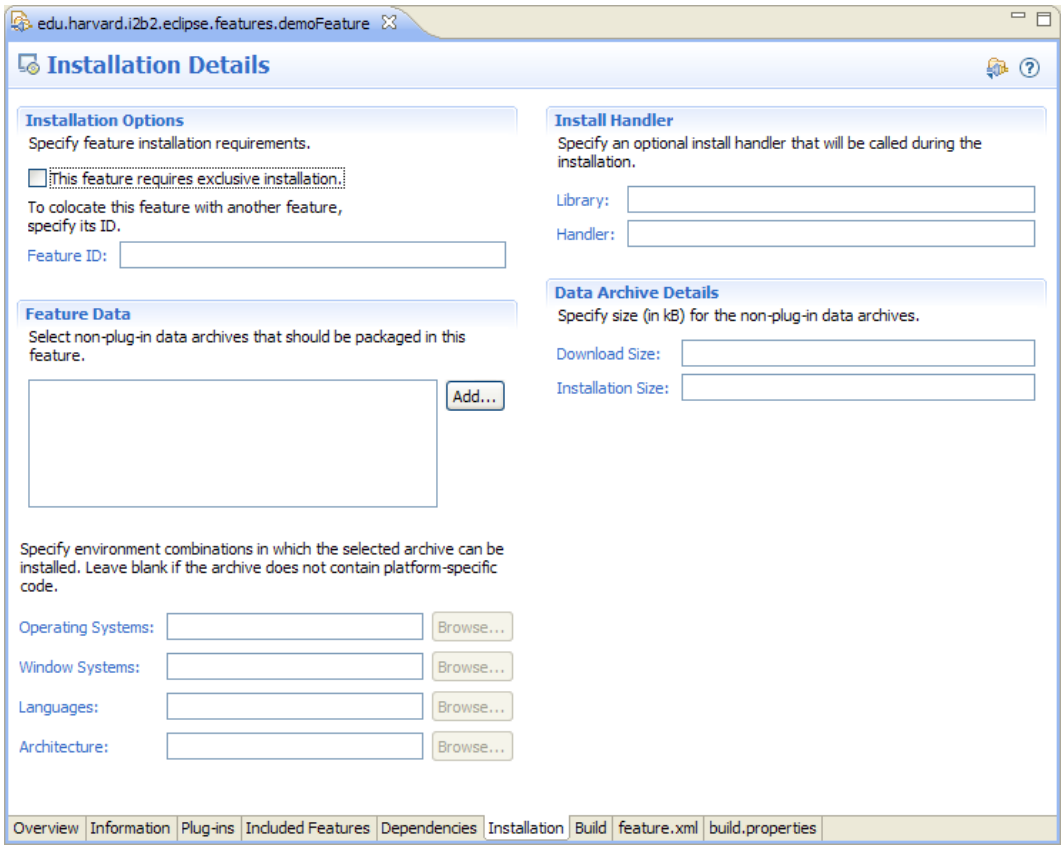


4.2.6.1 DEPENDENCY PROPERTIES

On the left side of the Dependencies page is a section called Dependency Properties. It is here that you can define a Version to match and Match Rule. The feature will not load if the version and the defined match criteria are not met.

