Informatics for Integrating Biology and the Bedside

Developer’s Guide

i2b2 Workbench

Document Version: *1.6.1*
I2b2 Software Release: *1.6*

Table of Contents

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.6.1	07/22/10	Janice Donahoe	Created 1.6 version of document.

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.***

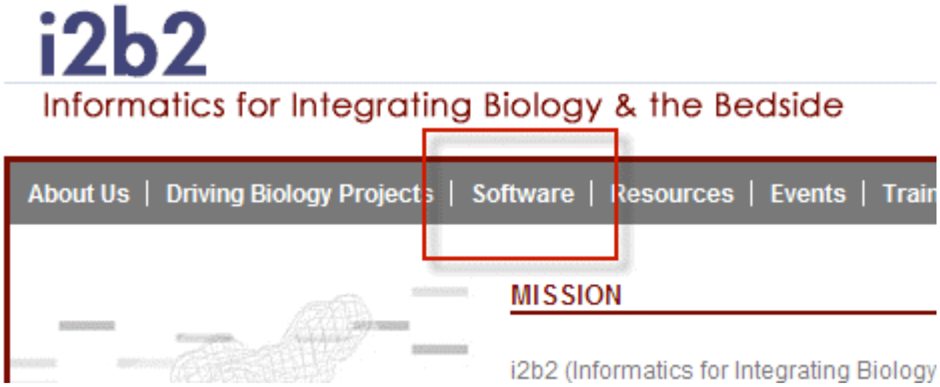
Further information regarding the eclipse product can be found on their website (http://www.eclipse.org/documentation/).

1.2 i2b2 Software

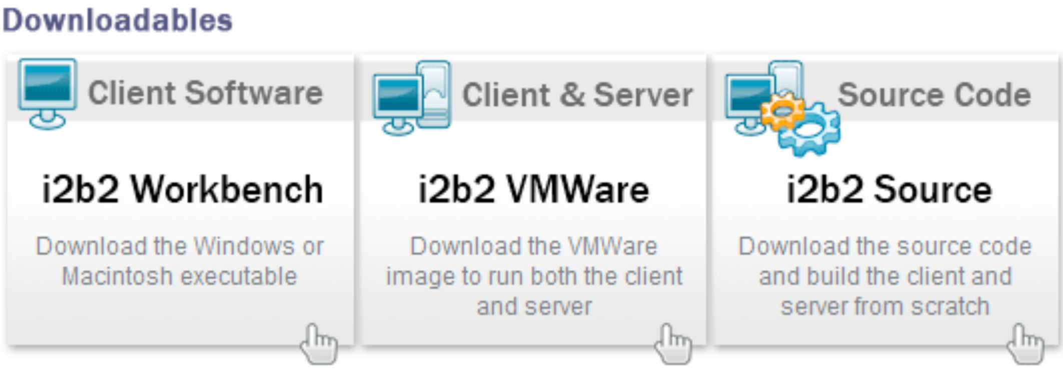
1.2.1 i2b2 Workbench Source Code

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*.


i2b2
Informatics for Integrating Biology & the Bedside
About Us
Driving Biology Projects
Software
Resources
Events
Train
MISSION
i2b2 (Informatics for Integrating Biology

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

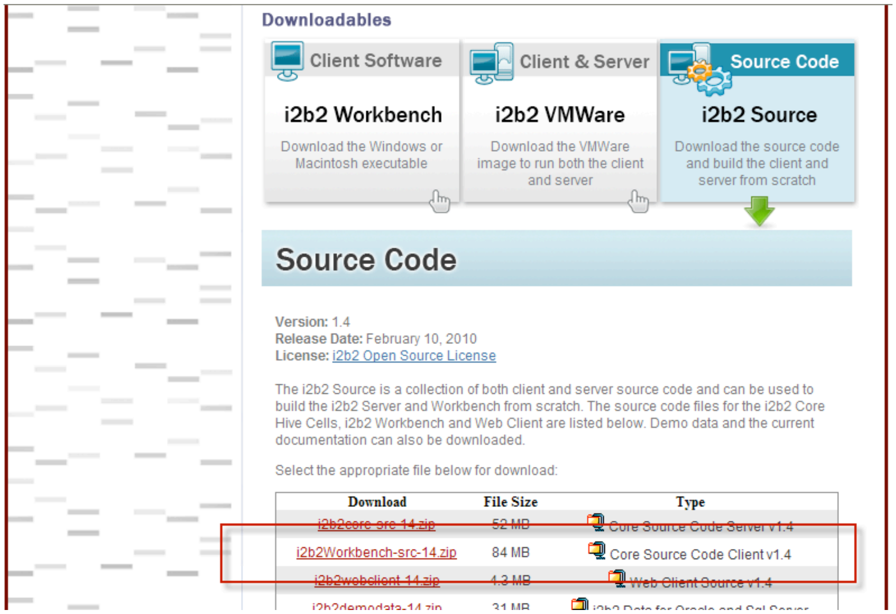


Download the Windows or Macintosh executable

Download the VMWare image to run both the client and server

Download the source code and build the client and server from scratch

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



i2b2 Workbench

Download the Windows or Macintosh executable

i2b2 VMWare

Download the VMWare image to run both the client and server

i2b2 Source

Download the source code and build the client and server from scratch

Source Code

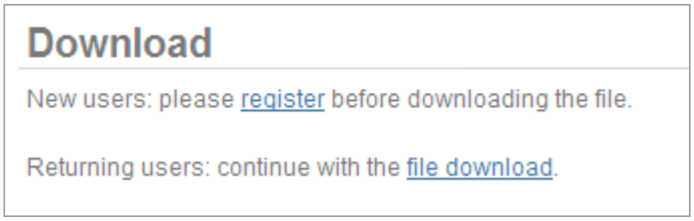
Version: 1.4
Release Date: February 10, 2010
License: i2b2 Open Source License

The i2b2 Source is a collection of both client and server source code and can be used to build the i2b2 Server and Workbench from scratch. The source code files for the i2b2 Core Hive Cells, i2b2 Workbench and Web Client are listed below. Demo data and the current documentation can also be downloaded.

Select the appropriate file below for download:

Download	File Size	Type
i2b2core-src-14.zip	52 MB	Core Source Code Server v1.4
i2b2Workbench-src-14.zip	84 MB	Core Source Code Client v1.4
i2b2webclient-14.zip	4.3 MB	Web Client Source v1.4
i2b2demodata-14.zip	31 MB	i2b2 Data for Oracle and Sql Server

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.



Download

New users: please register before downloading the file.

Returning users: continue with the file download.

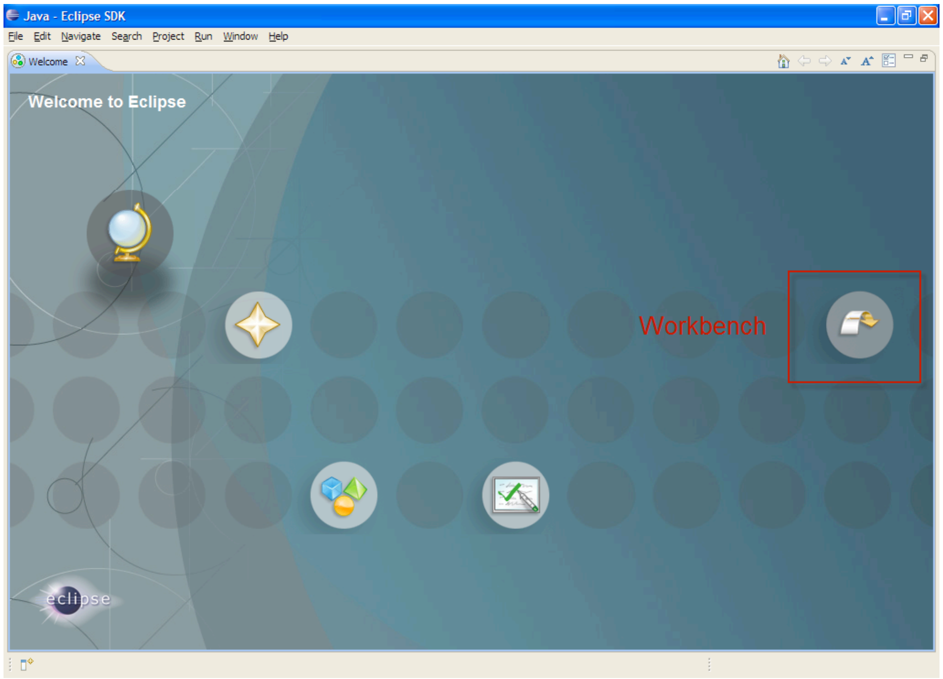
8. The next step is to review the i2b2 license and choose to either accept or decline.
   a. If you choose to **Decline** you will be brought back to the main i2b2 Software page.
   b. If you agree with the license click on **Accept** to continue with the download.

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".

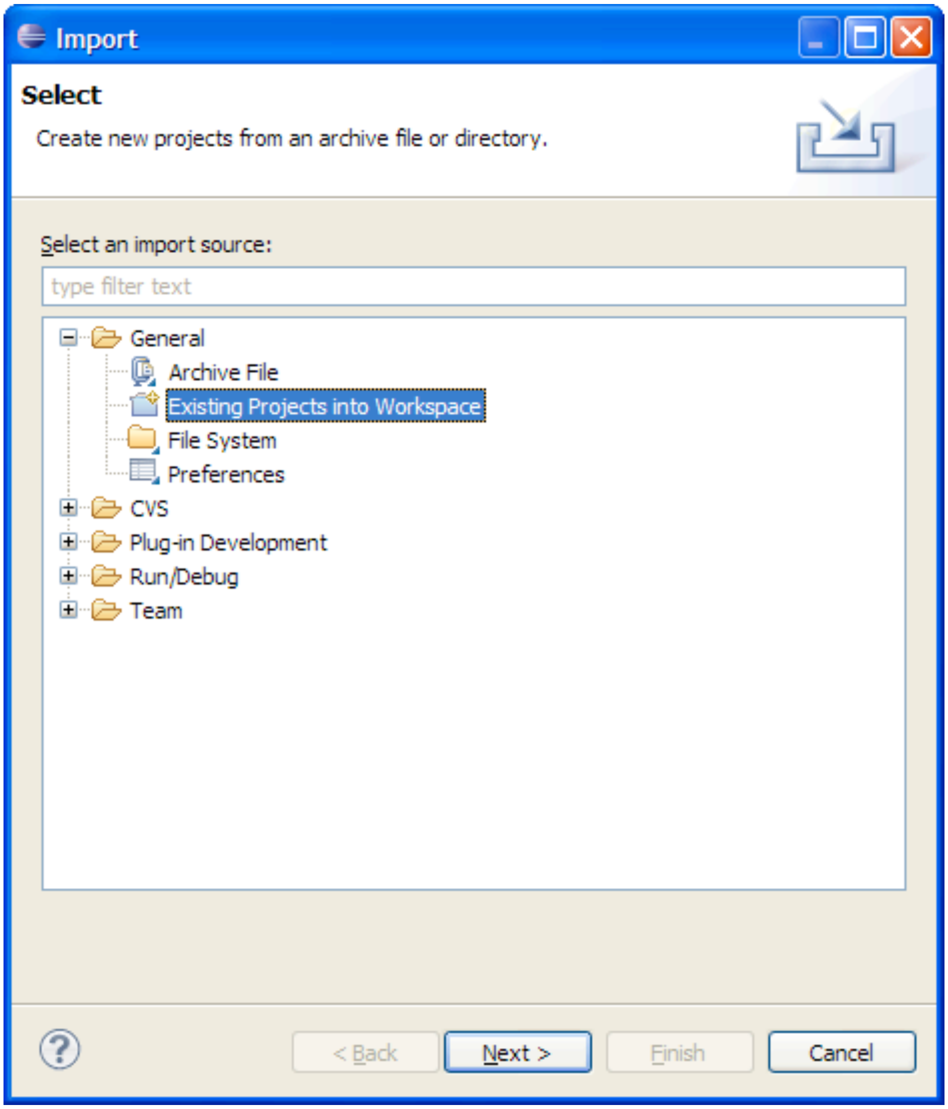
2.1.2 Importing the i2b2 Workbench projects

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

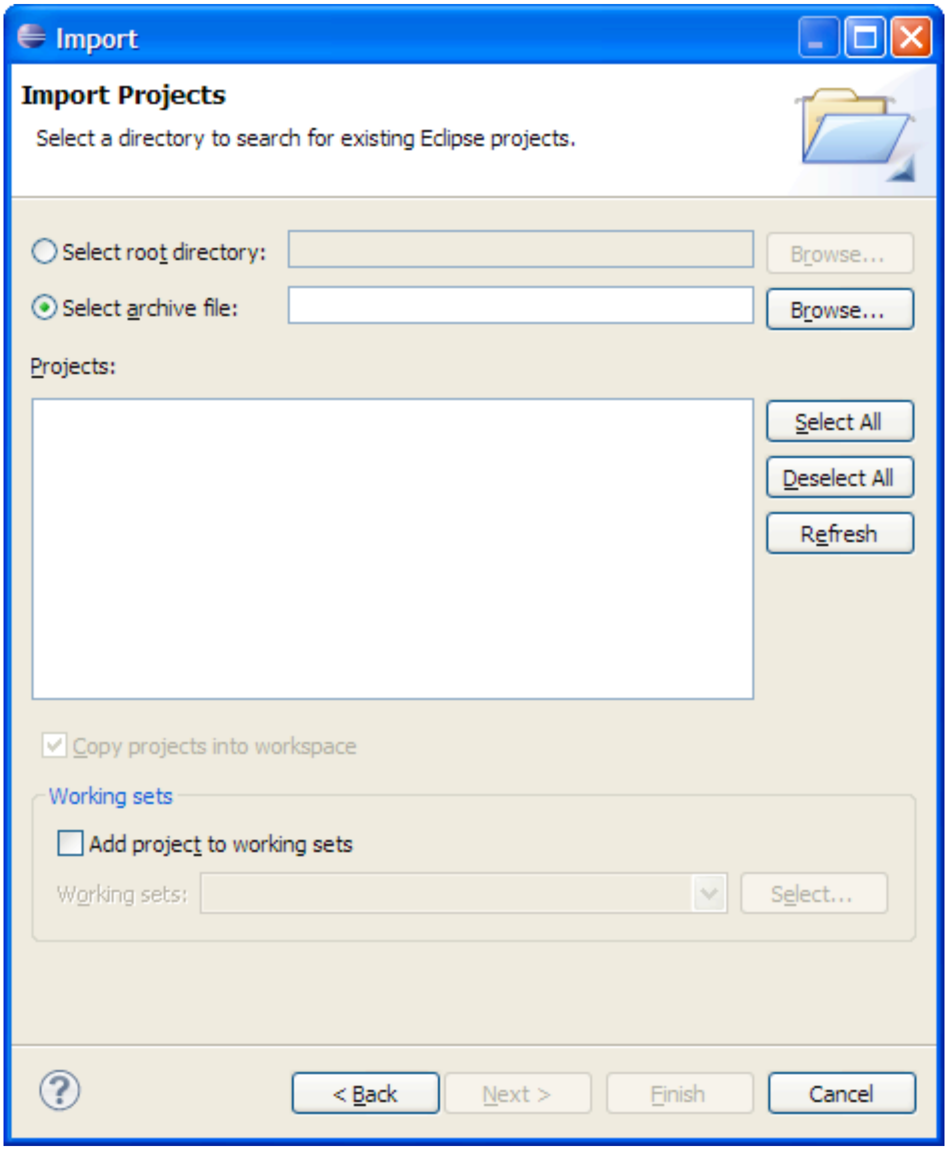
    Example (correct): c:\workspace\i2b2workbench

    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**.

