

i2b2 Workbench



Developer's Guide: Eclipse Neon & i2b2 Source Code

About this guide

Informatics for Integrating Biology and the Bedside (i2b2) began as 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 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.

This guide is designed to provide an overview of installing and configuring the i2b2 Workbench within the Eclipse IDE.

Operating Systems

The i2b2 Workbench does run on Microsoft Windows (“Windows”) and Apple’s Macintosh (“Mac”) operating systems. However, there is a problem with the i2b2 Workbench not running correctly on newer apple machines. Since the problem is with a third-party software and out of the hands of the i2b2 community we have decided to create two new virtual machines (VM) that mac users can download and use in their environment.

The Workbench VM Guide explains how to use the VM and can be used in combination with this guide to understand how to use the i2b2 Workbench source code within Eclipse.

Document management

Revision Number	Date	Author	Description of Change
2.0.01-0001	10/01/2017	Janice Donahoe	Created initial version of the developer's guide for the i2b2 Workbench 2.0.01 Software.

2.0.01-0002	10/17/2017	Janice Donahoe	Fixed link to Software Release page.
-------------	------------	----------------	--------------------------------------

Prerequisites

There are a few requirements that need to be met before you can continue with the developer's guide for the i2b2 Workbench. Mainly, you need to make sure the third-party software listed below has been downloaded and installed on the same machine that you will be running the i2b2 source code on. You will also need to download the

Third-Party Software

There are a few requirements that need to be met before you can continue with the developer's guide for the i2b2 Workbench. The next couple of sections outline these requirements.

Java JDK

There are two things to know about the i2b2 Workbench and Java JDK / JRE:

1. The **i2b2 Workbench Application / Executable** requires **JRE** to be *installed on the same machine*. If it is missing then a user will receive an error when trying to log into the i2b2 Workbench.
2. To **build** your Workbench **plugins** in Eclipse, **JDK / JRE** must be *installed on the same machine* as your **Eclipse application**. Otherwise you will receive errors and will not be able export the workbench application or build custom JAR files.

i2b2 Resource: Download & Install Java JDK

[Chapter 2.4.1 Java JDK](#) of the i2b2 Installation Guide provides links to download JDK for Linux or Windows and how to install it in your environment.

Eclipse

The i2b2 Workbench is a series of components that are designed as an Eclipse-based plugin and therefore the Eclipse SDK is an integral piece of software when developing new plugins for the i2b2 Workbench.

The latest version of the i2b2 Workbench (2.0.01) has been upgraded to work with Eclipse (Neon). The second prerequisite is to download and install version 4.x of the **Eclipse SDK (Neon)** from the Eclipse website by going to: <https://www.eclipse.org/neon/>

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

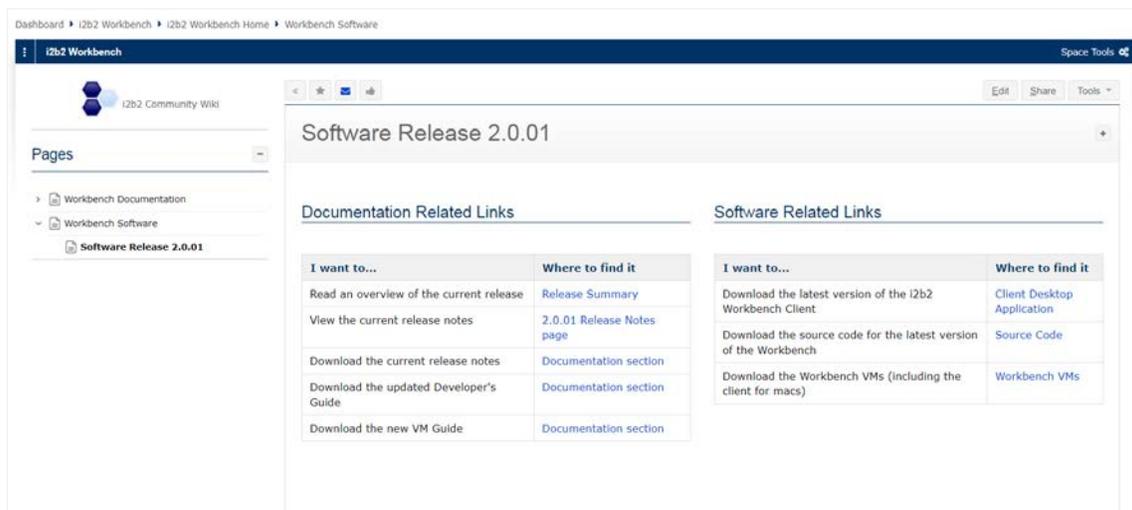
i2b2 Software

i2b2 Workbench Source Code

The latest version of the **source code** for the i2b2 Workbench needs to be downloaded. For your convenience, we have created a zip file that contains all the i2b2 Workbench plugins plus the required xml plugin and have made it available on the i2b2 Community Wiki. Here are some basic steps you can take to access this file and to download it to your local environment.

1. Go to **Software Release 2.0.01** page in the **Workbench Space** on the i2b2 Community Wiki.

(<https://community.i2b2.org/wiki/display/WB/Software+Release+2.0.01>)



The screenshot shows the i2b2 Community Wiki page for 'Software Release 2.0.01'. The page layout includes a navigation sidebar on the left with a 'Pages' section containing 'Workbench Documentation', 'Workbench Software', and 'Software Release 2.0.01'. The main content area is titled 'Software Release 2.0.01' and features two tables of links:

Documentation Related Links	
I want to...	Where to find it
Read an overview of the current release	Release Summary
View the current release notes	2.0.01 Release Notes page
Download the current release notes	Documentation section
Download the updated Developer's Guide	Documentation section
Download the new VM Guide	Documentation section

Software Related Links	
I want to...	Where to find it
Download the latest version of the i2b2 Workbench Client	Client Desktop Application
Download the source code for the latest version of the Workbench	Source Code
Download the Workbench VMs (including the client for macs)	Workbench VMs

2. Scroll to the **Source Code** section

Description	Software Version	Download Link	Additional Documentation
i2b2 Workbench Source	2.0.01	 i2b2WB-src-2001.zip	

3. In the **Download Link column**, click on the **name of the file** you want to download.
4. The download process will begin.
5. If prompted, answer any questions regarding the i2b2 license to continue.

Install

i2b2 Workbench Project Files

The project software is contained in the Eclipse “Archive” file called **i2b2Workbench-src-xxx.zip**, which you downloaded in the previous section.

The above file name has been modified slightly in that the numbers that would normally appear have been replaced with an ‘x’. In actuality, these ‘x’s would be the version of software in the zip file. Example: i2b2 Workbench-src-2001.zip means the zip file contains version 2.0.01 of the i2b2 Workbench.

The zip for the i2b2 Workbench source code contains the following Eclipse projects:

1. BuildAssist
2. edu.harvard.i2b2.common
3. edu.harvard.i2b2.eclipse
4. edu.harvard.i2b2.eclipse.plugins.adminTool
5. edu.harvard.i2b2.eclipse.plugins.analysis
6. edu.harvard.i2b2.eclipse.plugins.common.apache
7. edu.harvard.i2b2.eclipse.plugins.crcxmljxb
8. edu.harvard.i2b2.eclipse.plugins.createTables

9. edu.harvard.i2b2.eclipse.plugins.explorer
10. edu.harvard.i2b2.eclipse.plugins.jdnc_support
11. edu.harvard.i2b2.eclipse.plugins.log
12. edu.harvard.i2b2.eclipse.plugins.ontology
13. edu.harvard.i2b2.eclipse.plugins.patientMapping
14. edu.harvard.i2b2.eclipse.plugins.patientSet
15. edu.harvard.i2b2.eclipse.plugins.previousQuery
16. edu.harvard.i2b2.eclipse.plugins.query
17. edu.harvard.i2b2.eclipse.plugins.querytool
18. edu.harvard.i2b2.eclipse.plugins.webservicesAxis2
19. edu.harvard.i2b2.eclipse.plugins.workplace
20. edu.harvard.i2b2.eclipse.plugins.xml_support_jaxb
21. edu.harvard.i2b2.eclipse.plugins.xml_support_jdom
22. edu.harvard.i2b2.xml