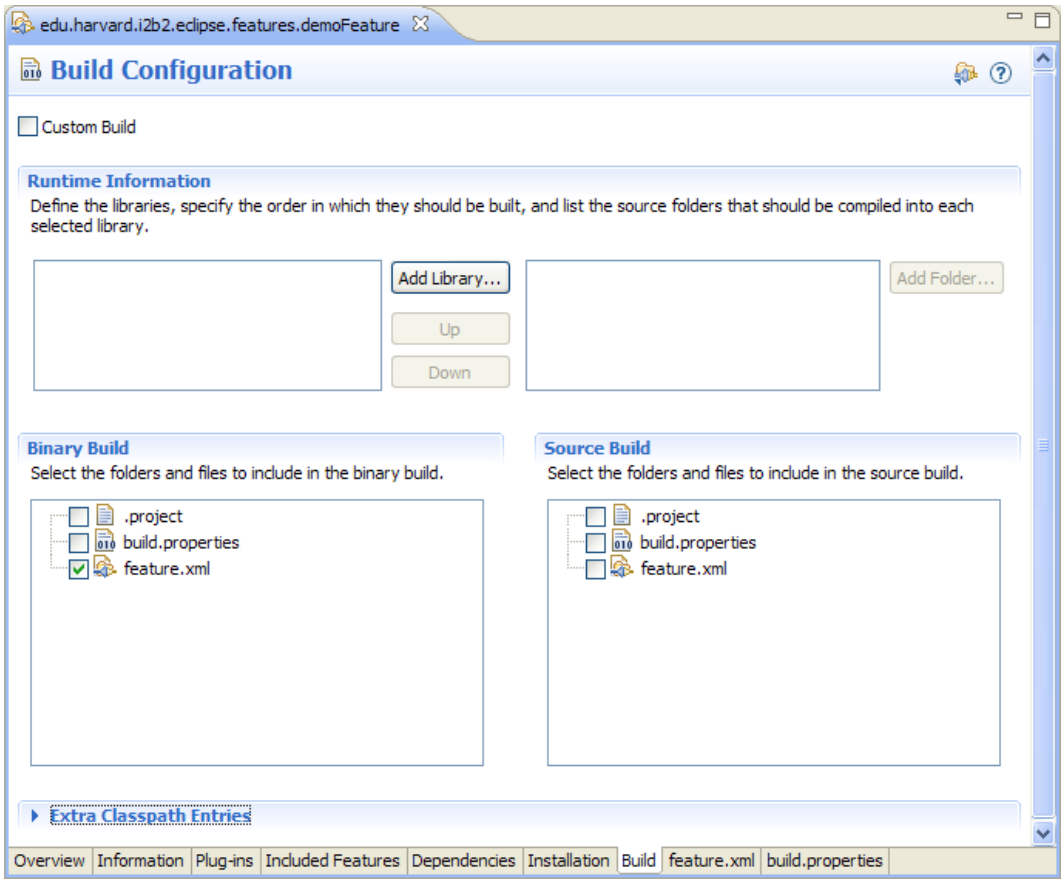
4.2.7 Installation Tab

Information regarding the installation of the feature is entered on the Installation page.



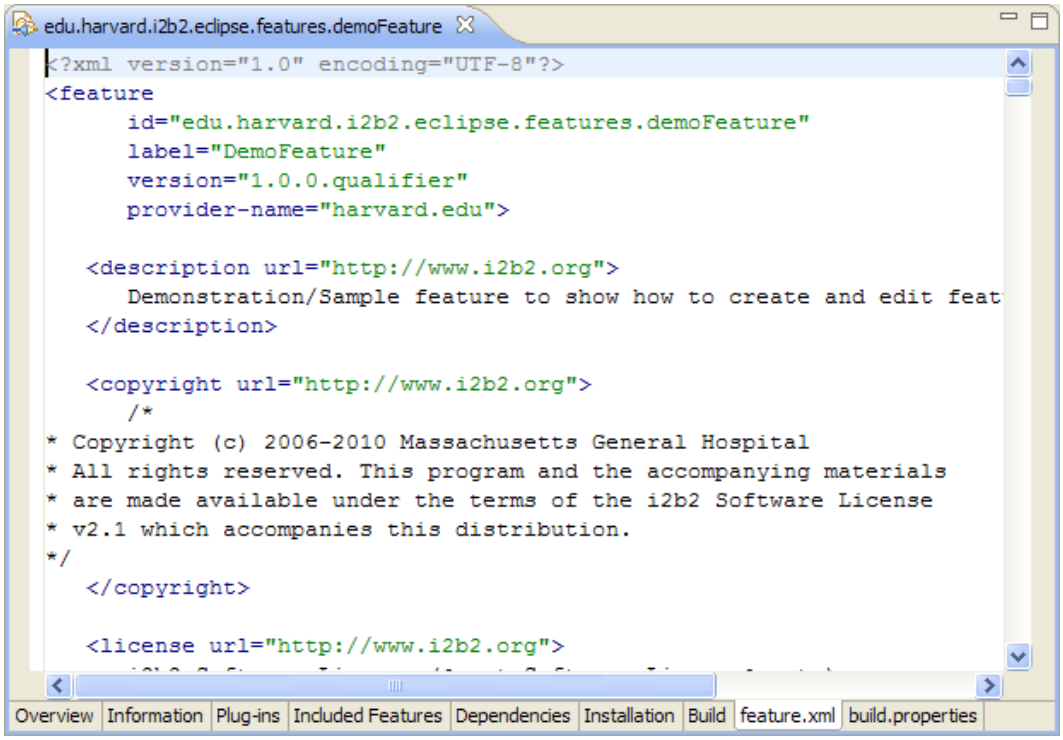
4.2.8 Build Tab

Information regarding the *libraries*, *folders* and *files* to include when building the feature are defined on the **Build Configuration** page.



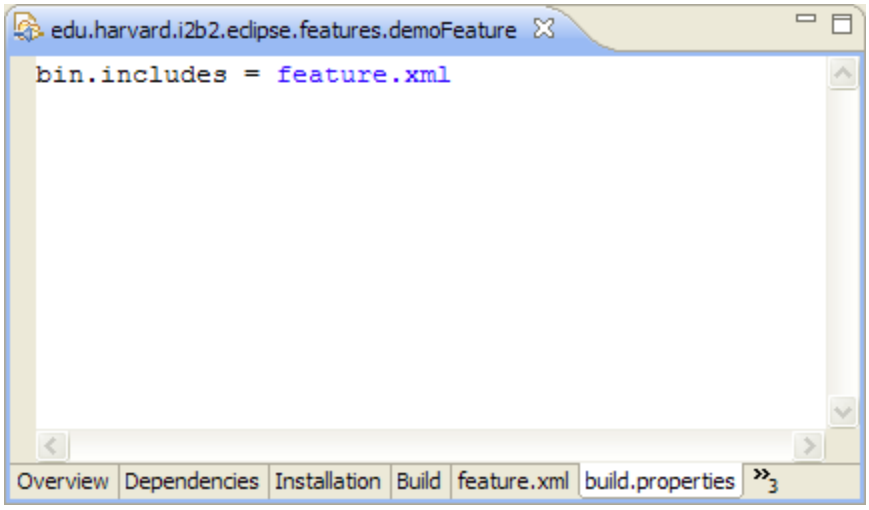
4.2.9 Feature.xml Tab

The **feature.xml** page is the XML for all the information defined on the other pages.



4.2.10 Build.properties Tab

The **build.properties** page is the properties file for the feature. The information that appears on this page correlates with what is defined on the **Build Configuration** page.



LICENSE

The i2b2 source code is licensed under the i2b2 Software License 2.1. This includes but is not limited to all code in the edu.harvard.i2b2.* package namespace.