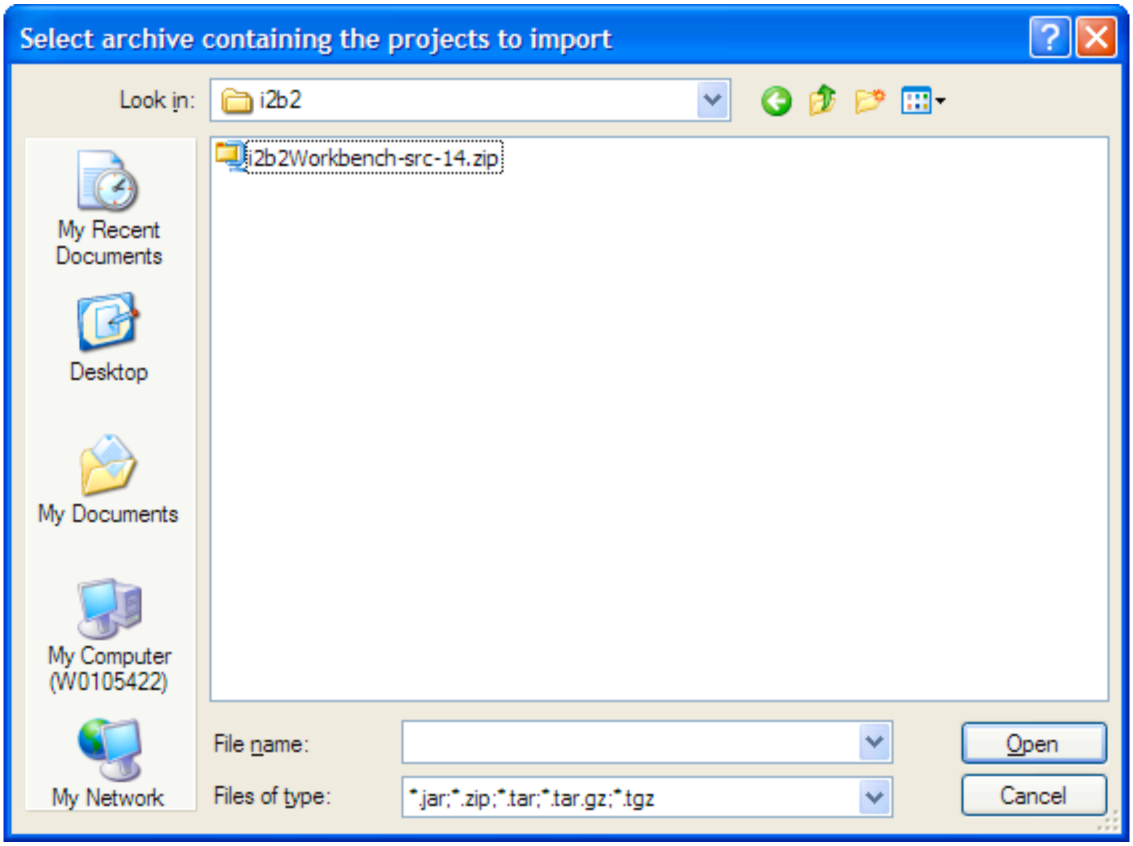


Select a directory to search for existing Eclipse projects.

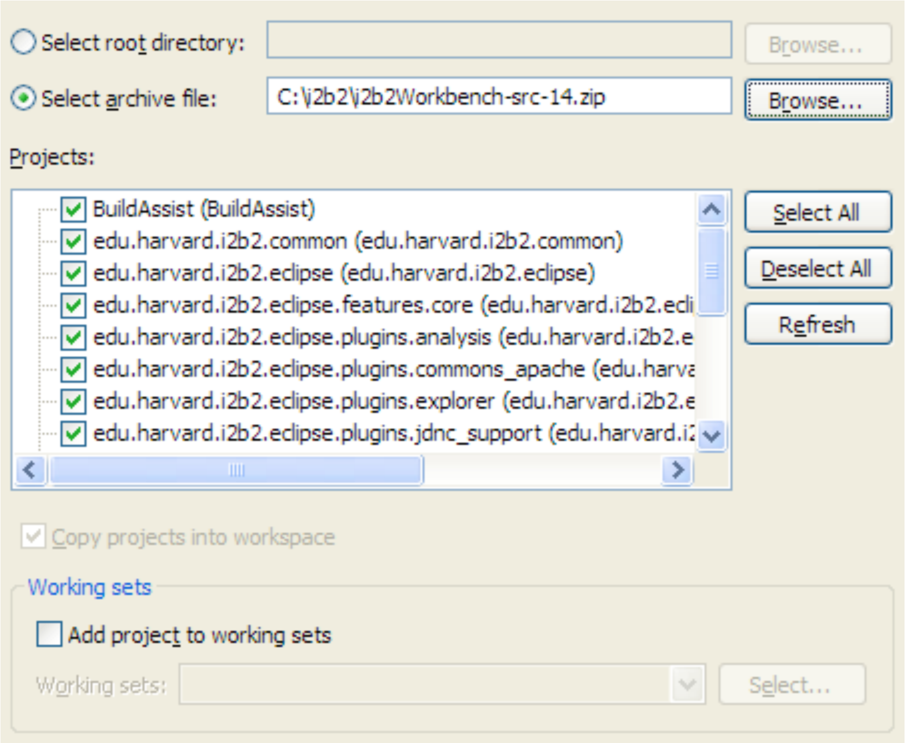
● Select archive file:

☑ Copy projects into workspace

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*.



○ Select root directory:

● Select archive file: C:\i2b2\i2b2Workbench-src-14.zip

Projects:

- ☑ BuildAssist (BuildAssist)
- ☑ edu.harvard.i2b2.common (edu.harvard.i2b2.common)
- ☑ edu.harvard.i2b2.eclipse (edu.harvard.i2b2.eclipse)
- ☑ edu.harvard.i2b2.eclipse.features.core (edu.harvard.i2b2.ecli
- ☑ edu.harvard.i2b2.eclipse.plugins.analysis (edu.harvard.i2b2.e
- ☑ edu.harvard.i2b2.eclipse.plugins.commons_apache (edu.harva
- ☑ edu.harvard.i2b2.eclipse.plugins.explorer (edu.harvard.i2b2.e
- ☑ edu.harvard.i2b2.eclipse.plugins.jdnc_support (edu.harvard.i2

☑ Copy projects into workspace

Working sets

☐ Add project to working sets

Working sets:

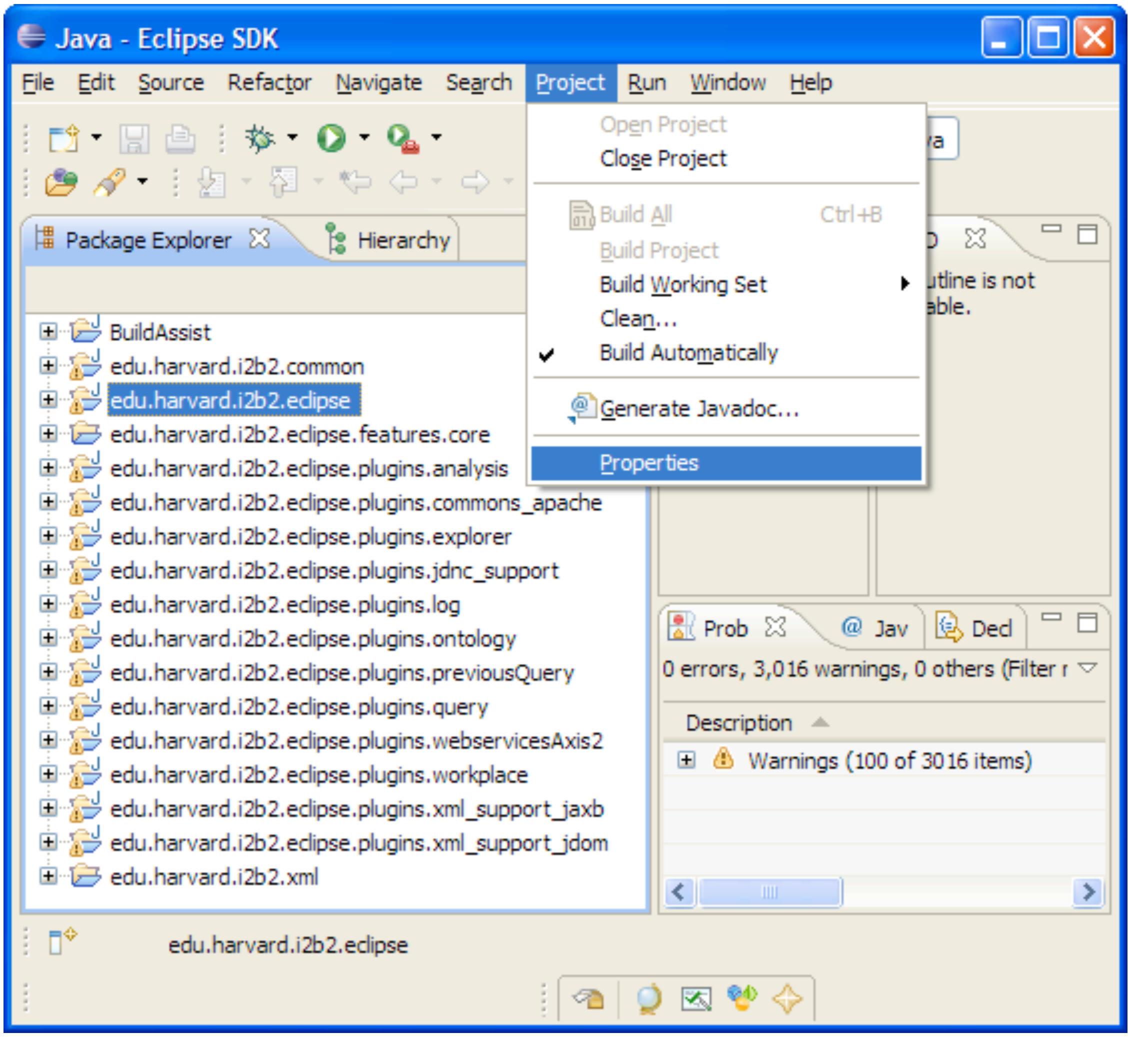
10. Verify all the projects are checked off
11. Click on the **Finish** button.
12. Once the projects have finished importing they will appear in your Eclipse Workbench.

2.2 Building the i2b2 Workbench within the Eclipse IDE

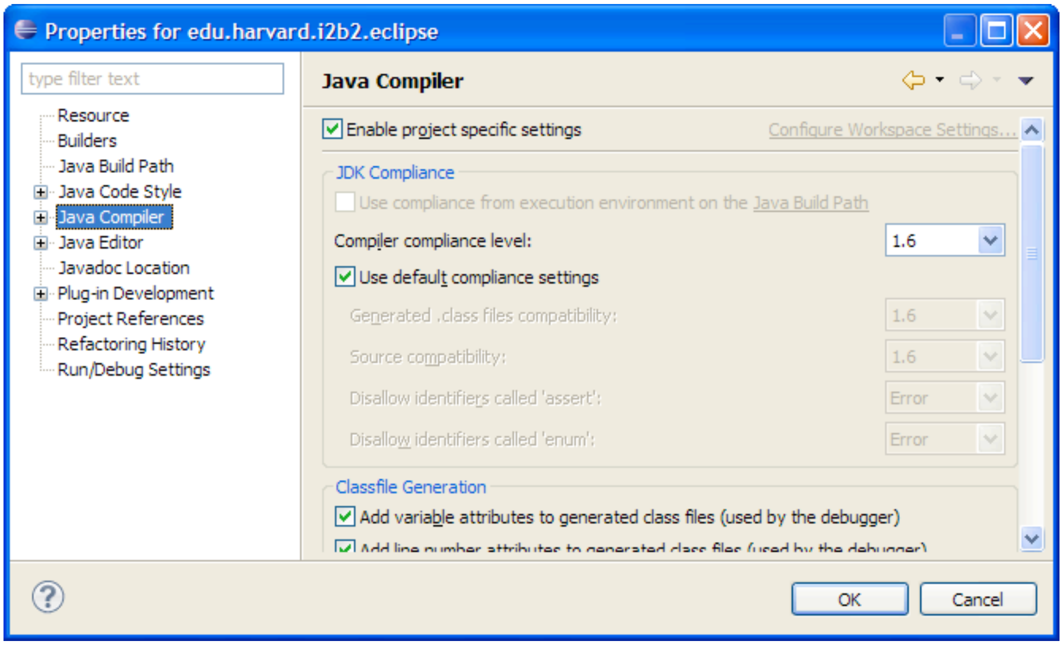
2.2.1 Java Compiler Settings

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.


Java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Open Project
Close Project
Build All Ctrl+B
Build Project
Build Working Set
Clean...
Build Automatically
Generate Javadoc...
Properties
Package Explorer
Hierarchy
BuildAssist
edu.harvard.i2b2.common
edu.harvard.i2b2.eclipse
edu.harvard.i2b2.eclipse.features.core
edu.harvard.i2b2.eclipse.plugins.analysis
edu.harvard.i2b2.eclipse.plugins.commons_apache
edu.harvard.i2b2.eclipse.plugins.explorer
edu.harvard.i2b2.eclipse.plugins.jdnc_support
edu.harvard.i2b2.eclipse.plugins.log
edu.harvard.i2b2.eclipse.plugins.ontology
edu.harvard.i2b2.eclipse.plugins.previousQuery
edu.harvard.i2b2.eclipse.plugins.query
edu.harvard.i2b2.eclipse.plugins.webservicesAxis2
edu.harvard.i2b2.eclipse.plugins.workplace
edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb
edu.harvard.i2b2.eclipse.plugins.xml_support_jdom
edu.harvard.i2b2.xml
utline is not
able.
Prob
Jav
Ded
0 errors, 3,016 warnings, 0 others (Filter r
Description
Warnings (100 of 3016 items)
edu.harvard.i2b2.eclipse

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



Java Compiler

☑ Enable project specific settings

JDK Compliance

☐ Use compliance from execution environment on the Java Build Path

Compiler compliance level: 1.6

☑ Use default compliance settings

Generated .class files compatibility: 1.6

Source compatibility: 1.6

Disallow identifiers called 'assert': Error

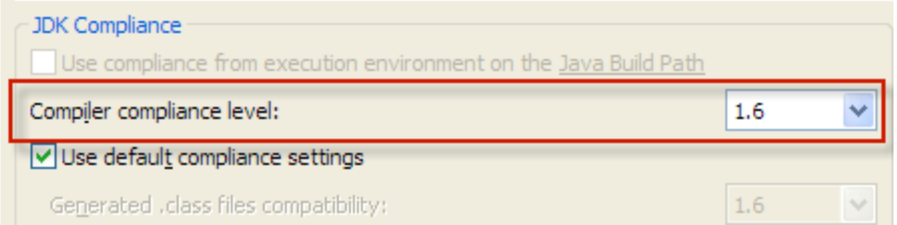
Disallow identifiers called 'enum': Error

Classfile Generation

☑ Add variable attributes to generated class files (used by the debugger)

☑ Add line number attributes to generated class files (used by the debugger)