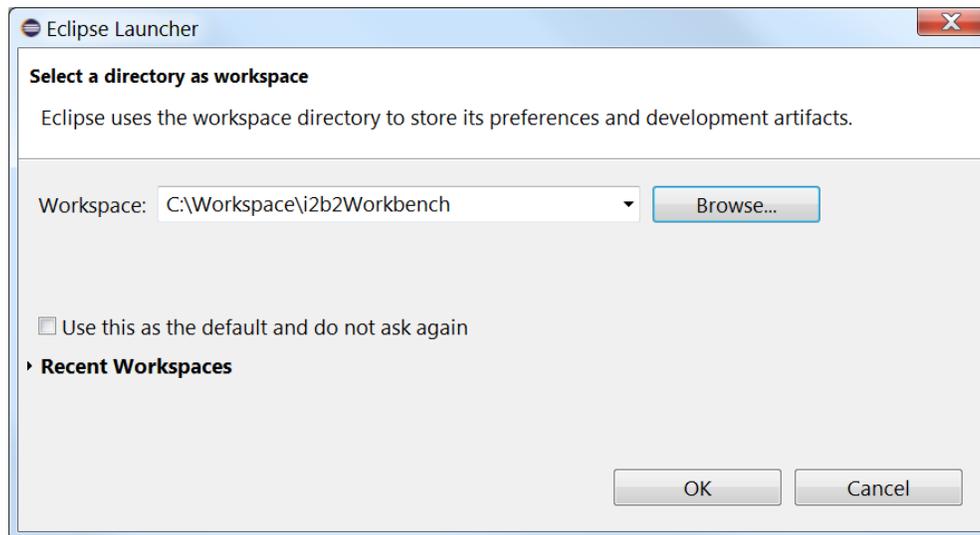
 **Note**

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

Setup the i2b2 Workbench within Eclipse

Starting Eclipse

1. Go to the location where you installed Eclipse
2. Run the **Eclipse application** by double clicking on  **eclipse.exe**
3. The **Eclipse Launcher window** will open.

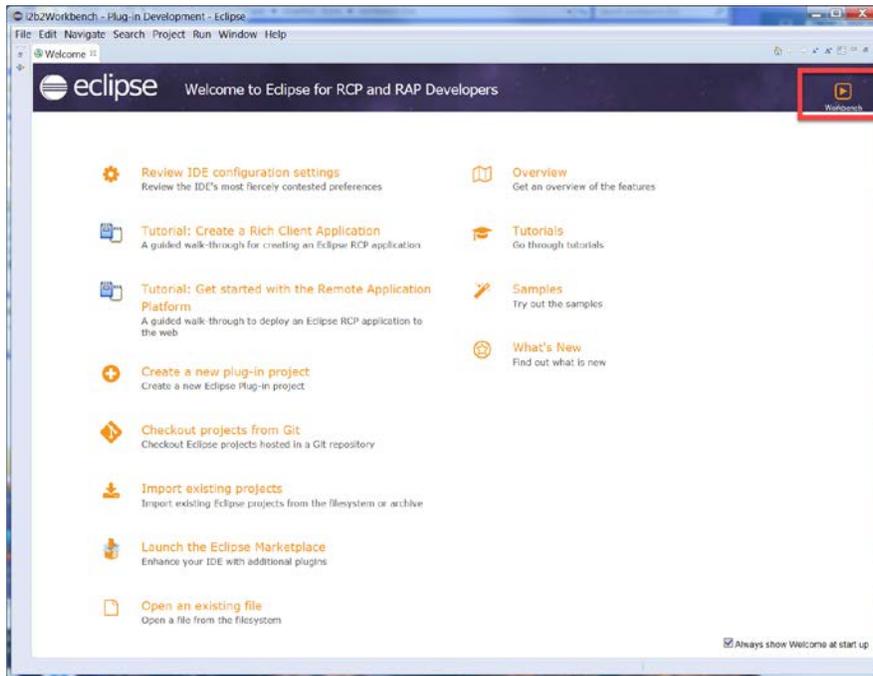


4. Select a workspace whose **pathname** *does not* contain spaces.

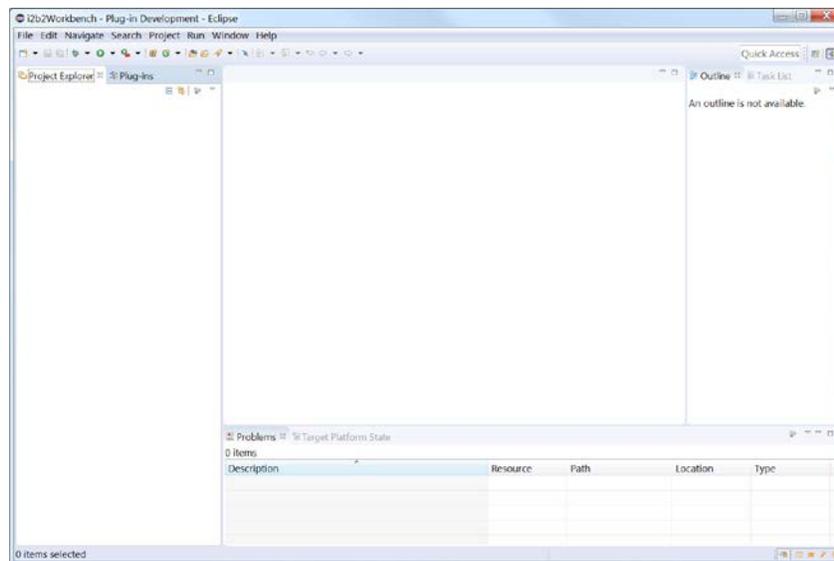
Example (correct): c:\workspace\i2b2workbench

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

5. If launching Eclipse for the first time you will be greeted by a **Welcome page**.
6. In the right upper corner of the page there is an icon of an arrow that says **Workbench**. Click on this icon to startup the Eclipse SDK.



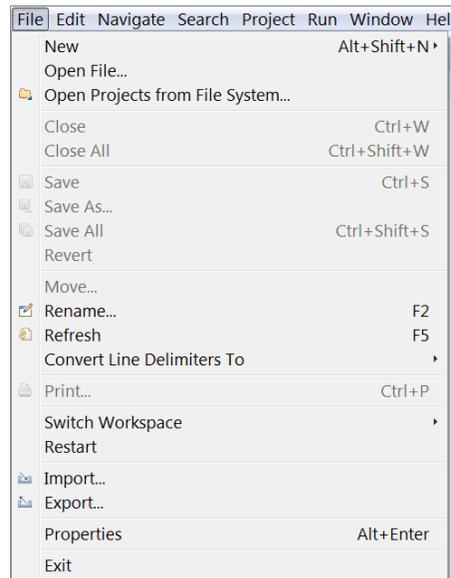
7. The Eclipse workbench will open.



Importing the i2b2 Workbench projects

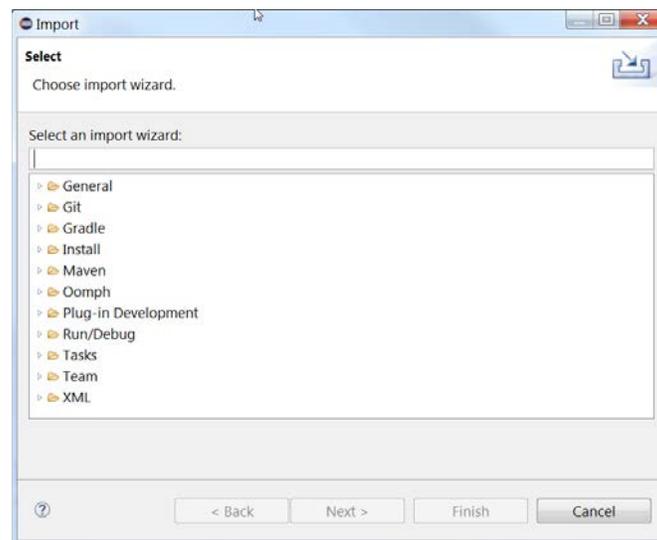
1. Select **File** from the **Eclipse toolbar**.

2. A drop-down menu will display.



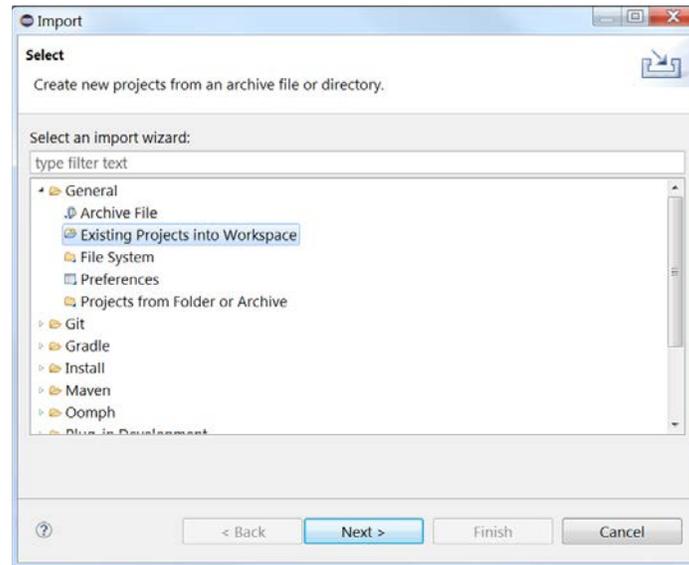
3. Select **Import**.

4. The **Import wizard** will open to the **Select** page.

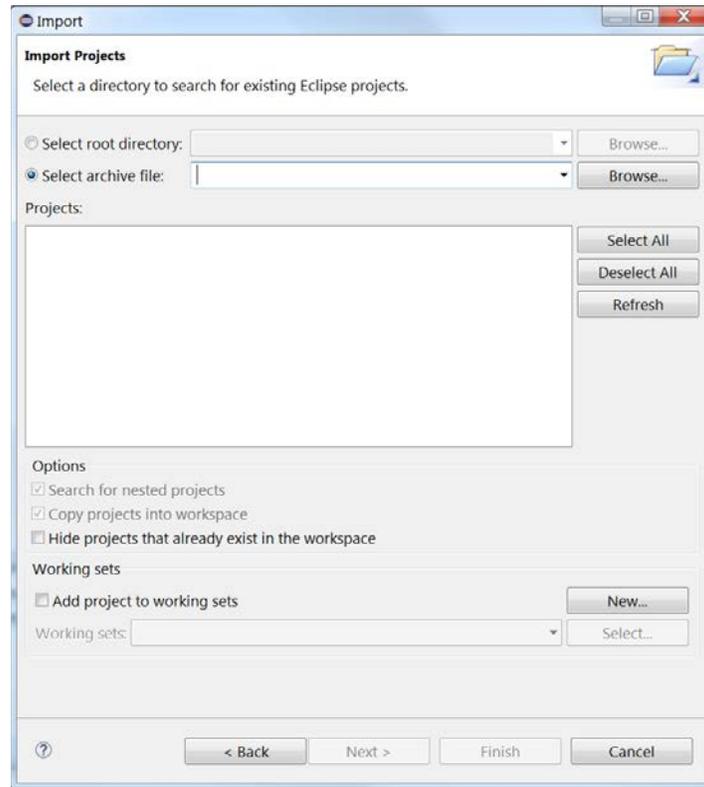


5. Double click on the folder called **General**.

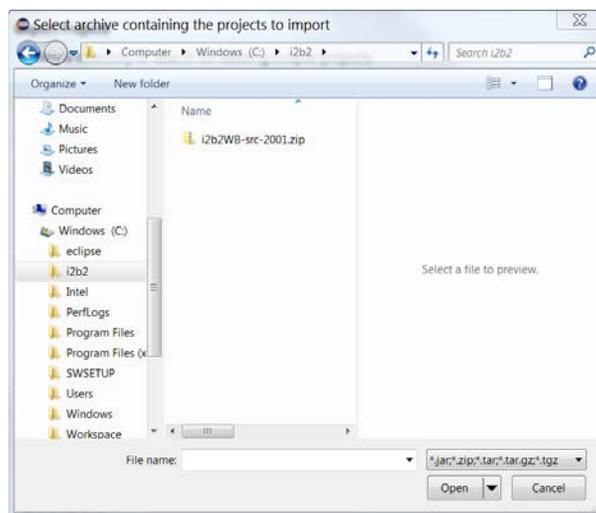
6. Click on **Existing Projects into Workspace**.



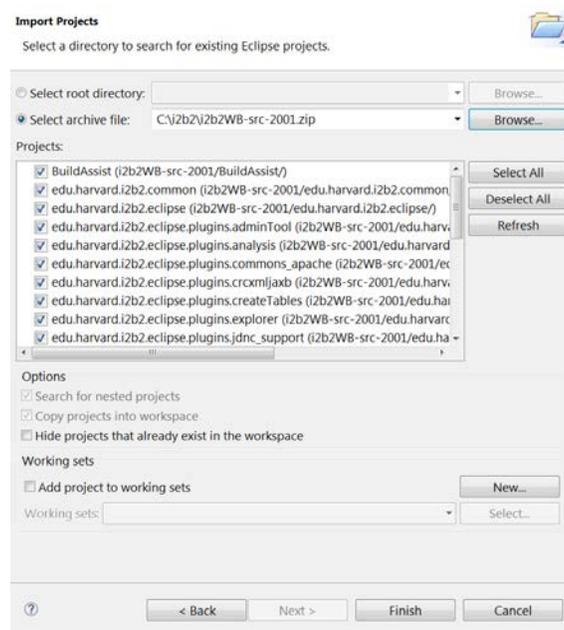
7. Click on the **Next** button.
8. The **Import Projects** page of the import wizard will open.
9. Click on the *radial button* next to **Select archive file**.



10. Enter the path of the **i2b2 archive file** or browse for the location where you downloaded the **i2b2Workbench-src-xxx.zip** file. (Step was done in the Prerequisite section called i2b2 Workbench Source Code)



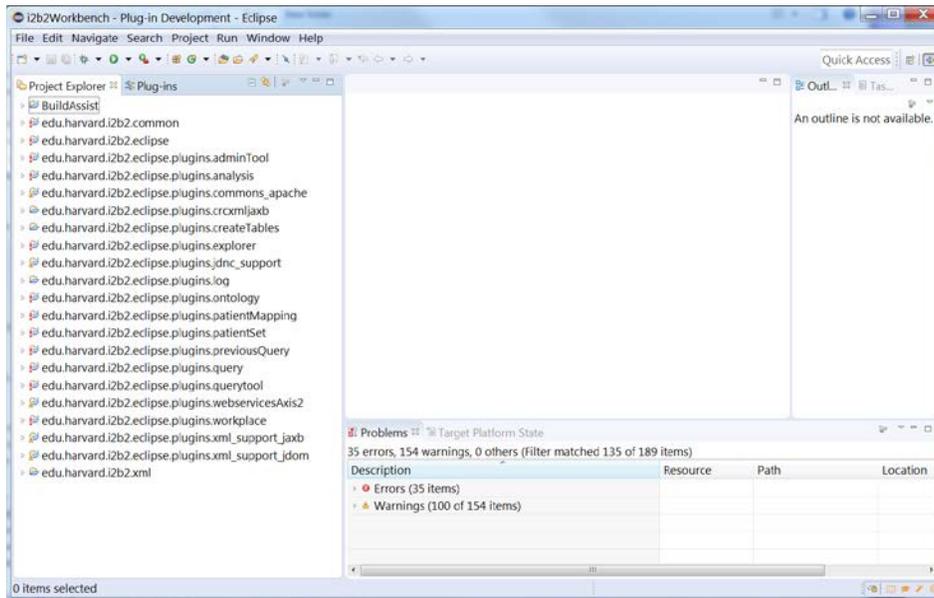
11. Highlight the zip file by clicking on its name.
12. Click on the **Open** button.
13. All the projects in the zip file will now be listed in the **Projects** section of the *Import Projects* page.