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



JDK Compliance

☐ Use compliance from execution environment on the Java Build Path

Compiler compliance level: 1.6

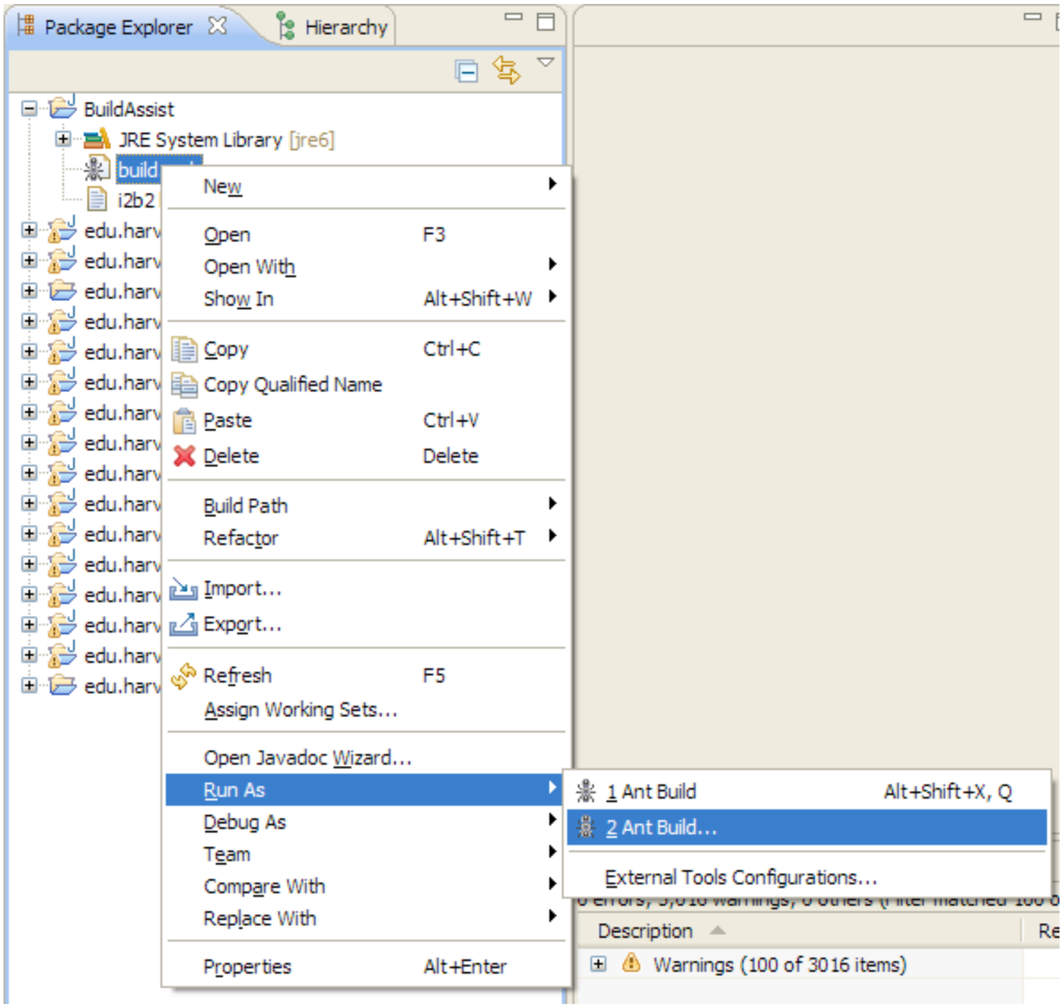
☑ Use default compliance settings

Generated .class files compatibility: 1.6

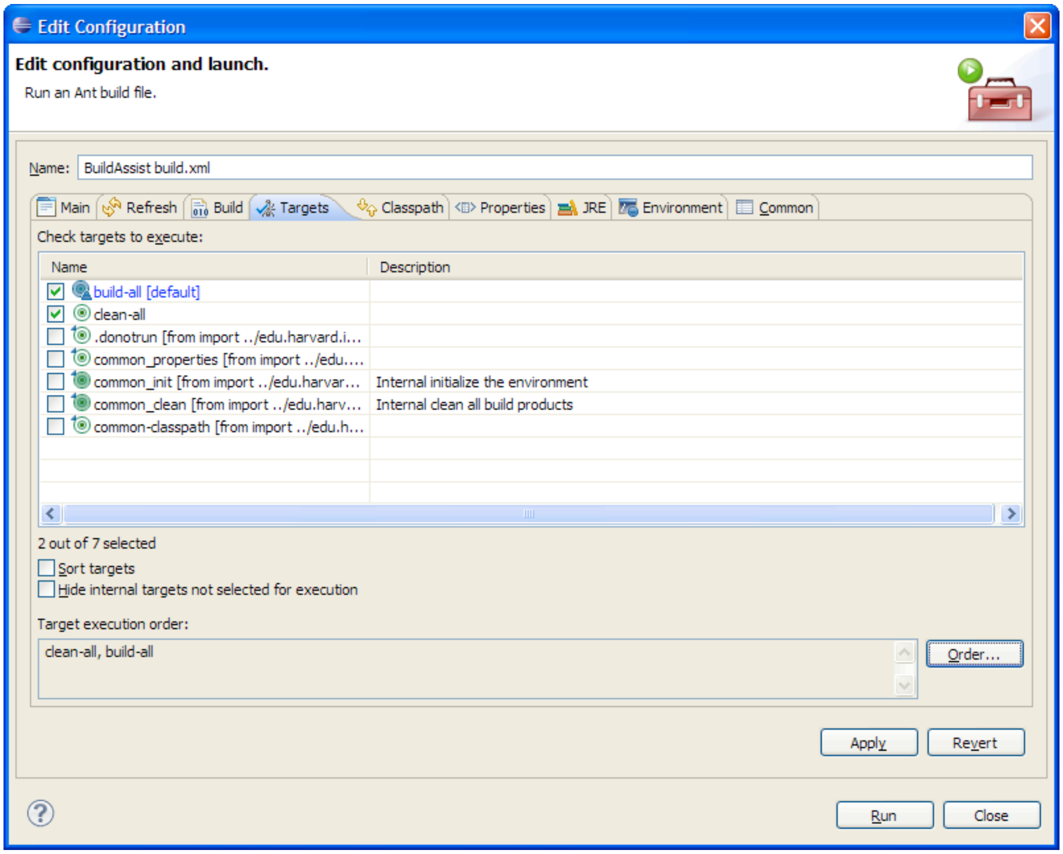
2.2.2 Cleaning and Building the Projects

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”



Name: BuildAssist build.xml

Check targets to execute:

	Name	Description
☑	build-all [default]	
☑	clean-all	
☐	.donotrun [from import ../edu.harvard.i...	
☐	common_properties [from import ../edu....	
☐	common_init [from import ../edu.harvar...	Internal initialize the environment
☐	common_clean [from import ../edu.harv...	Internal clean all build products
☐	common-classpath [from import ../edu.h...	

2 out of 7 selected

☐ Sort targets

☐ Hide internal targets not selected for execution

Target execution order:

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

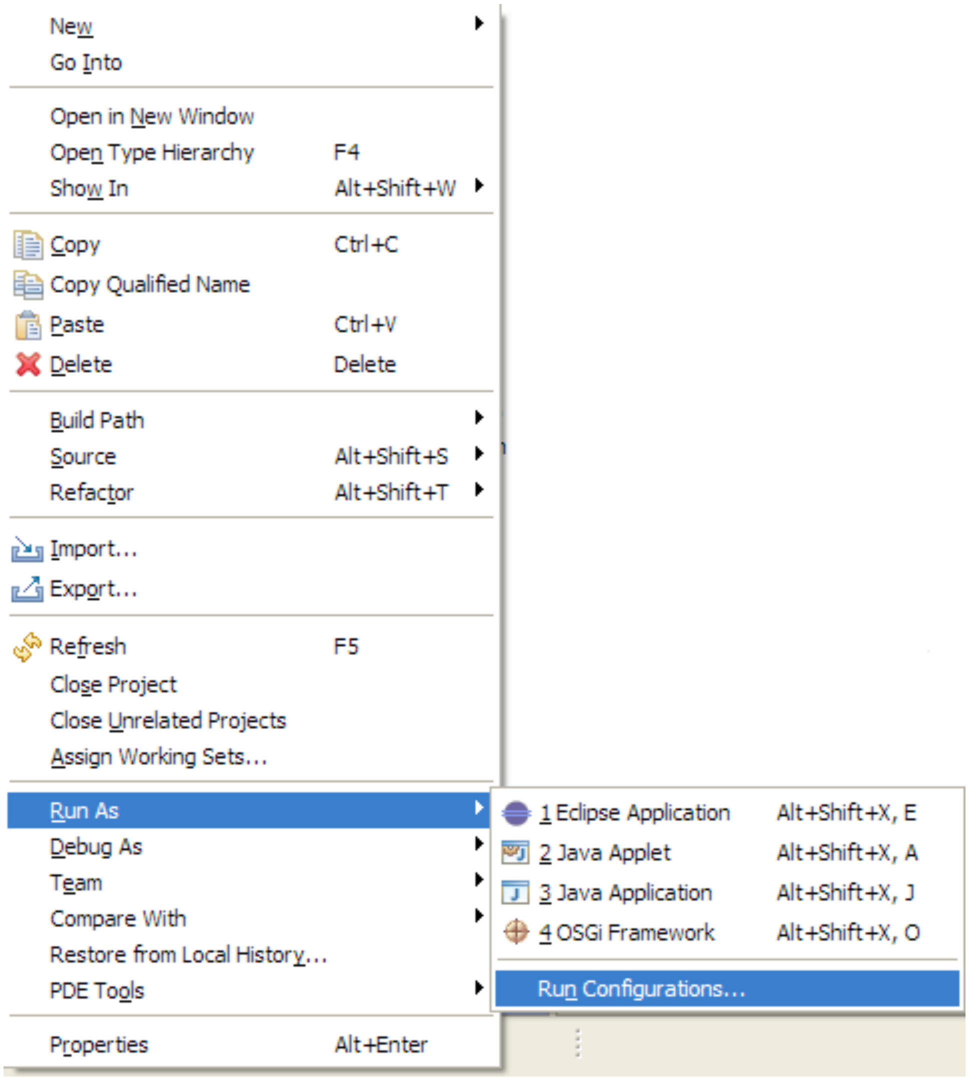
2.3.1 Define Run Configurations

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.



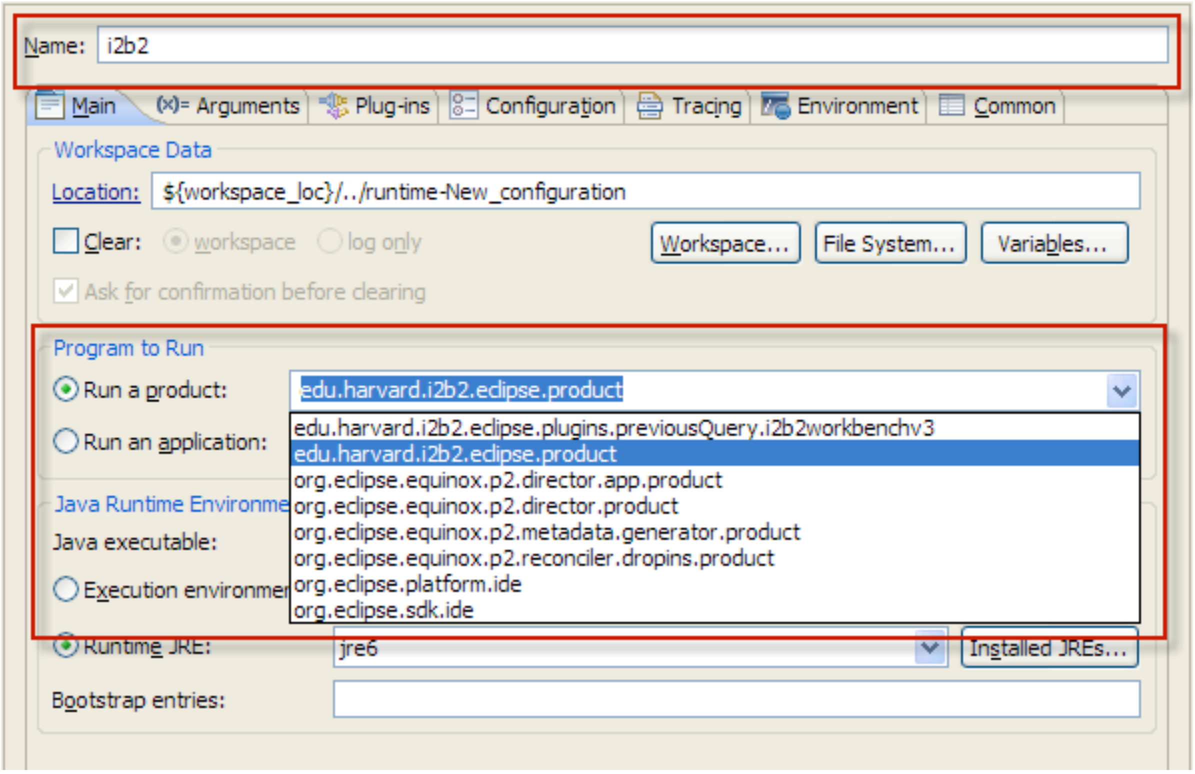
The next three sections outline the process of setting up the configuration information on the *Main*, *Arguments*, and *Plug-ins* tabs.

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.



Name: i2b2

Workspace Data

Location: ${workspace_loc}/../runtime-New_configuration

☐ Clear: ● workspace ○ log only

☑ Ask for confirmation before clearing

Program to Run

● Run a product: edu.harvard.i2b2.eclipse.product

○ Run an application:

edu.harvard.i2b2.eclipse.plugins.previousQuery.i2b2workbenchv3
edu.harvard.i2b2.eclipse.product
org.eclipse.equinox.p2.director.app.product
org.eclipse.equinox.p2.director.product
org.eclipse.equinox.p2.metadata.generator.product
org.eclipse.equinox.p2.reconciler.dropins.product
org.eclipse.platform.ide
org.eclipse.sdk.ide

Java Runtime Environme

Java executable:

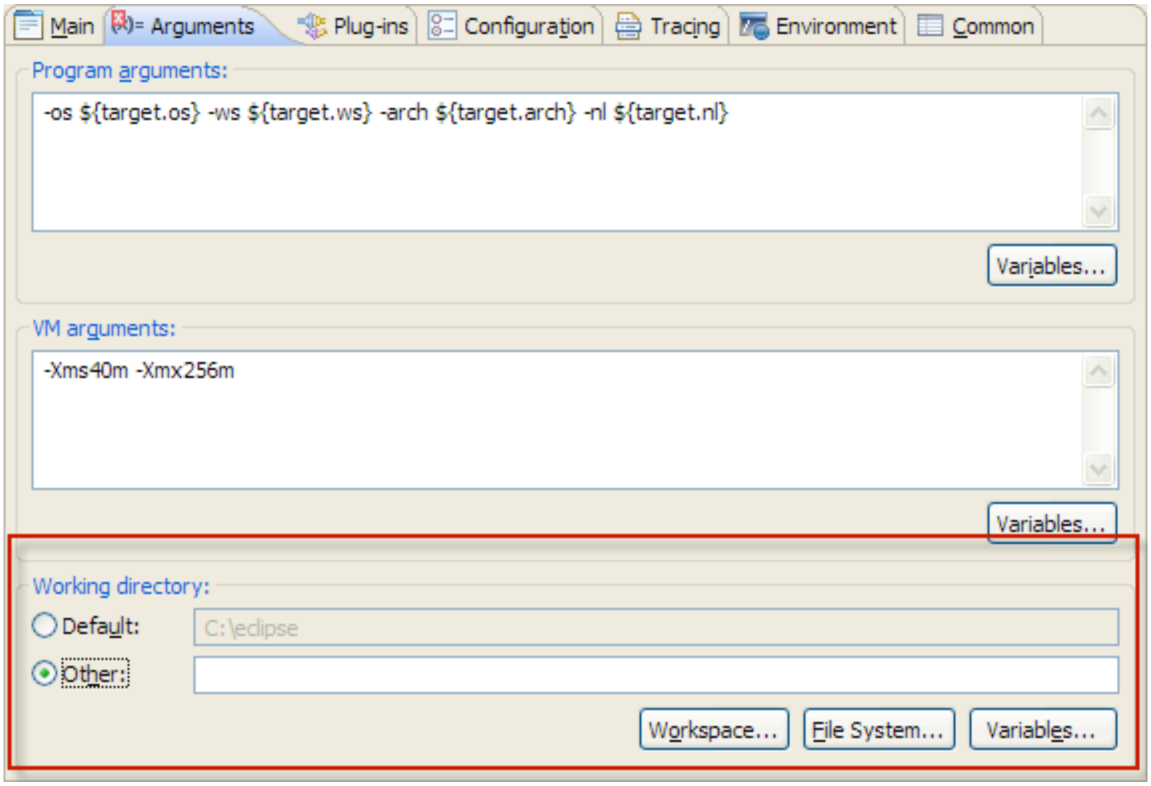
○ Execution environme

● Runtime JRE: jre6

Bootstrap entries:

2.3.1.2 ARGUMENTS TAB

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



Program arguments:

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

VM arguments:

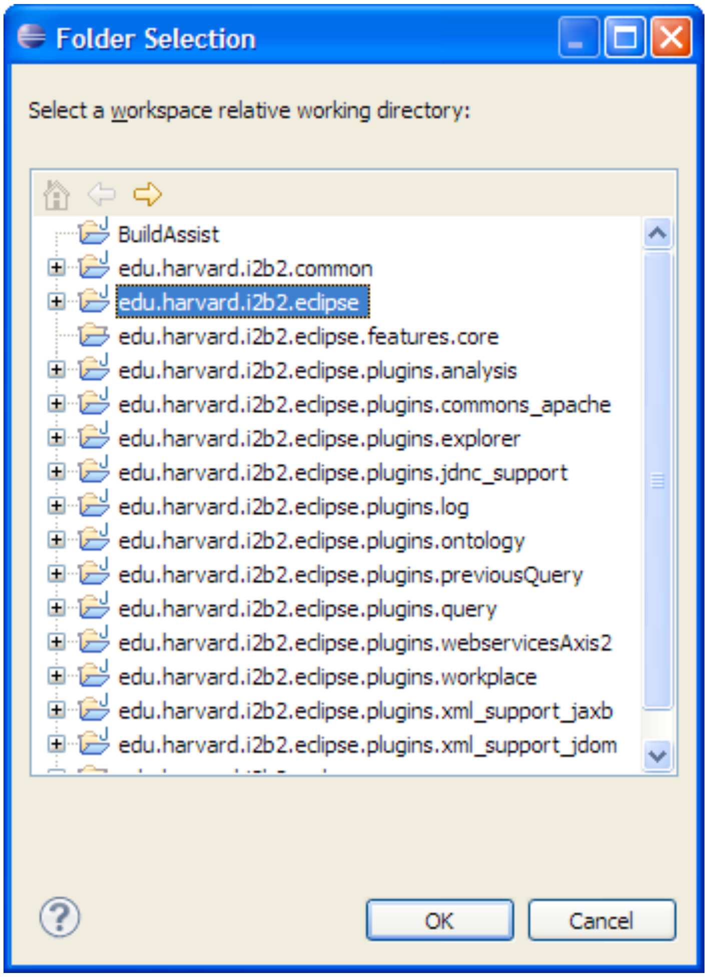
```
-Xms40m -Xmx256m
```

Working directory:

○ Default: C:\eclipse

● 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.