14. Verify all the projects are checked off.
15. Click on **Finish**.
16. At the bottom of the Import Projects window you will see a status bar indicating the projects are being imported.



17. Once the projects have finished importing the Import Projects window will close and all the i2b2 Workbench projects will appear in your Eclipse workbench.

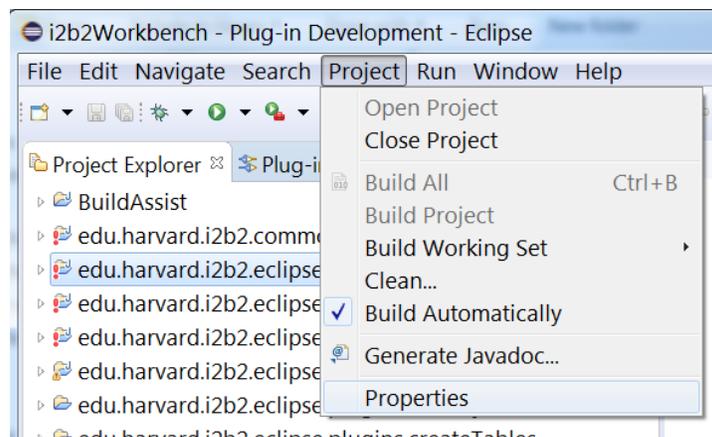


Building the i2b2 Workbench within the Eclipse IDE

Java Compiler Settings

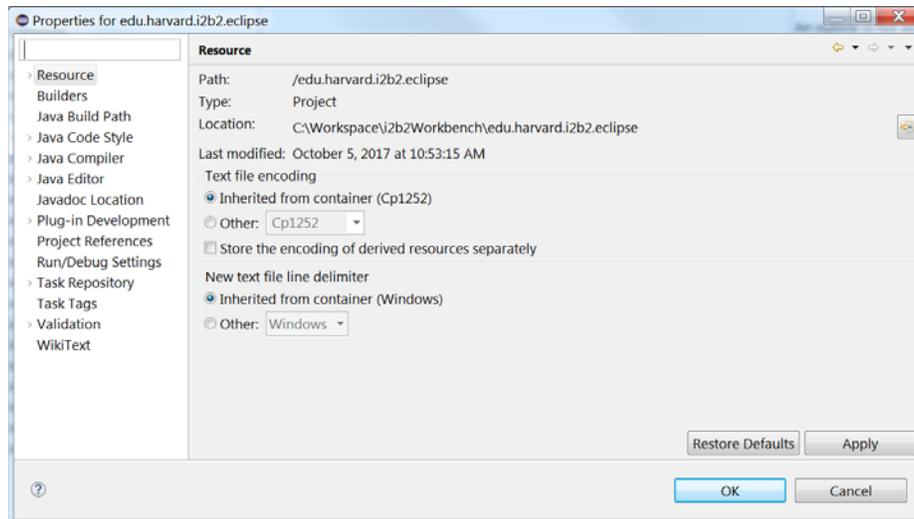
The following are the steps to check your Java Compiler settings.

1. In the Eclipse workbench locate the project called **edu.harvard.i2b2.eclipse**.
2. Highlight the project by clicking on its name.
3. On the **Eclipse menu bar**, click on **Project**.

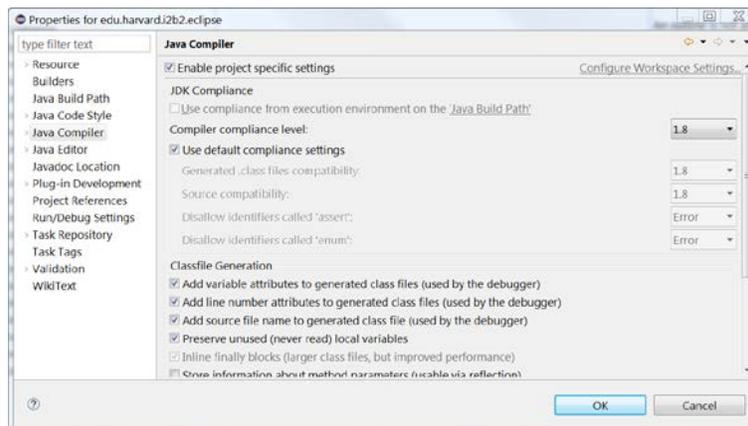


4. Select **Properties** from the drop-down menu.

5. The **Properties window** will open.



6. In the left navigation bar, click on **Java compiler** to display the Java compiler settings.



7. Verify the **Compiler compliance level** is set to 1.8.

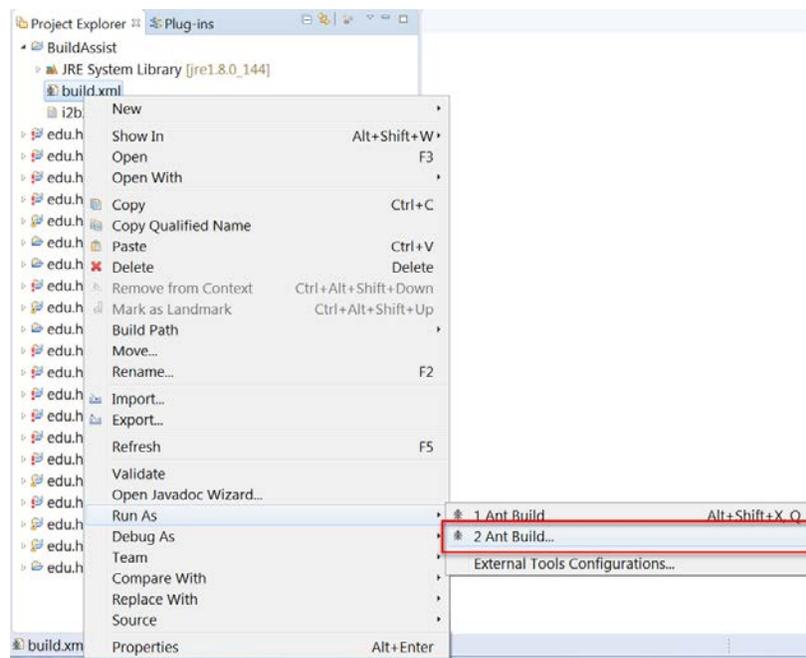


8. Click on **OK**.

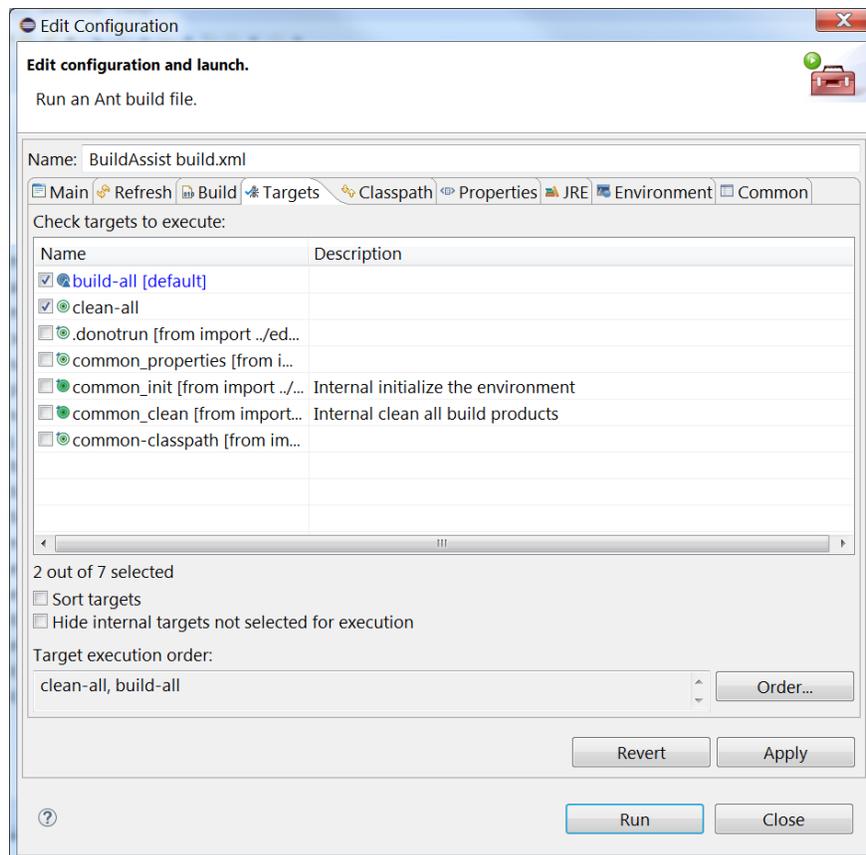
Cleaning and Building the Projects

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

1. Double click on the **BuildAssist** project to open it.
2. Using the *right mouse button*, click on the **build.xml** file.
3. A pop-up menu will display with a list of options.
4. Using your *mouse*, hover over **Run As** to display a sub-menu.
5. On the sub-menu, there are two **Ant-Build** options, select the **second** occurrence.



6. The **Edit Configuration dialog box** for the Ant build script will open.

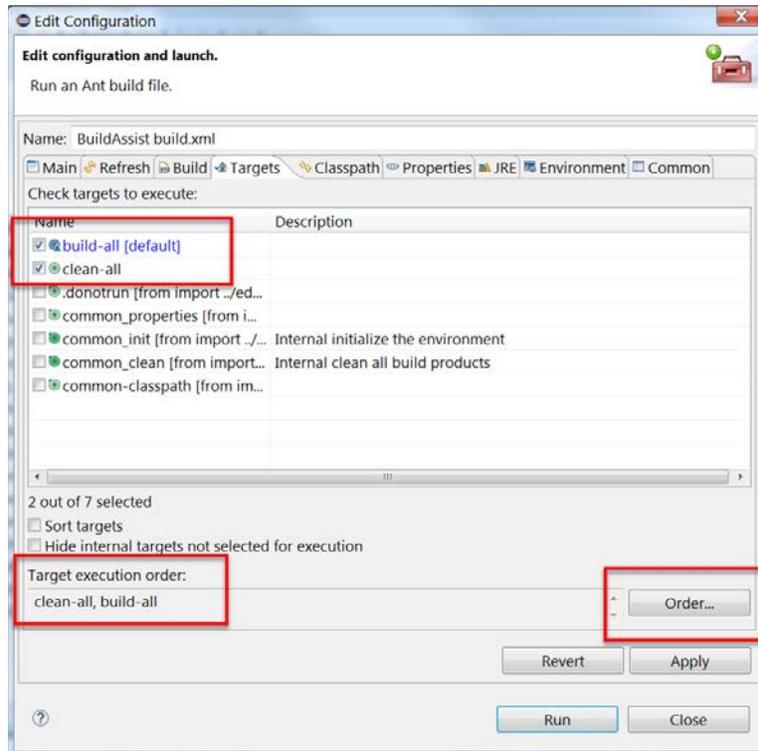


7. Verify the following settings:

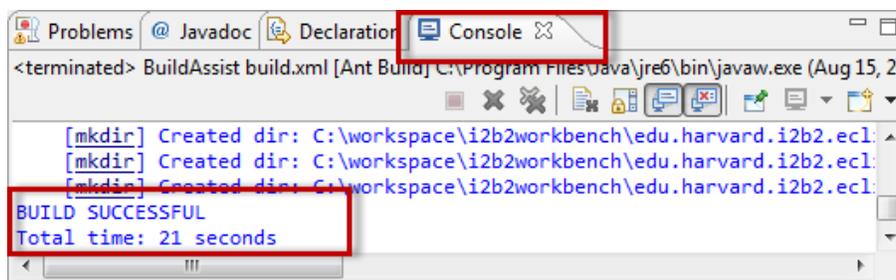
- The **clean-all** and **build-all** boxes are checked.
- The **target execution order** is: "**clean-all, build-all**"

Warning

The execution order is important because if you are building first and then cleaning you will be wiping out everything that you just built and will have errors. If either of these items are not checked off, then add them and fix the order by clicking on the **Order** button to change it.

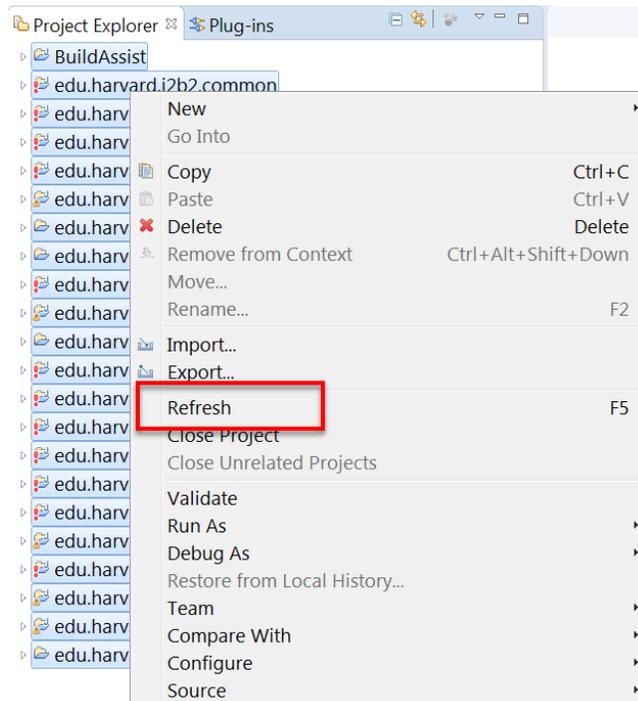


8. Click on the **Run** button to execute.
9. Once the build has finished you should see **BUILD SUCCESSFUL** in the console view.



10. The last step in the build process is to refresh all the projects.
11. Select all the projects.
12. Using the *right mouse button*, click on any one of the highlighted projects.
13. A pop-up menu will display.

14. Select **Refresh**.



15. A window will open showing the status of the refresh process.

16. Once the refresh has finished the status window will close.

Note

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

17. Verify there are no errors once the process of building the workbench has completed. There are two ways to check for errors:

18. A red X appears on the project folder (📁) indicating there is an error in that project.

19. Look at the **Problems view** to see if there are any errors listed.

Note

If an error does occur you may want to try and repeat the steps shown above (build / refresh). Sometimes this will fix any errors that occur during the build process.

Configure the i2b2 Workbench

Define Hive Location

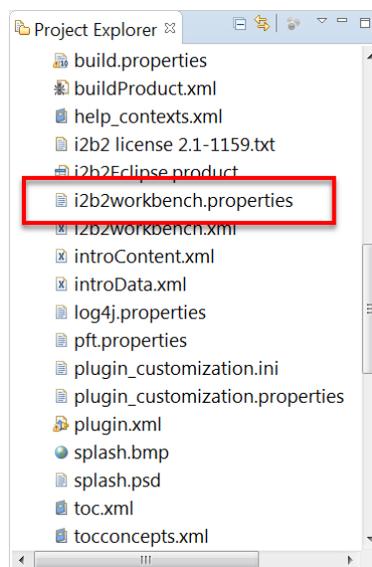
In the [edu.harvard.i2b2.eclipse project](#) there is a file called **i2b2workbench.properties**. When a user starts the logon process the i2b2 Workbench looks to this file to find out where the i2b2 hive is located.

By default, the i2b2 Workbench is setup to work with the **i2b2demo** hive. For the i2b2 Workbench to work with your hive you will need to update the properties file with your site information.

Access i2b2Workbench.properties file within Eclipse application

The method of accessing the i2b2Workbench.properties file from within the Eclipse applications is the same for both Windows and Macintosh.

1. Double click on the **edu.harvard.i2b2.eclipse** project to open its folder.
2. Double click on the **i2b2workbench.properties** file.