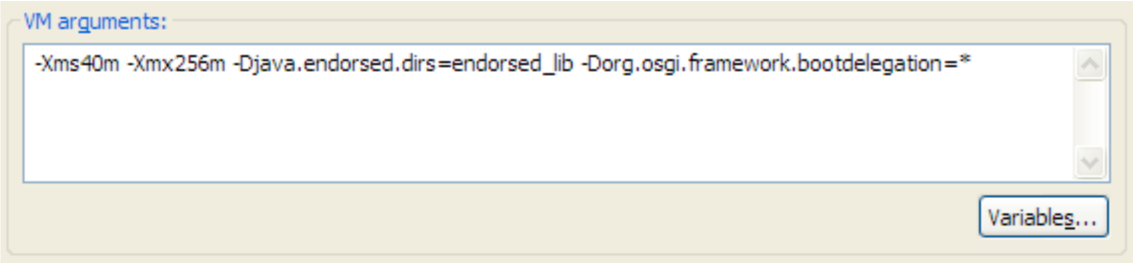


Select a workspace relative working directory:

- BuildAssist
- edu.harvard.i2b2.common
- edu.harvard.i2b2.eclipse
- edu.harvard.i2b2.eclipse.features.core
- edu.harvard.i2b2.eclipse.plugins.analysis
- edu.harvard.i2b2.eclipse.plugins.commons_apache
- edu.harvard.i2b2.eclipse.plugins.explorer
- edu.harvard.i2b2.eclipse.plugins.jdnc_support
- edu.harvard.i2b2.eclipse.plugins.log
- edu.harvard.i2b2.eclipse.plugins.ontology
- edu.harvard.i2b2.eclipse.plugins.previousQuery
- edu.harvard.i2b2.eclipse.plugins.query
- edu.harvard.i2b2.eclipse.plugins.webservicesAxis2
- edu.harvard.i2b2.eclipse.plugins.workplace
- edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb
- edu.harvard.i2b2.eclipse.plugins.xml_support_jdom