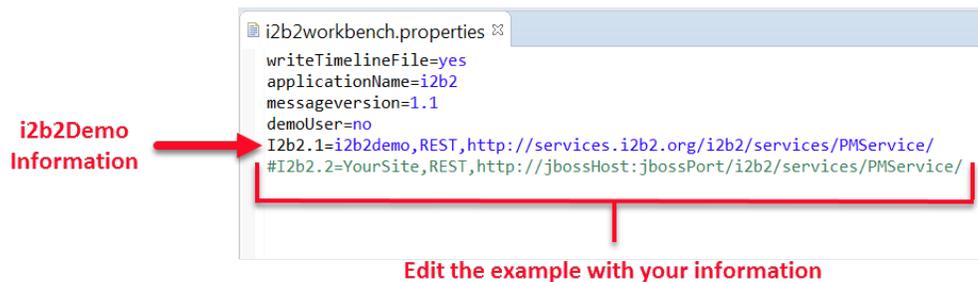


3. The properties file will open.

Editing the i2b2Workbench.properties file

The last two lines of the i2b2Workbench.properties file contain information about the location of the hive.

- The second to last line is the information for i2b2Demo
- The last line is an example that can be edited with your information



1. Edit the following information:

- a. Remove the pound sign (#) at the beginning of the line as this denotes a comment. If it is not removed the i2b2 Workbench will ignore the string and you will not be able to log onto that domain.
- b. Update **YourSite** with the named of **your** domain. Keep in mind this is case sensitive.
- c. Update **jbossHost:jbossPort** with the name of **your** jboss host and its port.

Note

If you do not want to see i2b2demo as an option when logging into the i2b2 Workbench you can either remove it or comment it out by placing a pound sign (#) in front of the string.

2. If you have more than one environment you can include all of them in this on properties file by doing the following:
 - a. Copy the string you just edited to a new line.
 - b. Update the domain and jboss information with the correct information for the additional hive.
 - c. Make sure you advance the number at the beginning of the string ("I2b2")

Example:

I2b2.1=i2b2demo,REST,http://services.i2b2.org/i2b2/services/PMService/

I2b2.2=**TestDomain**,REST,http://**mytestJboss.org:jbossPort**/i2b2/services/PMService/

I2b2.3=**ProdDomain**,REST,http://**mytestJboss.org:jbossPort**/i2b2/services/PMService/

3. Click on the **Save** button () to save the changes to the file.

Define Run Configurations

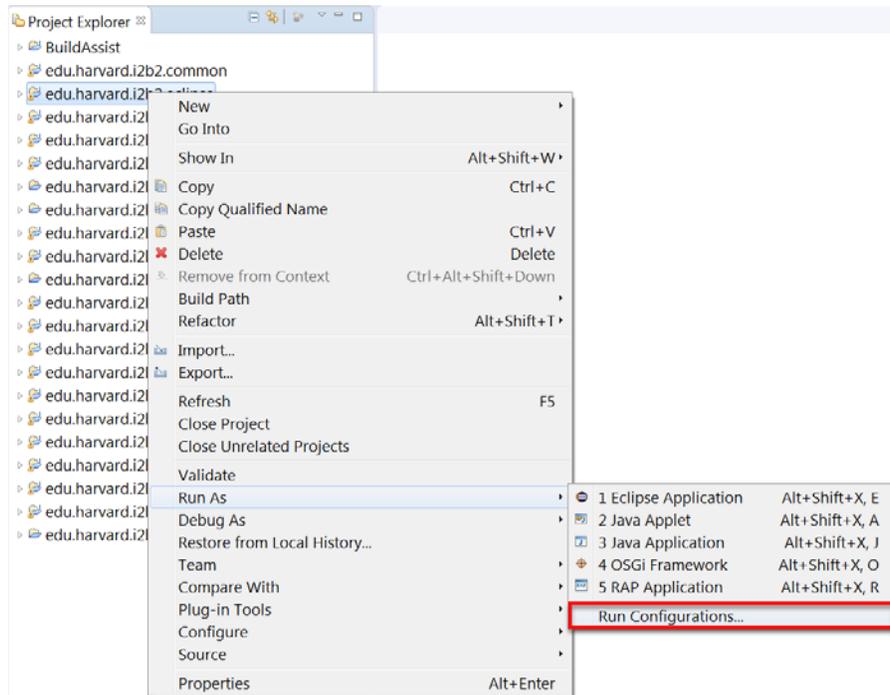
To run the i2b2 Workbench from within Eclipse, the *Eclipse Application* launcher needs to be configured to work correctly in your environment. You have a choice to either use once of the pre-configured platform specific launch configuration files provided in the i2b2 Workbench source code or to configure your own settings. The next few sections will outline how to access the run configurations window, using pre-configured configurations or setting up a new one

 **Note**

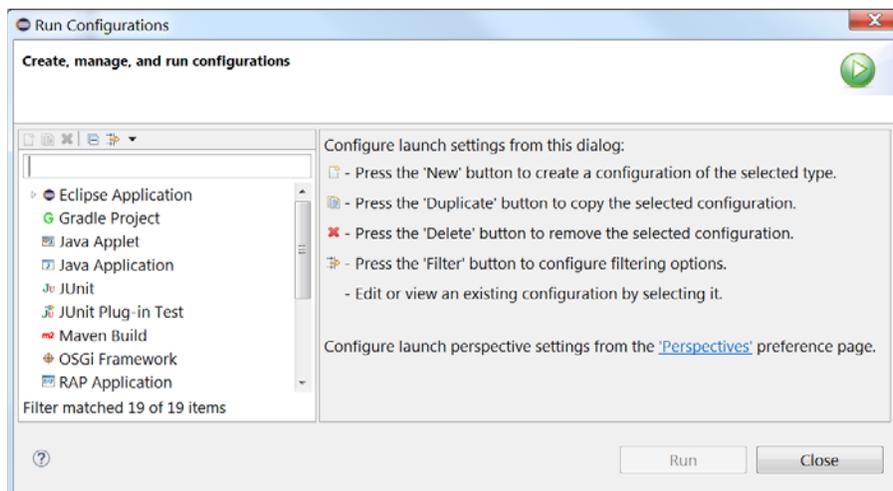
Workbench Release 2.0.01 is the first release to contain platform-specific launch configuration files.

Accessing Run Configurations Window

1. Using the *right mouse button*, click on the **edu.harvard.i2b2.eclipse** project.
2. A pop-up menu will display a list of options.
3. Using your *mouse*, hover over **Run As** to display a sub-menu.
4. Click on **Run Configurations**.



5. The **Run Configurations window** will open.



On the left side of the Run Configuration window is a navigation panel. Within this panel are several types of run configurations that can be defined. The items that display depend on what additional software / plugins you have installed to work within Eclipse. For this developer's guide and the i2b2 Workbench application we will be focused on the configuration settings for **Eclipse Application**.

Using one of the pre-configured launch files

As stated earlier, several platform-specific Eclipse Application launch files have been pre-configured and delivered with the i2b2 Workbench source code. This section will outline how to use one of these files.

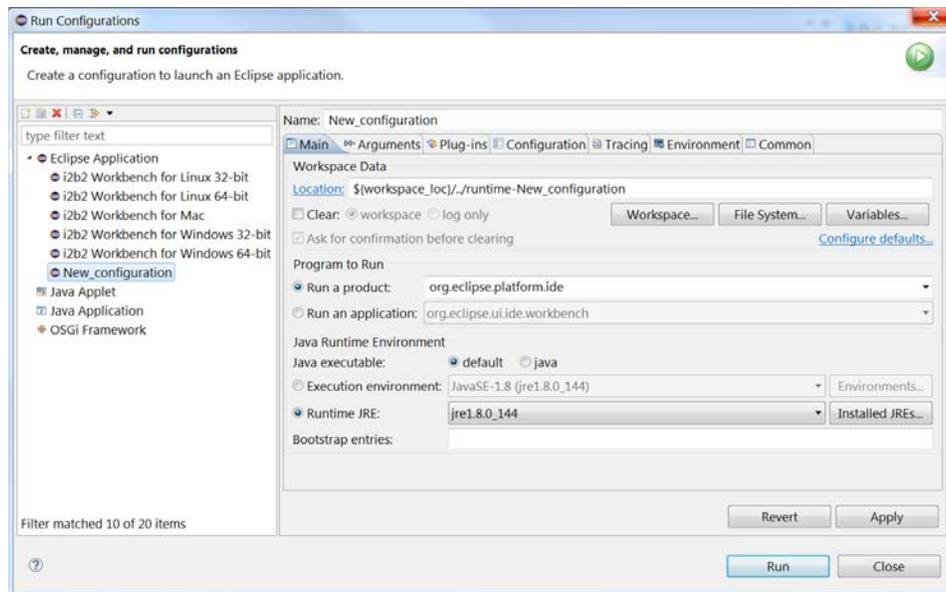
- i2b2 Workbench for Linux 32-bit
- i2b2 Workbench for Linux 64-bit
- i2b2 Workbench for Mac
- i2b2 Workbench for Windows 32-bit
- i2b2 Workbench for Windows 64-bit

- 1.** In the **Run Configurations** window, expand the **Eclipse Application**.
- 2.** A list of pre-configured launch files will display under the **Eclipse Application**.
- 3.** Select the configuration file that corresponds with your system.
- 4.** On the right side of the Run Configurations window, the **Create, manage, and run configurations** wizard will open to the Main tab.
- 5.** All the appropriate settings will default.
- 6.** Click on **Run** to launch the i2b2 Workbench.
- 7.** The **Splash screen** will display and the i2b2 Workbench will open to the login window.

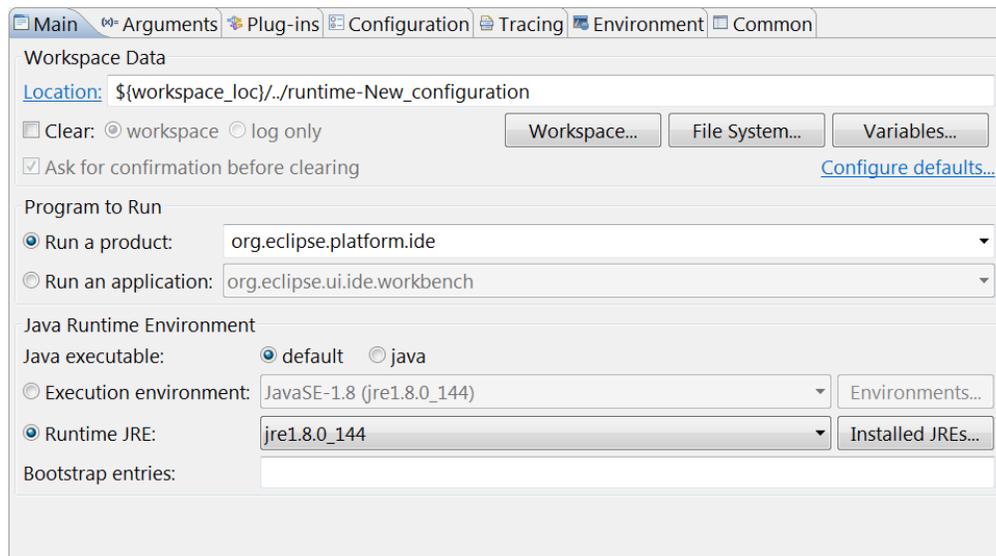
The next three sections outline the process of setting up the configuration information on the *Main*, *Arguments*, and *Plug-ins* tabs.

Setting up a new configuration

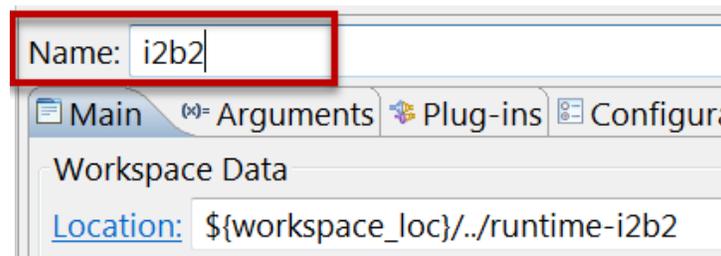
1. In the **Run Configurations** window, double click on **Eclipse Application** or in the *Run Configurations toolbar* click on the **New** button.
2. The following will occur:
 - a. A new configuration called **New_configuration** will appear under *Eclipse Application*.
 - b. On the right side of the Run Configurations window, the **Create, manage, and run configurations** wizard will open to the Main tab.



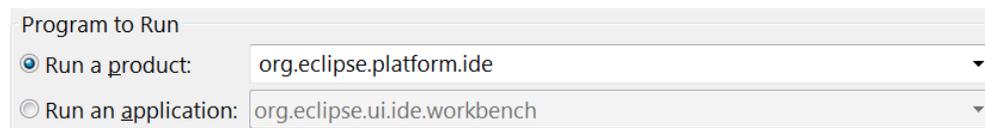
Main Tab



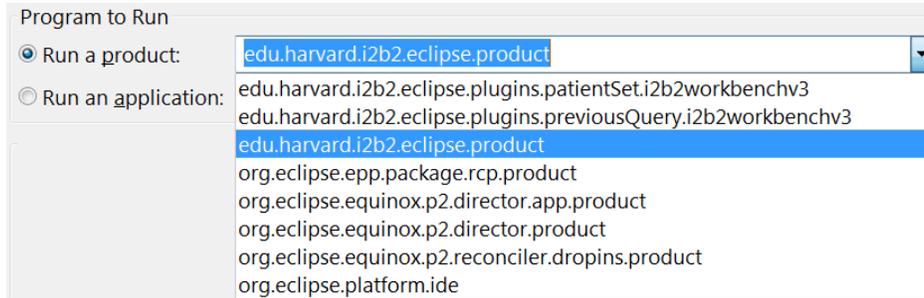
3. Enter a meaningful name at the **Name** field.



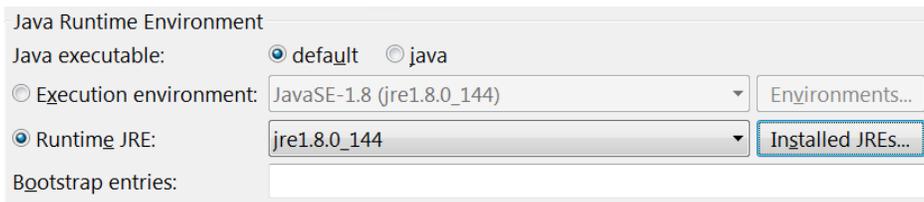
4. In the **Program to Run section**, verify the *radial button* next to **Run a product** is selected.



5. Click on the **down arrow** to display the *drop-down list*.
6. Select **edu.harvard.i2b2.eclipse.product** from the drop down list.

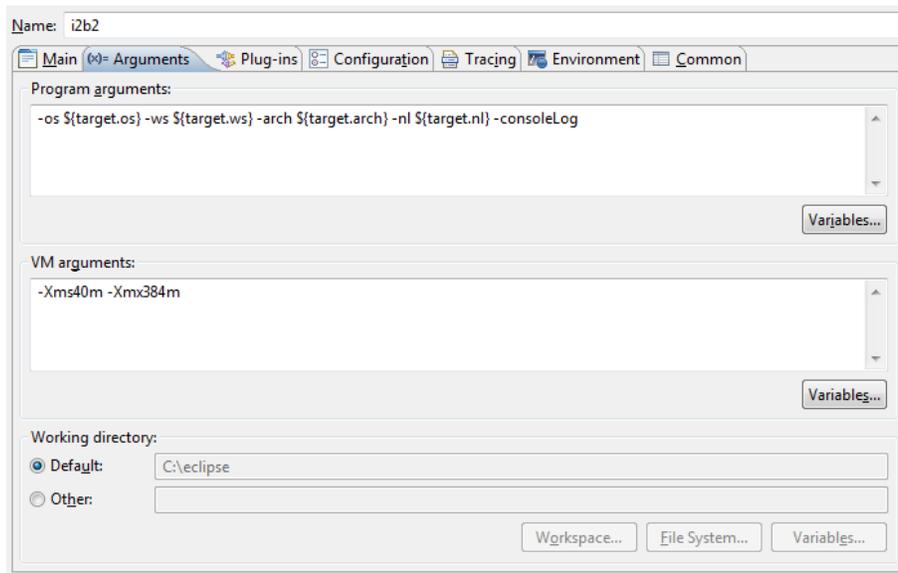


7. Update the **Runtime JRE** in the **Java Runtime Environment**.



Arguments Tab

- 1.** Click on the **Arguments** tab.
- 2.** The Arguments page will display.