6. In the “**VM arguments:**” section add the following:

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



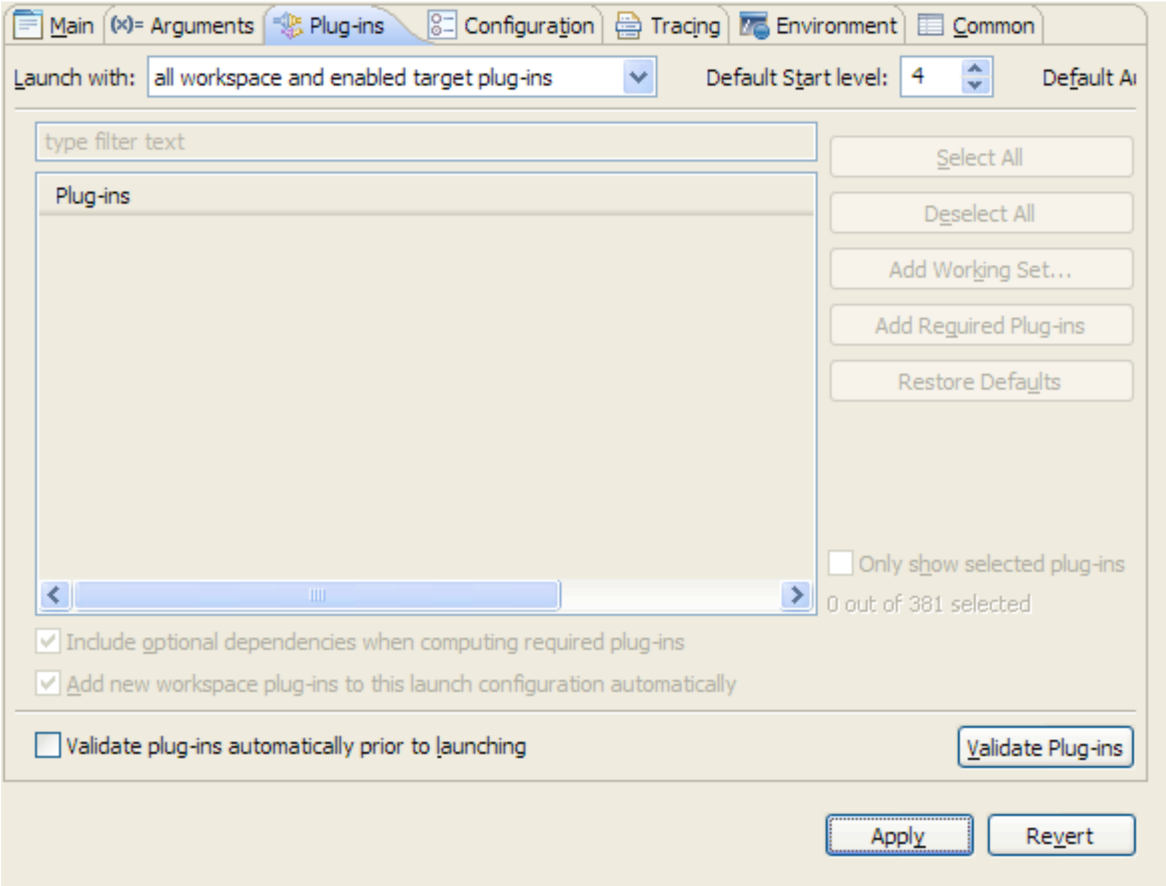
VM arguments:

```
-Xms40m -Xmx256m -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.



Launch with: all workspace and enabled target plug-ins

Default Start level: 4

☐ Only show selected plug-ins

0 out of 381 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

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

2.3.2 Running the i2b2 Workbench Application

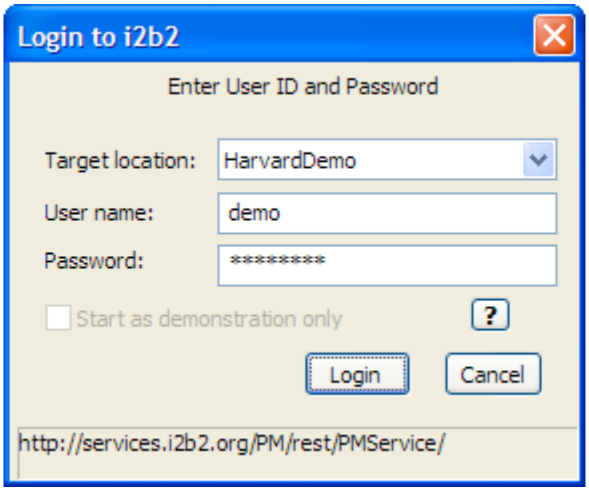
The following steps outline the process of running and logging into the i2b2 Workbench.

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.

i2b2 workbench
1.4
© Copyright i2b2 contributors and others 2005-2009. All rights reserved.

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


Login to i2b2

Enter User ID and Password

Target location: HarvardDemo
User name: demo
Password: *********
☐ Start as demonstration only

http://services.i2b2.org/PM/rest/PMService/

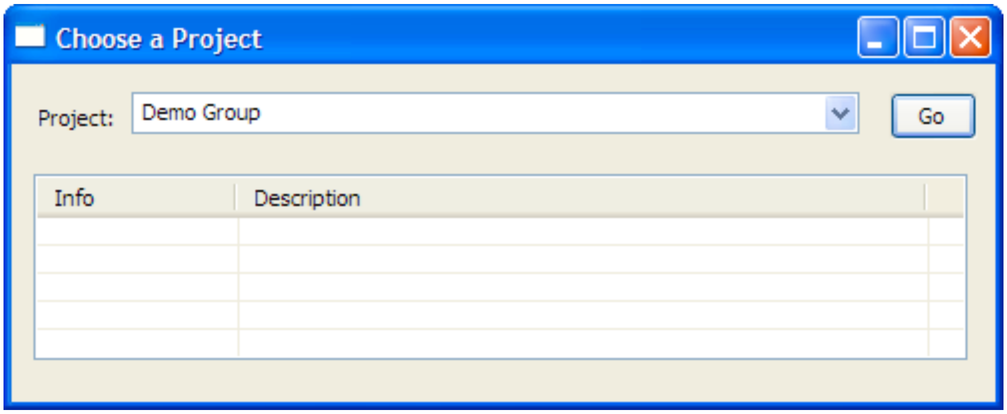
4. Enter the following information:

    User name: demo

    Password: demouser

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

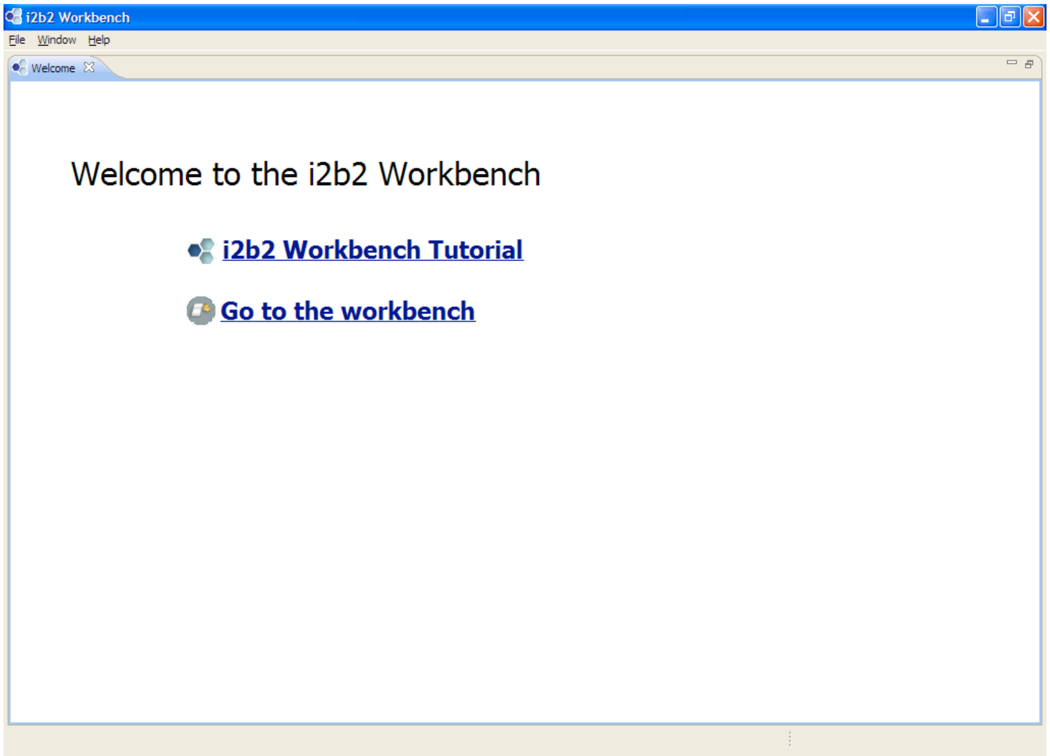
The project dialog box will only appear if you have access to multiple projects. If you only have access to one project you will be brought to the Welcome page.



Project: Demo Group

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.

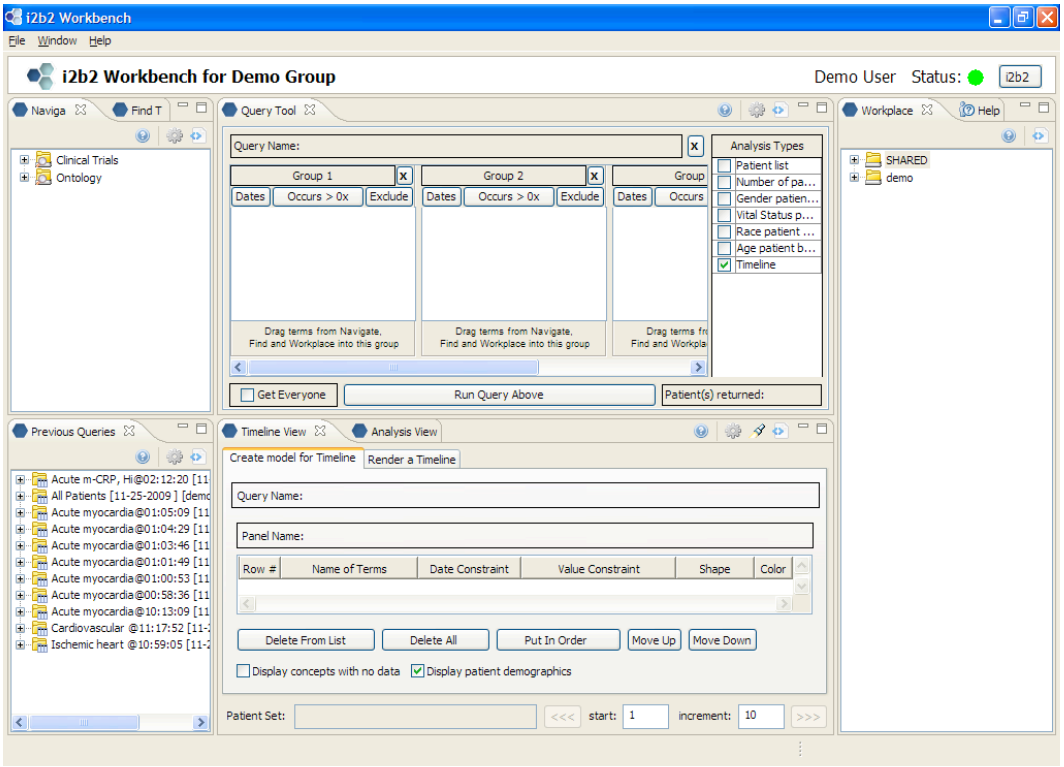


Welcome to the i2b2 Workbench

i2b2 Workbench Tutorial

Go to the workbench

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

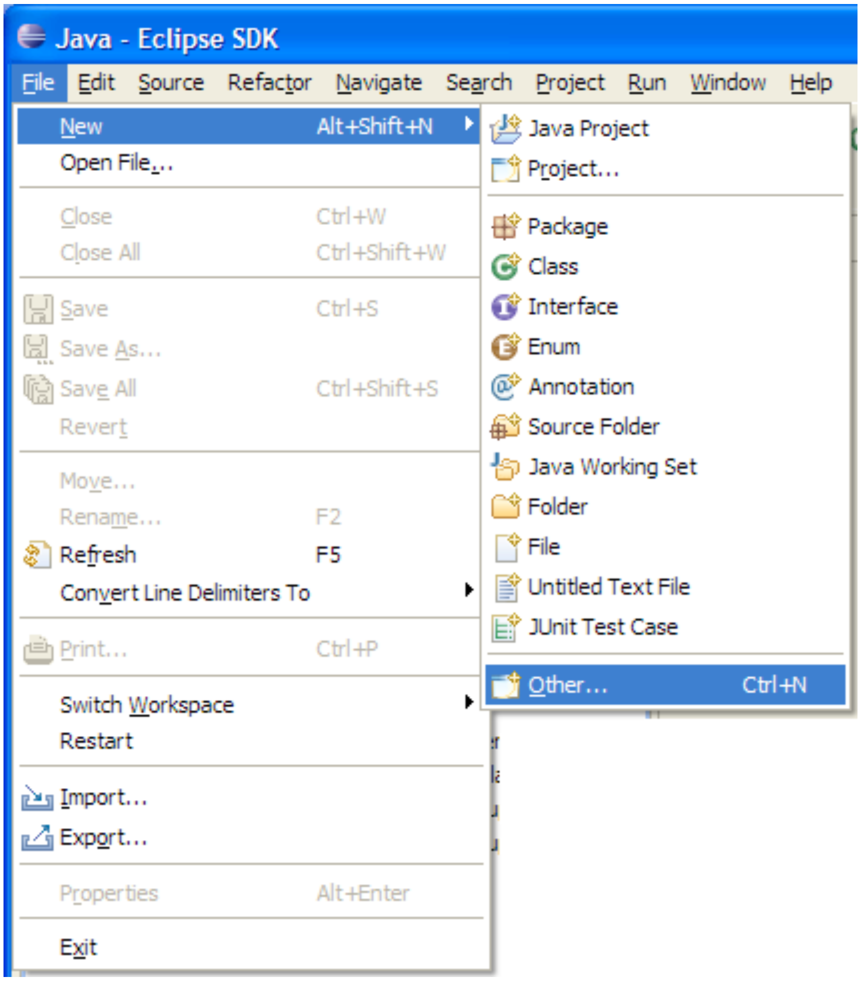
In Eclipse the plug-ins can be packaged together as an Eclipse Feature. The edu.harvard.i2b2.eclipse.features.core plug-in is the baseline Eclipse Feature for the i2b2 Workbench. Included in this feature are the core i2b2 Views and required plug-ins to run the i2b2 Workbench. As optional views are developed 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.

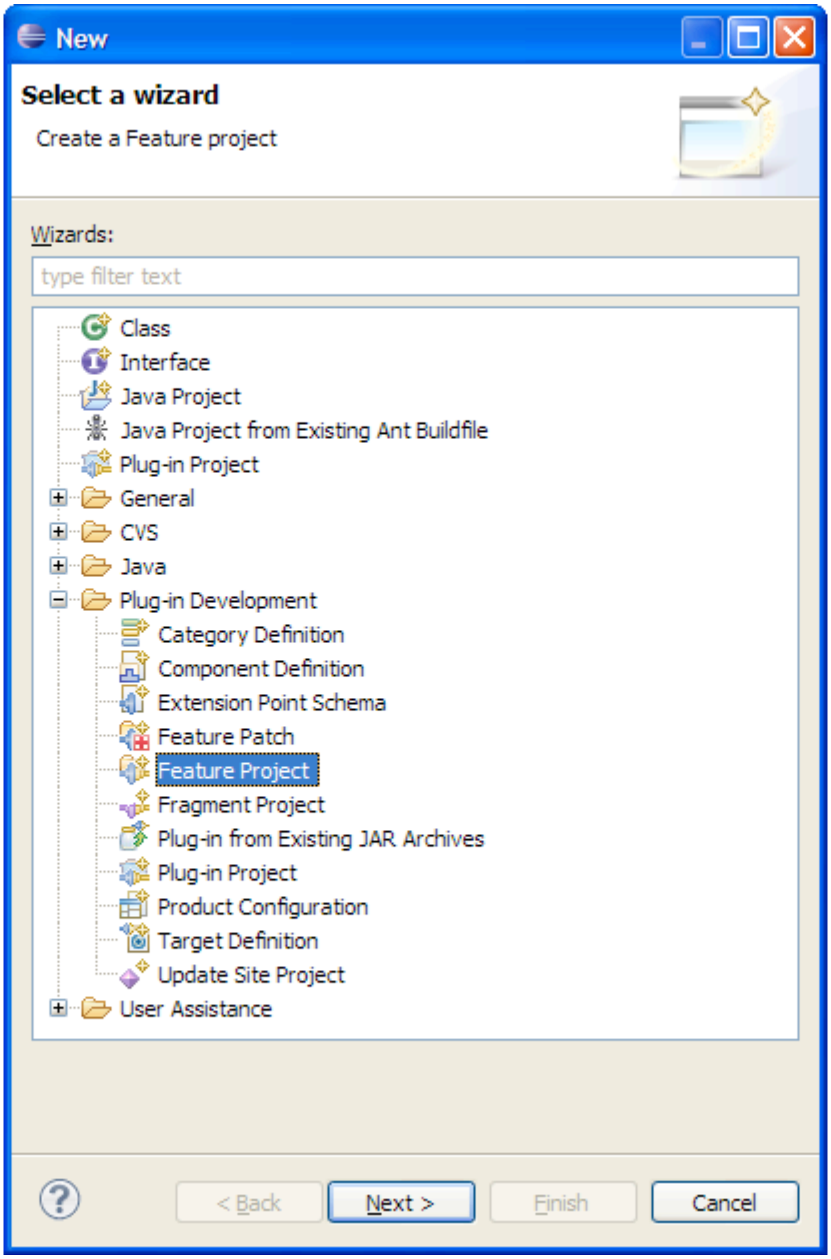
4.1 Creating a New "Feature" Project

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

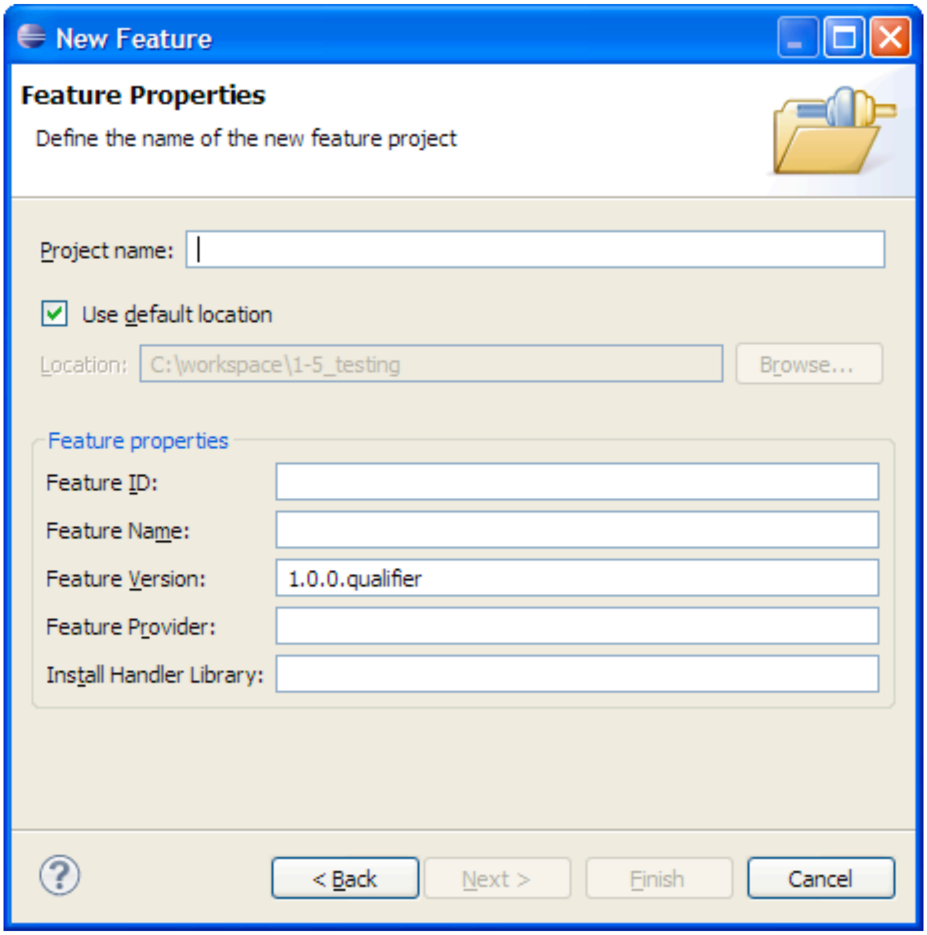
1. On the Eclipse menu bar click on **File**.
2. Select **New** from the drop down menu and **Other** from the sub-menu.



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.



Feature Properties

Define the name of the new feature project

☑ Use default location

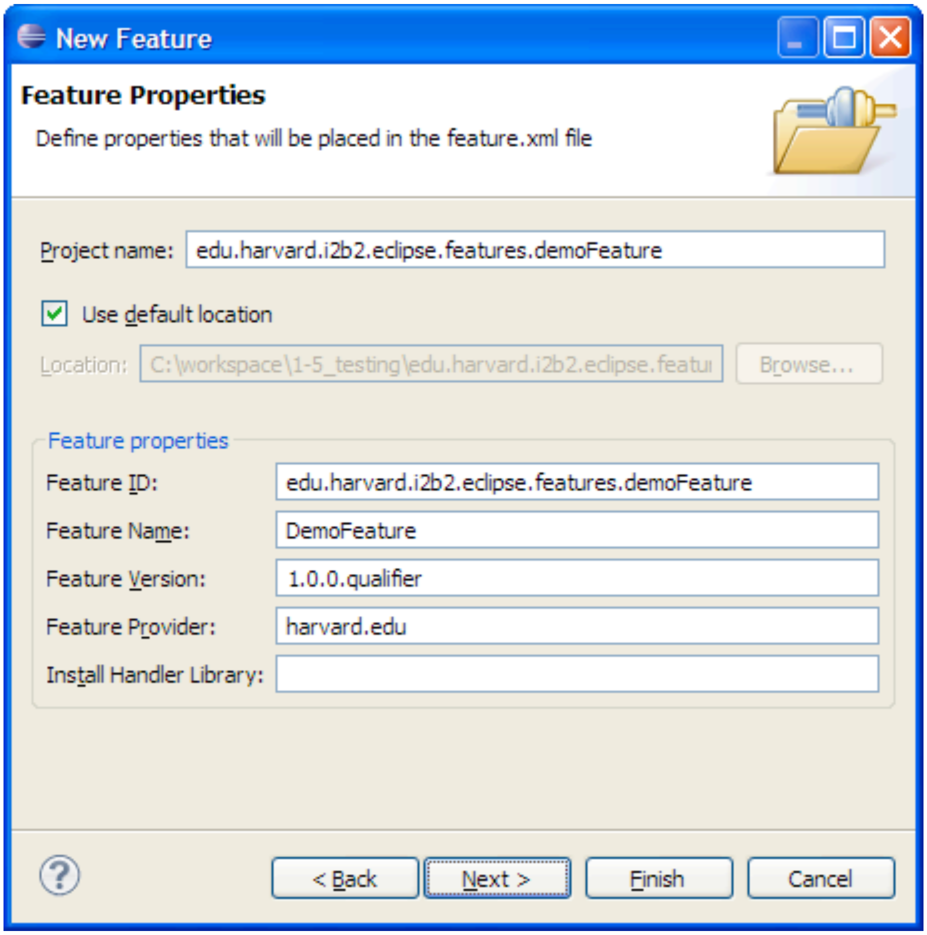
Location: C:\workspace\1-5_testing

Feature Version: 1.0.0.qualifier

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.



Feature Properties

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

Project name: edu.harvard.i2b2.eclipse.features.demoFeature

☑ Use default location

Location: C:\workspace\1-5_testing\edu.harvard.i2b2.eclipse.featu

Feature properties

Feature ID: edu.harvard.i2b2.eclipse.features.demoFeature

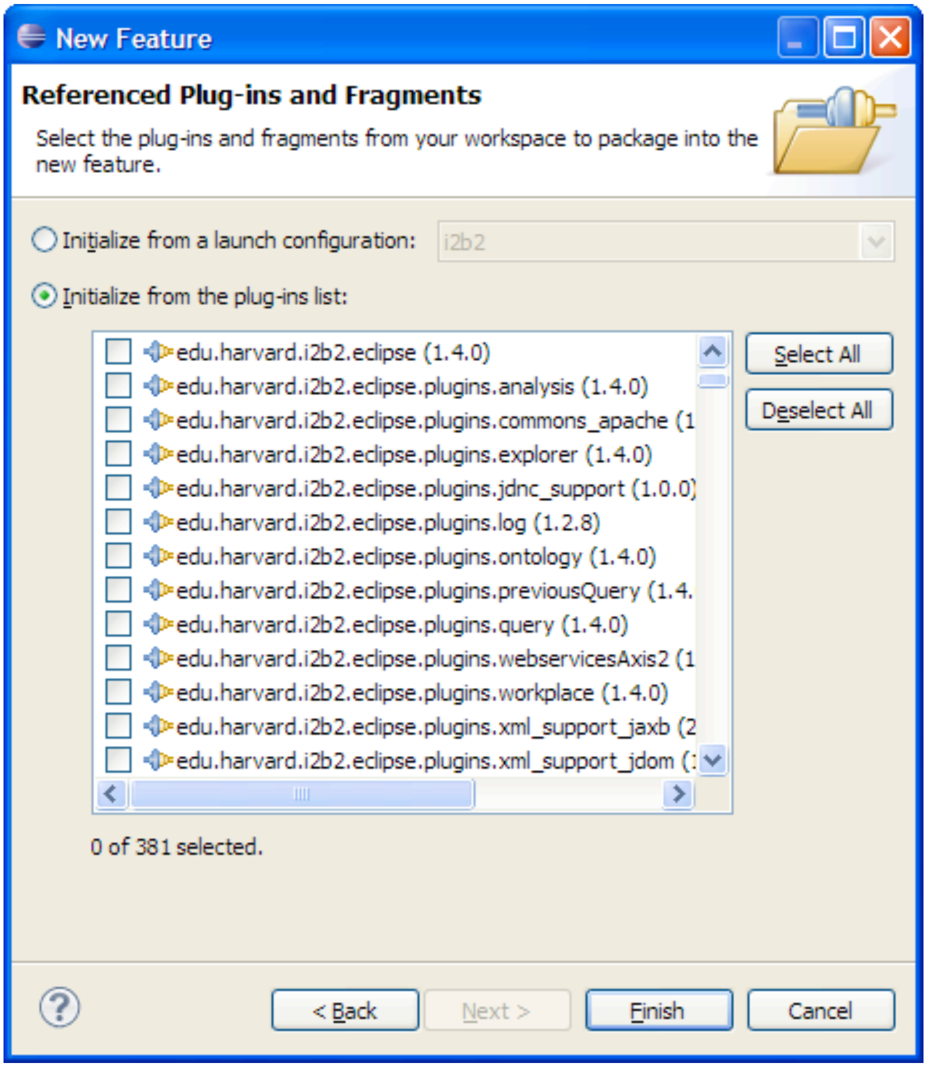
Feature Name: DemoFeature

Feature Version: 1.0.0.qualifier

Feature Provider: harvard.edu

Install Handler Library:

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.



Referenced Plug-ins and Fragments

Select the plug-ins and fragments from your workspace to package into the new feature.

○ Initialize from a launch configuration: i2b2

● Initialize from the plug-ins list:

- ☐ edu.harvard.i2b2.eclipse (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.analysis (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.commons_apache (1
- ☐ edu.harvard.i2b2.eclipse.plugins.explorer (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.jdnc_support (1.0.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.log (1.2.8)
- ☐ edu.harvard.i2b2.eclipse.plugins.ontology (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.previousQuery (1.4.
- ☐ edu.harvard.i2b2.eclipse.plugins.query (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.webservicesAxis2 (1
- ☐ edu.harvard.i2b2.eclipse.plugins.workplace (1.4.0)
- ☐ edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb (2
- ☐ edu.harvard.i2b2.eclipse.plugins.xml_support_jdom (

0 of 381 selected.

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

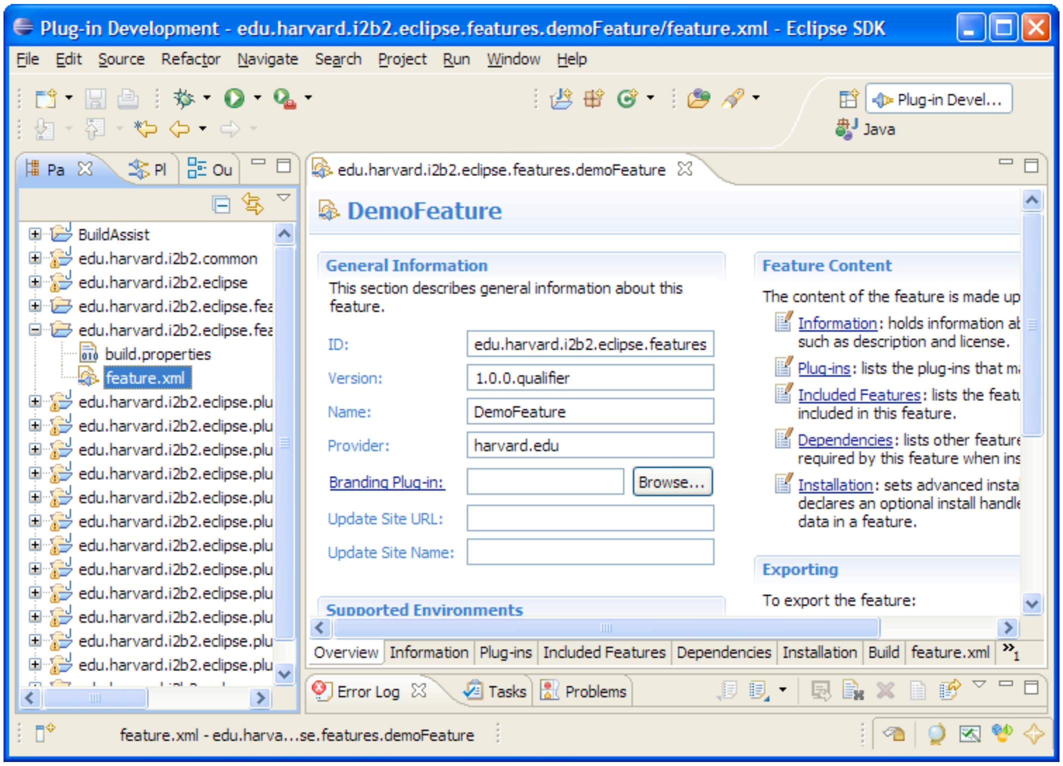
4.2 Feature Manifest (feature.xml)

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.


Plug-in Development - edu.harvard.i2b2.eclipse.features.demoFeature/feature.xml - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run Window Help
Plug-in Devel...
Java
Pa
Pl
Ou
BuildAssist
edu.harvard.i2b2.common
edu.harvard.i2b2.eclipse
edu.harvard.i2b2.eclipse.fea
edu.harvard.i2b2.eclipse.fea
build.properties
feature.xml
edu.harvard.i2b2.eclipse.plu
edu.harvard.i2b2.eclipse.feature
DemoFeature
General Information
Feature Content
Browse...
Exporting
Supported Environments
Overview Information Plug-ins Included Features Dependencies Installation Build feature.xml
Error Log Tasks Problems
feature.xml - edu.harva...se.features.demoFeature

This section describes general information about this feature.

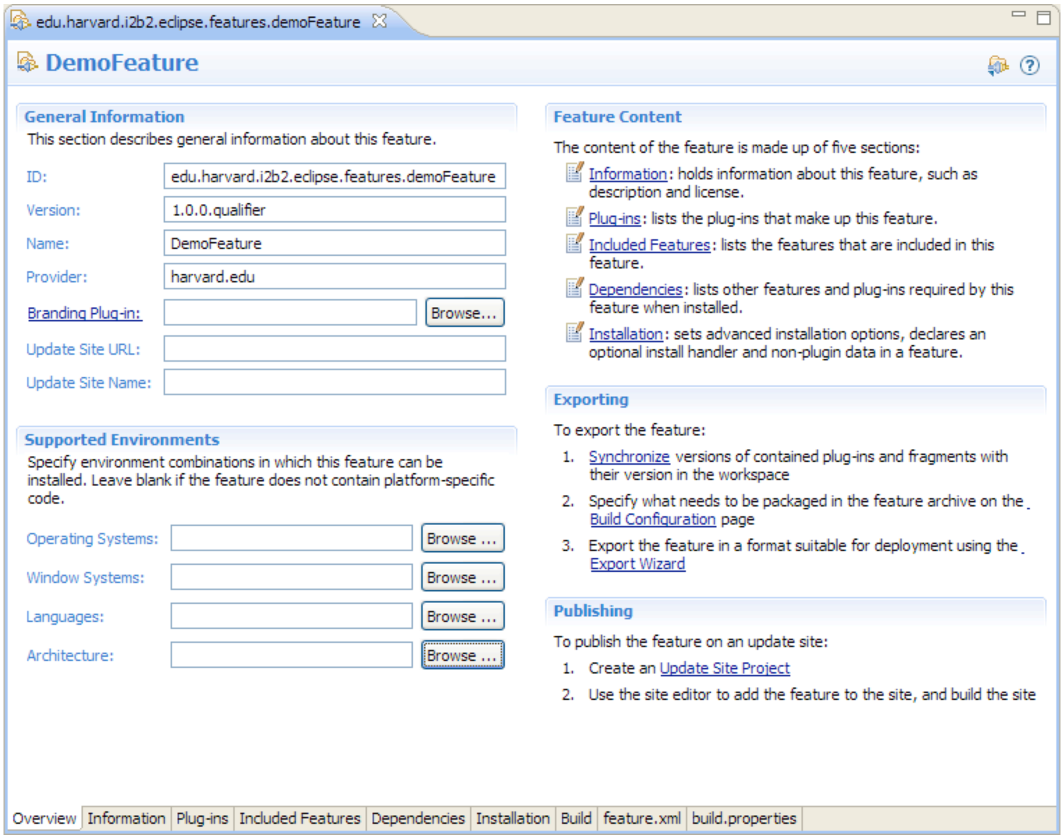
ID: edu.harvard.i2b2.eclipse.features
Version: 1.0.0.qualifier
Name: DemoFeature
Provider: harvard.edu
Branding Plug-in:
Update Site URL:
Update Site Name:

The content of the feature is made up
Information: holds information al such as description and license.
Plug-ins: lists the plug-ins that m
Included Features: lists the featu included in this feature.
Dependencies: lists other feature required by this feature when ins
Installation: sets advanced insta declares an optional install handl data in a feature.

To export the feature:

4.2.2 Overview Tab

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



DemoFeature

General Information

This section describes general information about this feature.

ID: edu.harvard.i2b2.eclipse.features.demoFeature

Version: 1.0.0.qualifier

Name: DemoFeature

Provider: harvard.edu

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:

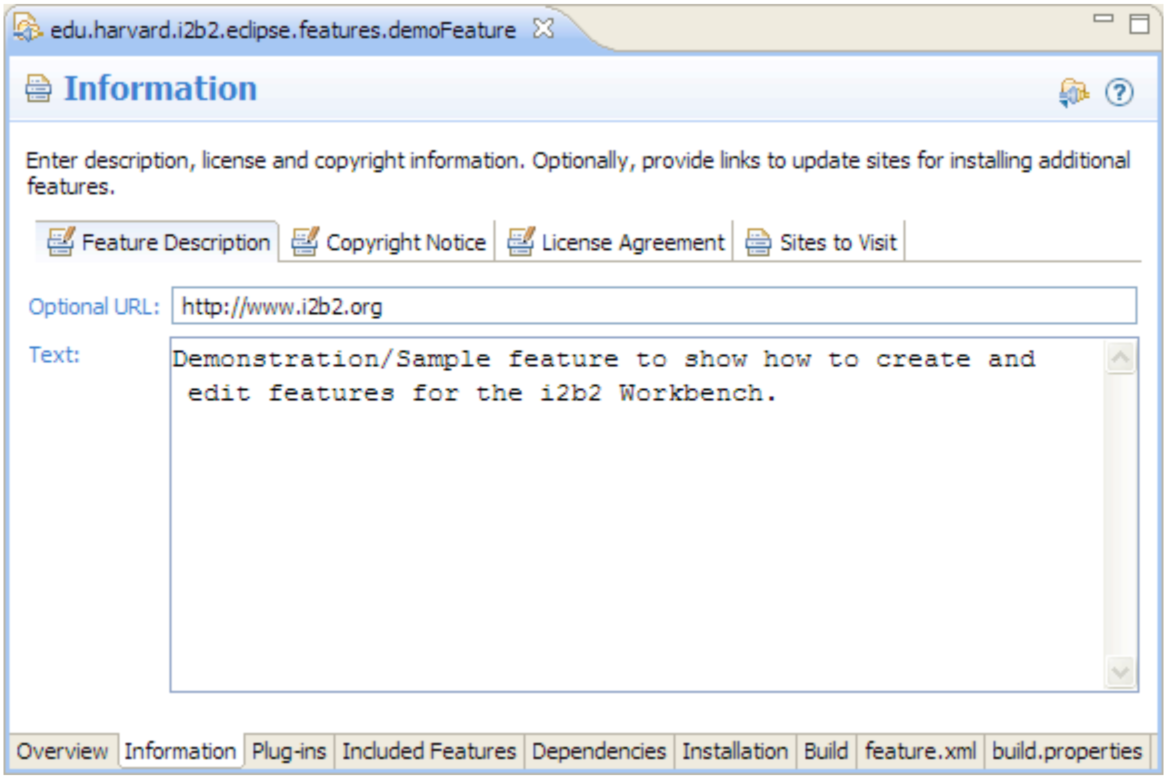
1. Create an Update Site Project
2. Use the site editor to add the feature to the site, and build the site

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.



Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

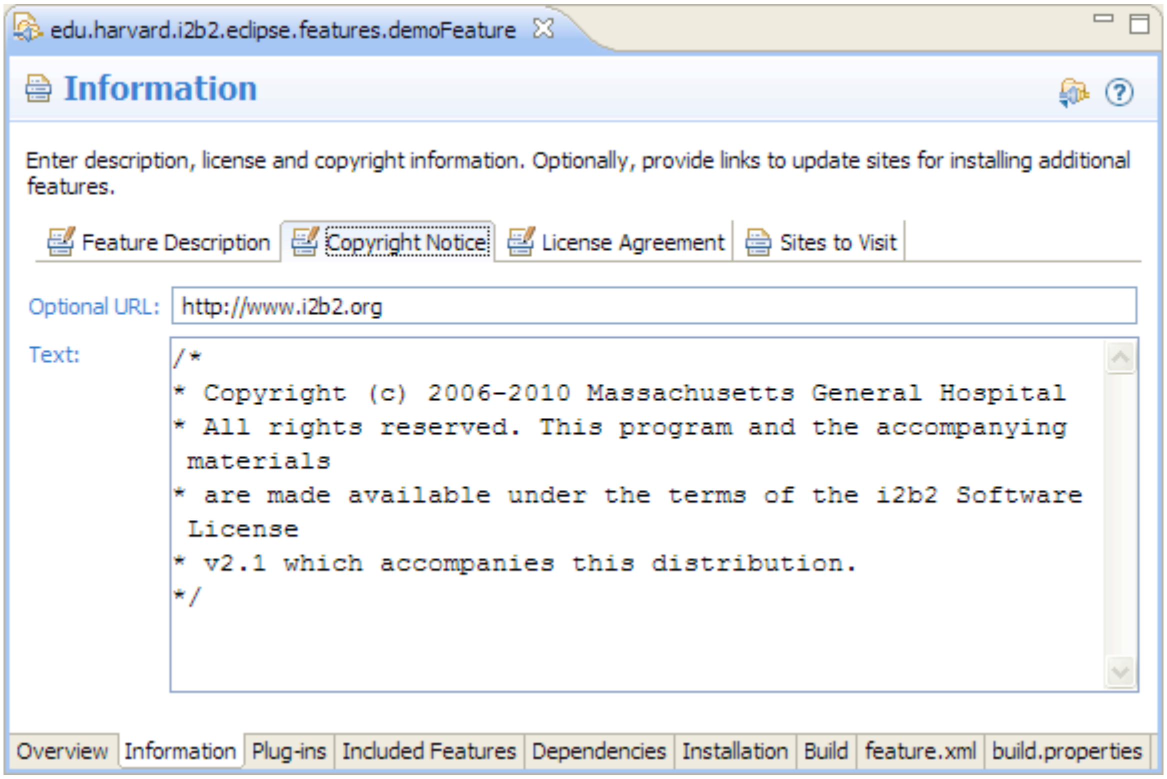
Optional URL: http://www.i2b2.org

Text:

```
Demonstration/Sample feature to show how to create and
 edit features for the i2b2 Workbench.
```

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.



Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

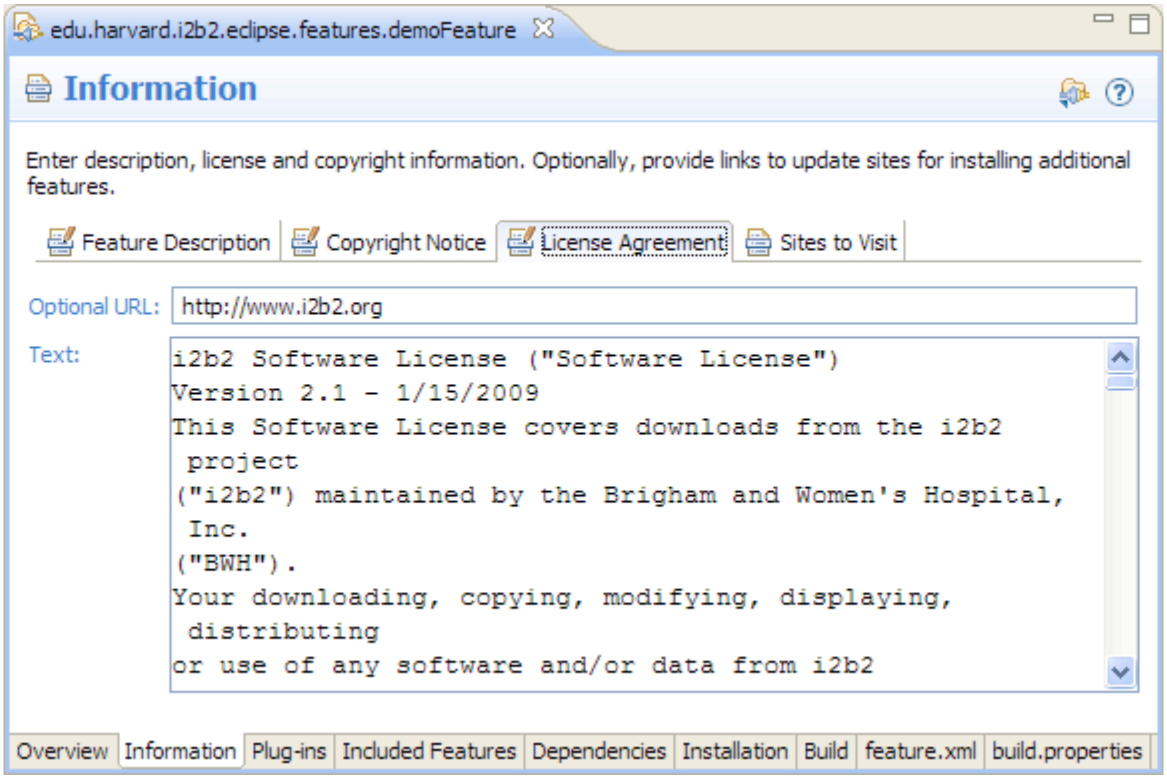
Optional URL: http://www.i2b2.org

Text:

```
/*
* Copyright (c) 2006-2010 Massachusetts General Hospital
* All rights reserved. This program and the accompanying
 materials
* are made available under the terms of the i2b2 Software
 License
* v2.1 which accompanies this distribution.
*/
```

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.



Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

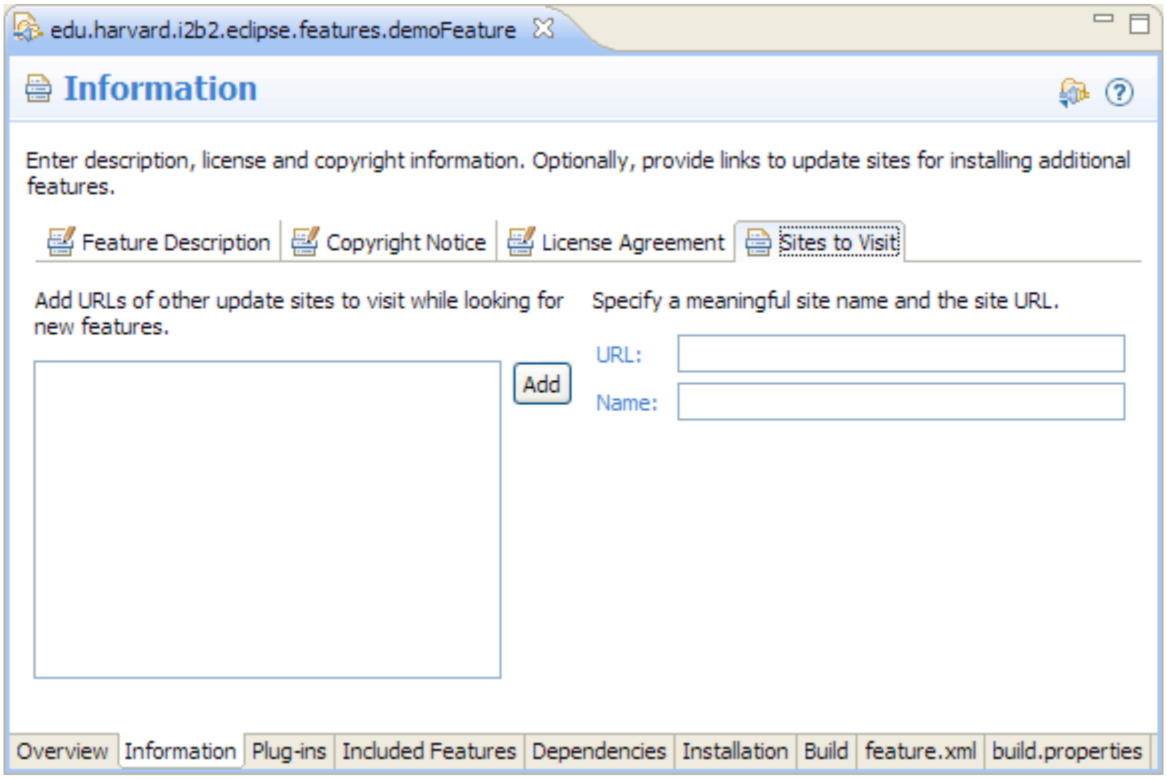
Optional URL: http://www.i2b2.org

Text:

```
i2b2 Software License ("Software License")
Version 2.1 - 1/15/2009
This Software License covers downloads from the i2b2
 project
("i2b2") maintained by the Brigham and Women's Hospital,
 Inc.
("BWH").
Your downloading, copying, modifying, displaying,
 distributing
or use of any software and/or data from i2b2
```

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.



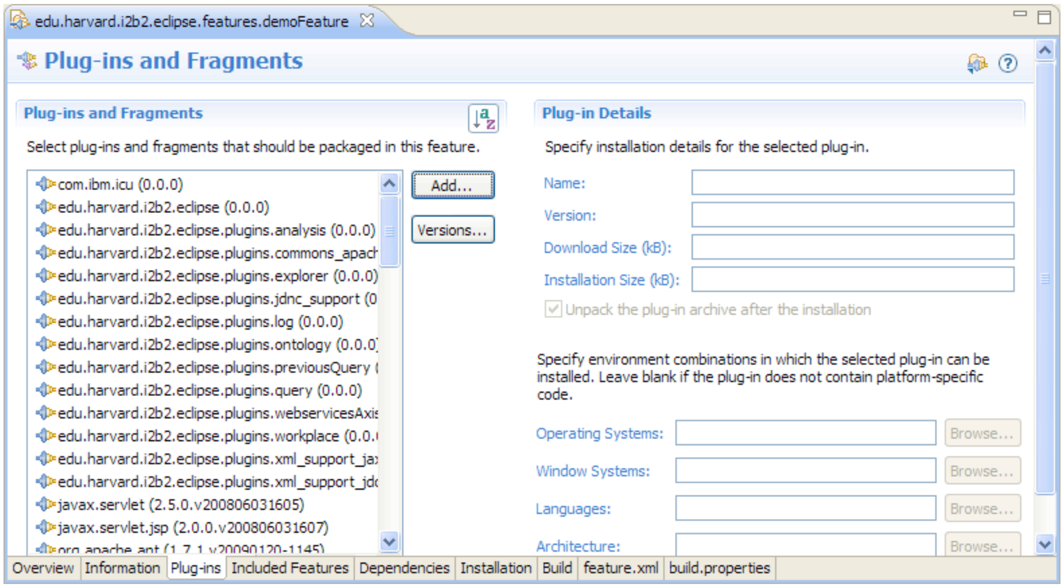
Enter description, license and copyright information. Optionally, provide links to update sites for installing additional features.

Add URLs of other update sites to visit while looking for new features.

Specify a meaningful site name and the site URL.

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.



Select plug-ins and fragments that should be packaged in this feature.

com.ibm.icu (0.0.0)
edu.harvard.i2b2.eclipse (0.0.0)
edu.harvard.i2b2.eclipse.plugins.analysis (0.0.0)
edu.harvard.i2b2.eclipse.plugins.commons_apach
edu.harvard.i2b2.eclipse.plugins.explorer (0.0.0)
edu.harvard.i2b2.eclipse.plugins.jdnc_support (0
edu.harvard.i2b2.eclipse.plugins.log (0.0.0)
edu.harvard.i2b2.eclipse.plugins.ontology (0.0.0)
edu.harvard.i2b2.eclipse.plugins.previousQuery (
edu.harvard.i2b2.eclipse.plugins.query (0.0.0)
edu.harvard.i2b2.eclipse.plugins.webservicesAxis
edu.harvard.i2b2.eclipse.plugins.workplace (0.0.
edu.harvard.i2b2.eclipse.plugins.xml_support_ja
edu.harvard.i2b2.eclipse.plugins.xml_support_jd
javax.servlet (2.5.0.v200806031605)
javax.servlet.jsp (2.0.0.v200806031607)
org.apache.ant (1.7.1.v20090120-1145)

Specify installation details for the selected plug-in.

☑ Unpack the plug-in archive after the installation

Specify environment combinations in which the selected plug-in can be installed. Leave blank if the plug-in does not contain platform-specific code.

4.2.4.1 ADDING AND REMOVING PLUG-INS AND FRAGMENTS

Add Plug-in/Fragment

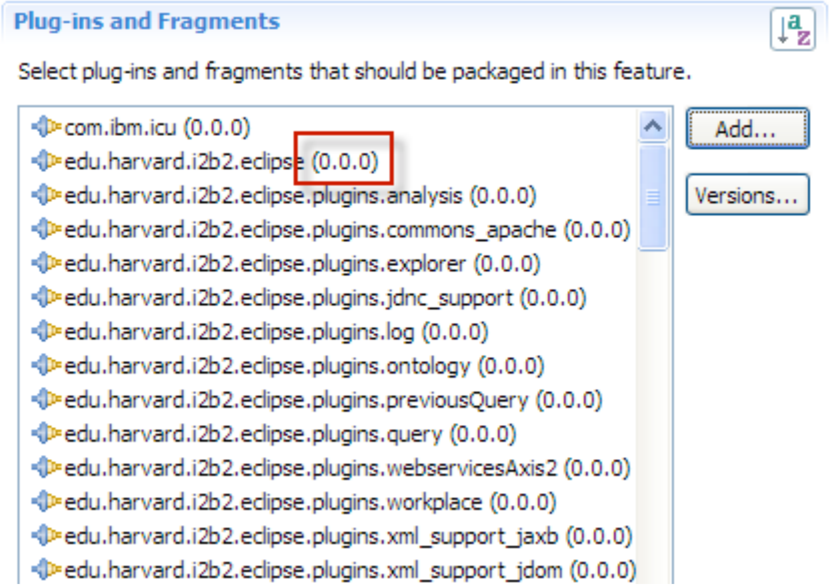
1. Click on the **Add** button and the **Plug-in Selection** dialog box will open.
2. Select the plug-in and/or fragment you want to add.
3. Click on the **OK** button.

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

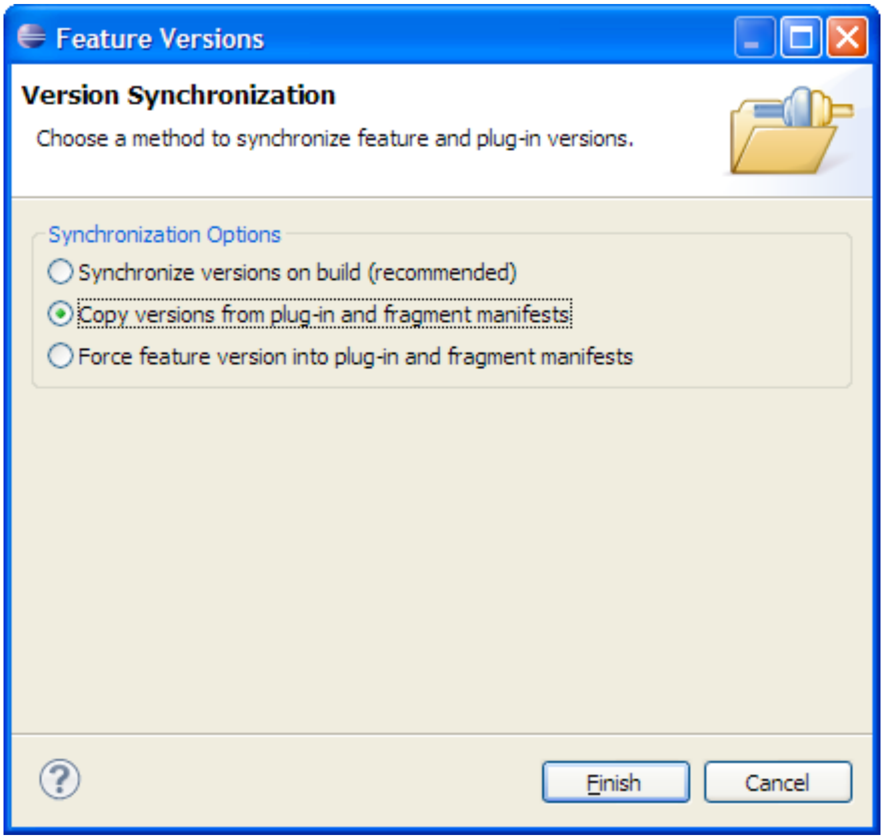
When the feature is first created the version of the plug-ins/fragments to be included is set to 0.0.0. Therefore, it is important that you update them to reflect the correct version information.



Select plug-ins and fragments that should be packaged in this feature.

- com.ibm.icu (0.0.0)
- edu.harvard.i2b2.eclipse (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.analysis (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.commons_apache (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.explorer (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.jdnc_support (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.log (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.ontology (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.previousQuery (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.query (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.webservicesAxis2 (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.workplace (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb (0.0.0)
- edu.harvard.i2b2.eclipse.plugins.xml_support_jdom (0.0.0)

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



Version Synchronization

Choose a method to synchronize feature and plug-in versions.

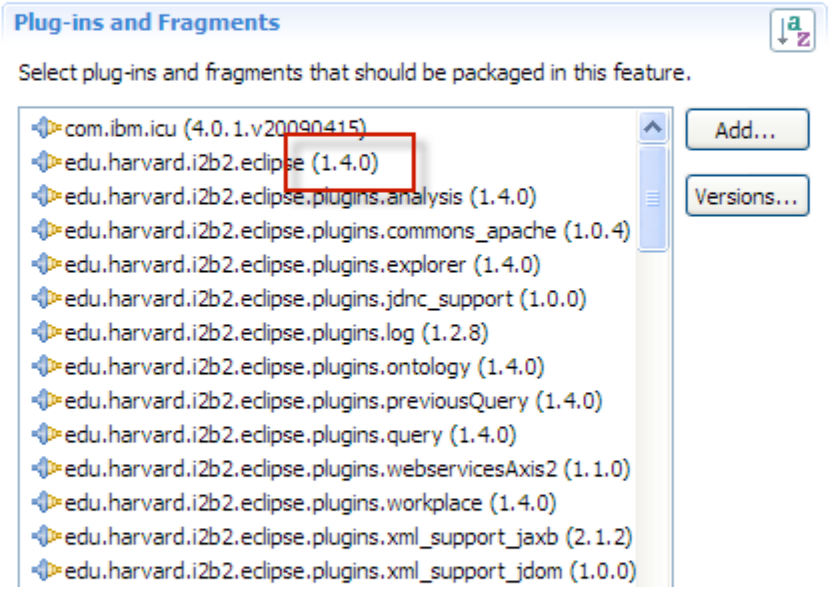
Synchronization Options

○ Synchronize versions on build (recommended)

● Copy versions from plug-in and fragment manifests

○ Force feature version into plug-in and fragment manifests

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.



Plug-ins and Fragments

Select plug-ins and fragments that should be packaged in this feature.

com.ibm.icu (4.0.1.v20090415)
edu.harvard.i2b2.eclipse (1.4.0)
edu.harvard.i2b2.eclipse.plugins.analysis (1.4.0)
edu.harvard.i2b2.eclipse.plugins.commons_apache (1.0.4)
edu.harvard.i2b2.eclipse.plugins.explorer (1.4.0)
edu.harvard.i2b2.eclipse.plugins.jdnc_support (1.0.0)
edu.harvard.i2b2.eclipse.plugins.log (1.2.8)
edu.harvard.i2b2.eclipse.plugins.ontology (1.4.0)
edu.harvard.i2b2.eclipse.plugins.previousQuery (1.4.0)
edu.harvard.i2b2.eclipse.plugins.query (1.4.0)
edu.harvard.i2b2.eclipse.plugins.webservicesAxis2 (1.1.0)
edu.harvard.i2b2.eclipse.plugins.workplace (1.4.0)
edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb (2.1.2)
edu.harvard.i2b2.eclipse.plugins.xml_support_jdom (1.0.0)

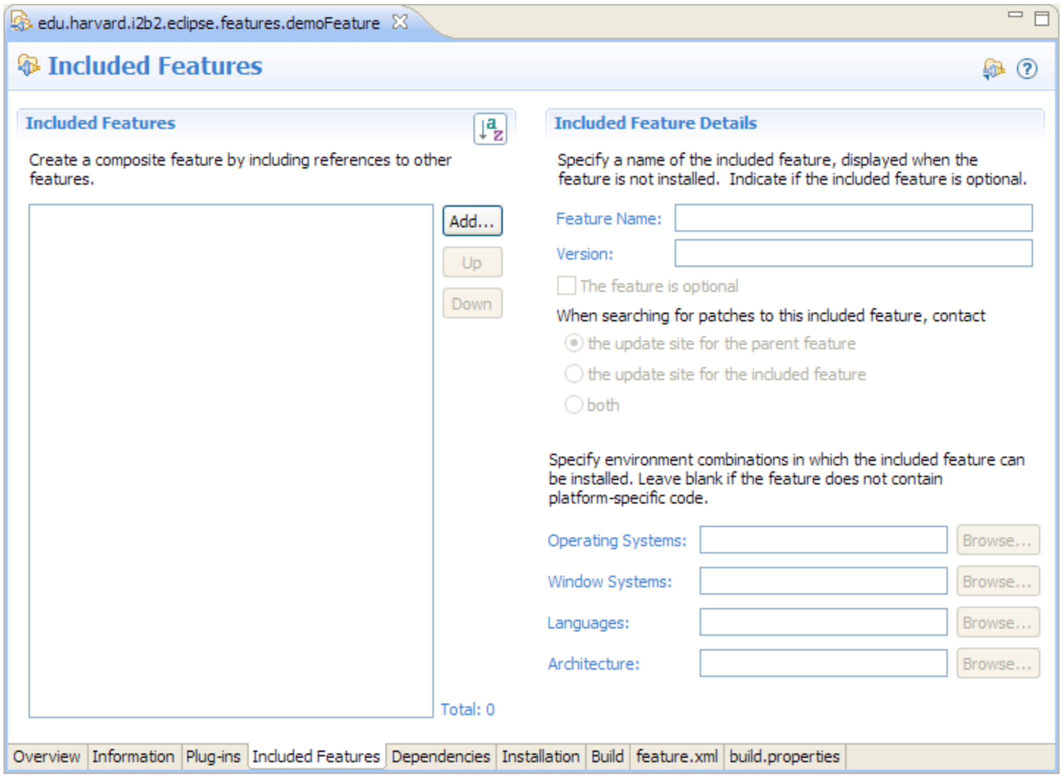
4.2.4.3 UPDATE PLUG-IN DETAILS

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.



Create a composite feature by including references to other features.

Specify a name of the included feature, displayed when the feature is not installed. Indicate if the included feature is optional.

When searching for patches to this included feature, contact

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

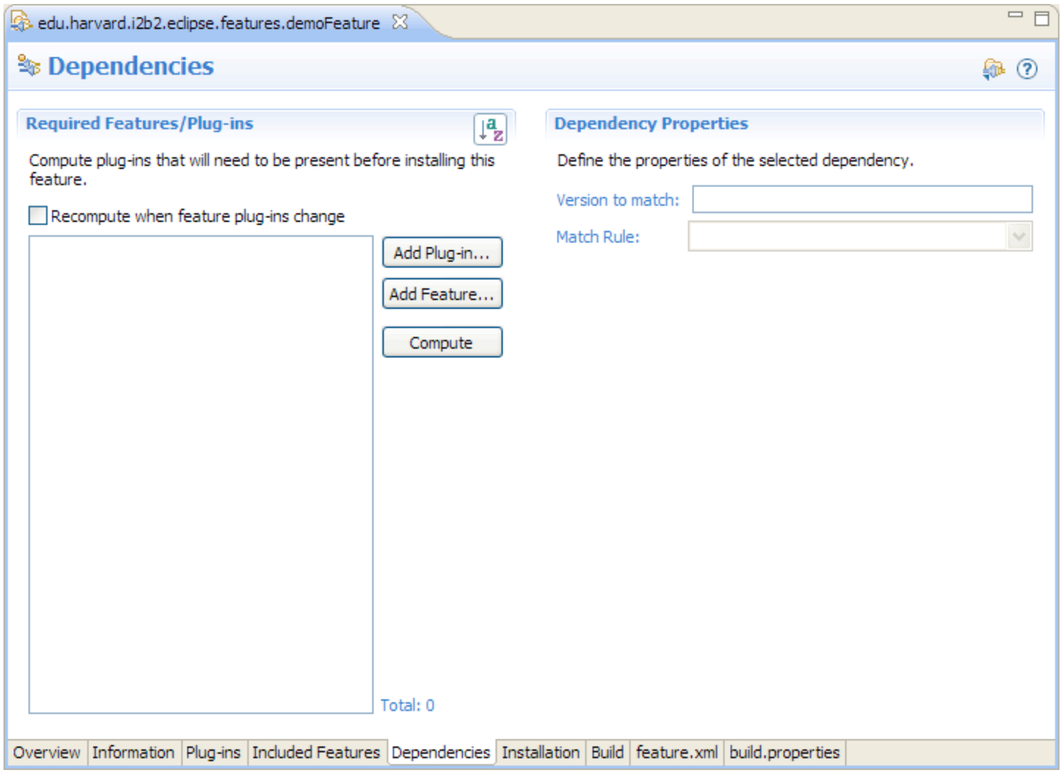
4.2.6 Dependencies Tab

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.

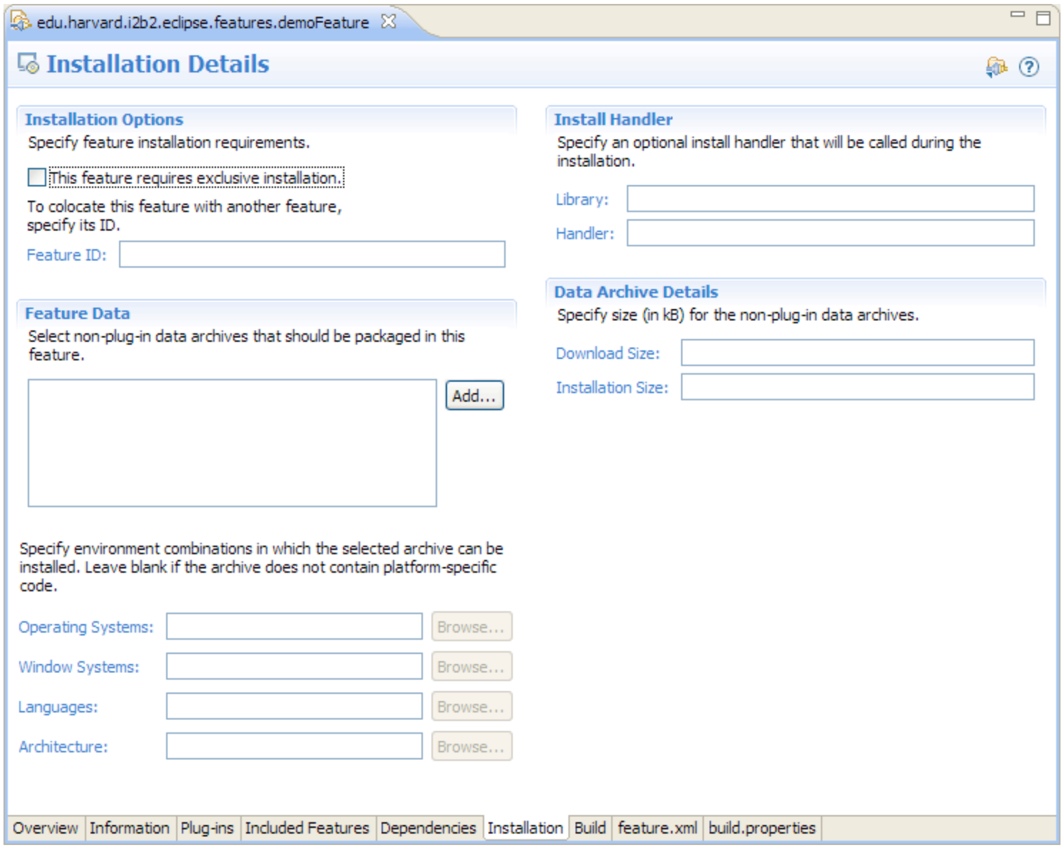

edu.harvard.i2b2.eclipse.features.demoFeature
Dependencies
Required Features/Plug-ins
Compute plug-ins that will need to be present before installing this feature.
☐ Recompute when feature plug-ins change
Add Plug-in...
Add Feature...
Compute
Total: 0
Dependency Properties
Define the properties of the selected dependency.
Version to match:
Match Rule:
Overview
Information
Plug-ins
Included Features
Dependencies
Installation
Build
feature.xml
build.properties

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.

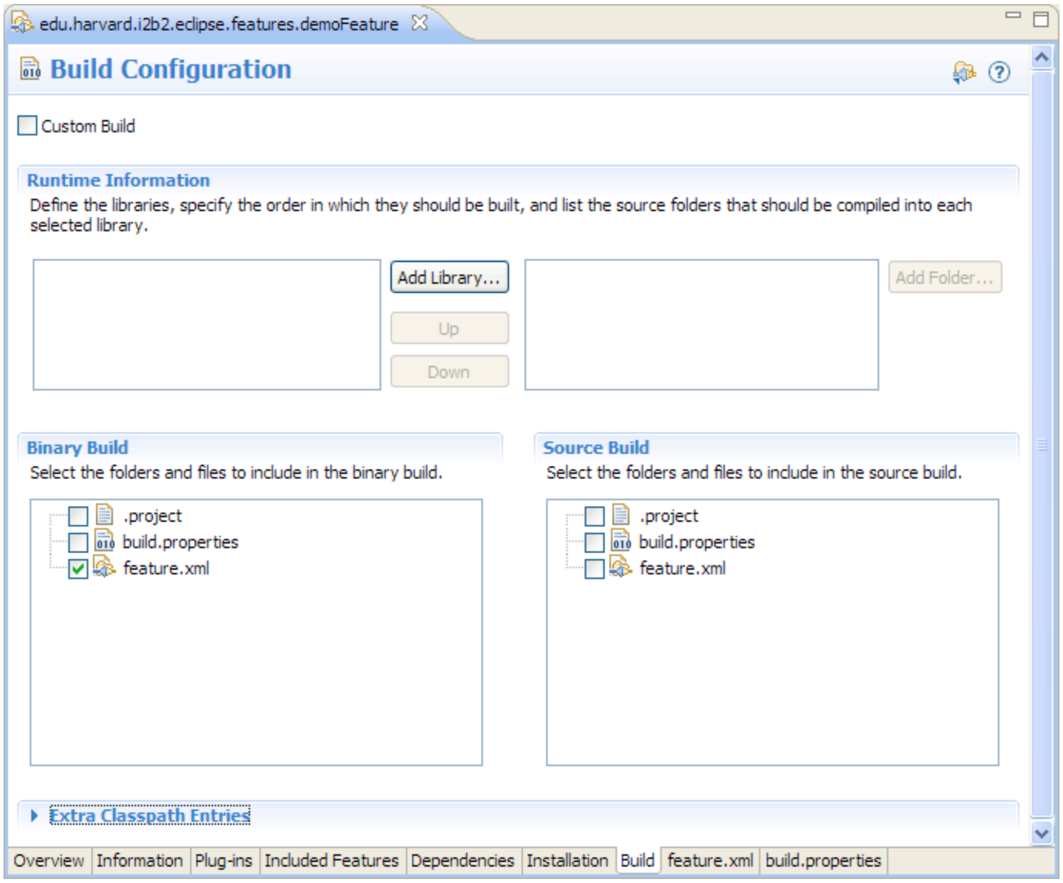
4.2.7 Installation Tab

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


edu.harvard.i2b2.eclipse.features.demoFeature
Installation Details
Installation Options
Specify feature installation requirements.
☐ This feature requires exclusive installation.
To colocate this feature with another feature, specify its ID.
Feature ID:
Feature Data
Select non-plug-in data archives that should be packaged in this feature.
Add...
Specify environment combinations in which the selected archive can be installed. Leave blank if the archive does not contain platform-specific code.
Operating Systems:
Browse...
Window Systems:
Browse...
Languages:
Browse...
Architecture:
Browse...
Install Handler
Specify an optional install handler that will be called during the installation.
Library:
Handler:
Data Archive Details
Specify size (in kB) for the non-plug-in data archives.
Download Size:
Installation Size:
Overview
Information
Plug-ins
Included Features
Dependencies
Installation
Build
feature.xml
build.properties

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.



☐ Custom Build

Runtime Information

Define the libraries, specify the order in which they should be built, and list the source folders that should be compiled into each selected library.

Binary Build

Select the folders and files to include in the binary build.

☐ .project
☐ build.properties
☑ feature.xml

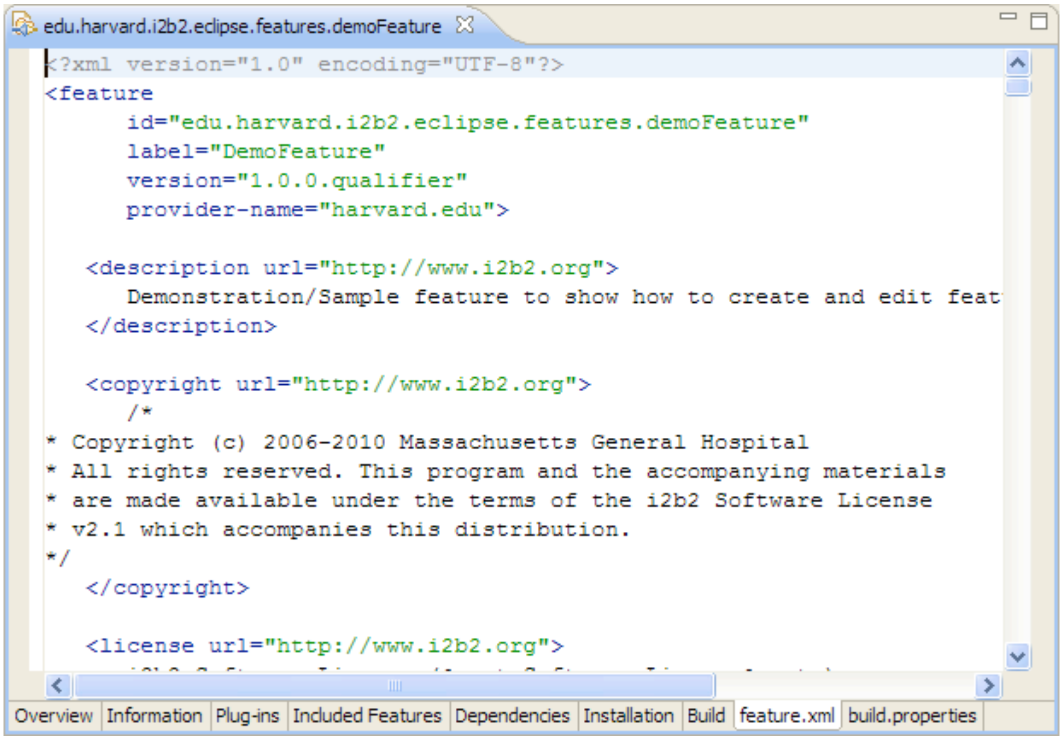
Source Build

Select the folders and files to include in the source build.

☐ .project
☐ build.properties
☐ feature.xml

4.2.9 Feature.xml Tab

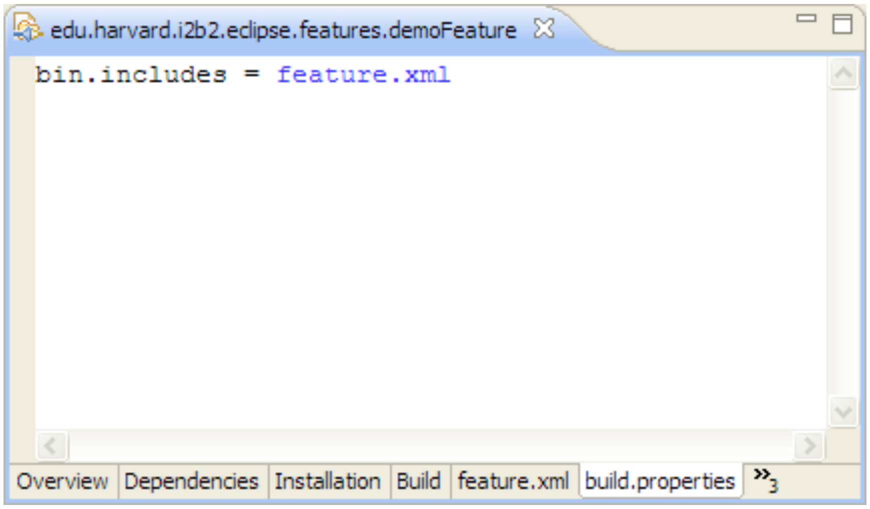
The **feature.xml** page is the XML for all the information defined on the other pages.



```
<?xml version="1.0" encoding="UTF-8"?>
<feature
      id="edu.harvard.i2b2.eclipse.features.demoFeature"
      label="DemoFeature"
      version="1.0.0.qualifier"
      provider-name="harvard.edu">

   <description url="http://www.i2b2.org">
      Demonstration/Sample feature to show how to create and edit feat
   </description>

   <copyright url="http://www.i2b2.org">
      /*
* Copyright (c) 2006-2010 Massachusetts General Hospital
* All rights reserved. This program and the accompanying materials
* are made available under the terms of the i2b2 Software License
* v2.1 which accompanies this distribution.
*/
   </copyright>

   <license url="http://www.i2b2.org">
```

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.



```
bin.includes = feature.xml
```

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.