3. Edit the **VM arguments** section with the following information.

Windows:

Keep the following arguments that Eclipse will default automatically. If not then they need to be added.

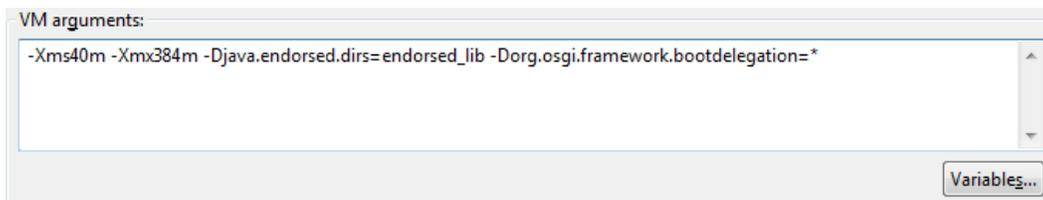
-Xms40m

-Xmx384m

Add the following arguments:

-Djava.endorsed.dirs=endorsed_lib

-Dorg.osgi.framework.bootdelegation=*



Note

The value defined at Xmx is equal to the amount of physical memory on your computer. If needed you can increase this value to equal the physical memory on your computer.

Macintosh:

Keep the following arguments that Eclipse will default automatically. If not then they need to be added.

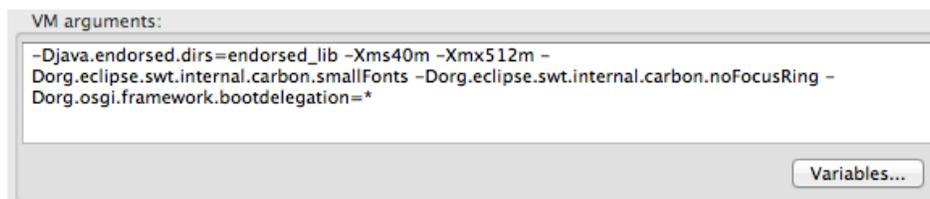
- Xms40m
- Xmx512m
- Dorg.eclipse.swt.internal.carbon.smallFonts

Remove the following arguments that Eclipse will default automatically.

- Xdock:icon=../Resources/Eclipse.icns
- XstartOnFirstThread

Add the following arguments:

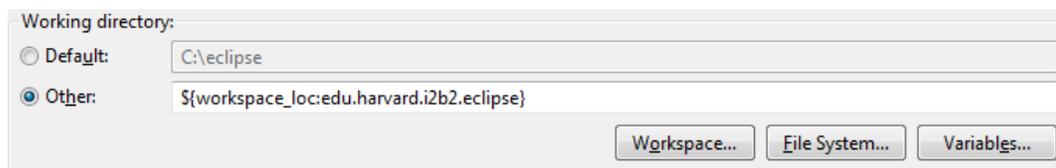
- Djava.endorsed.dirs=endorsed_lib
- Dorg.eclipse.swt.internal.carbon.noFocusRing
- Dorg.osgi.framework.bootdelegation=*



4. In the **Working directory** click on the *radial button* next to **Other**.

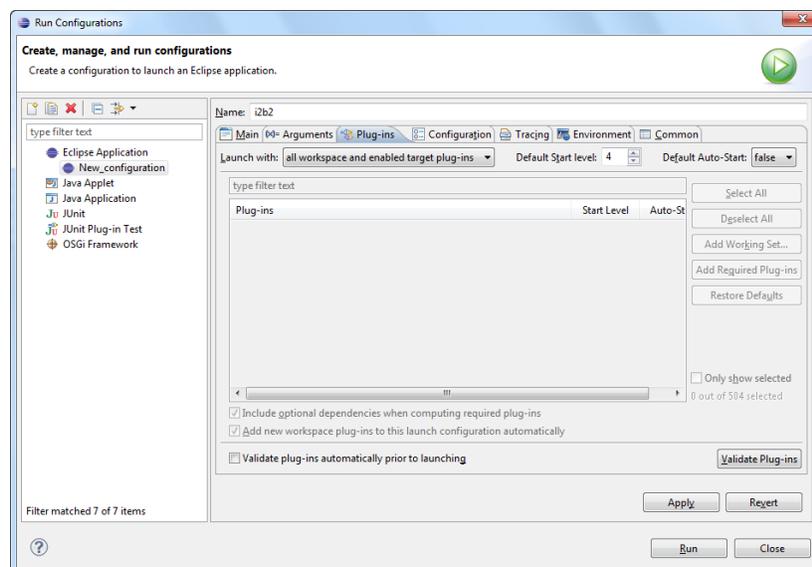


5. Click on the **Workspace** button.
6. The **Folder Selection window** will open.
7. Select **edu.harvard.i2b2.eclipse**.
8. Click on the **OK** button.
9. The **Other** field will be populated with the workspace information.



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 **workspace and enabled target plug-ins** are launched.



Important

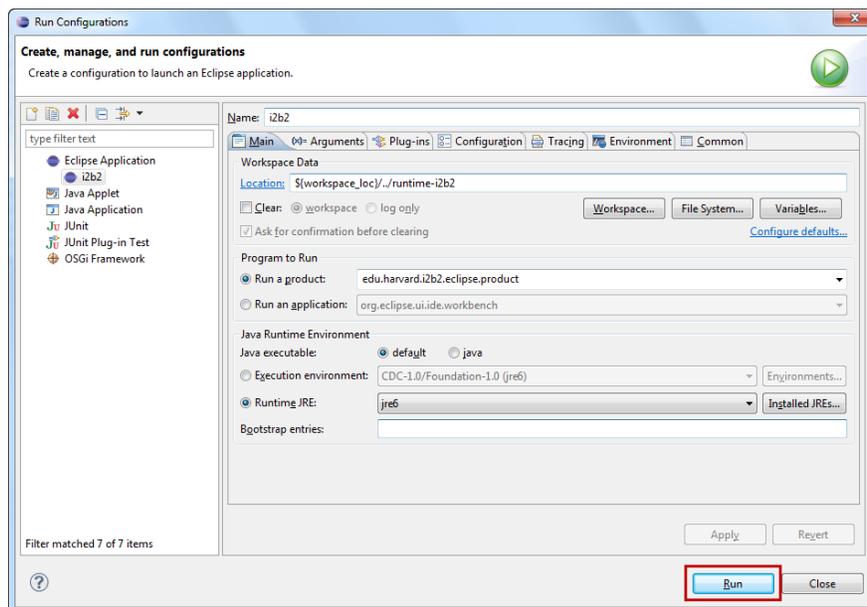
We recommend you click on the **Apply** button to save your settings.

Execute

Different Methods to Run an Eclipse Application

There are three ways in which you can run an Eclipse Application like the i2b2 Workbench.

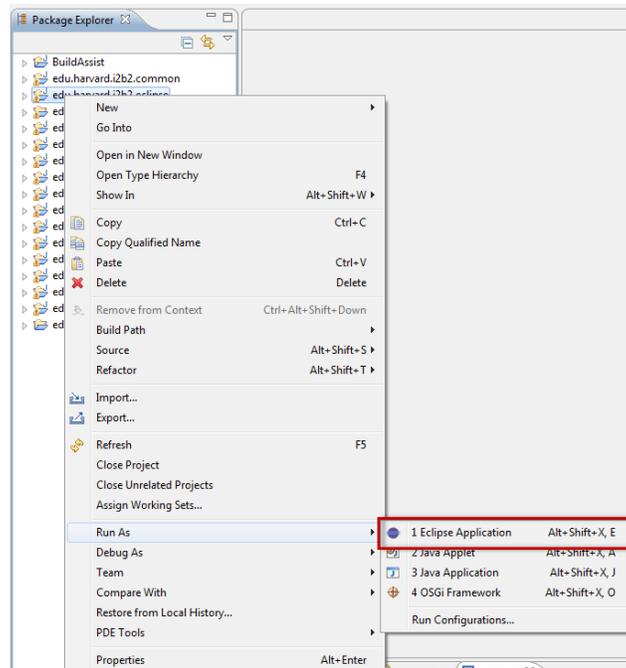
METHOD 1: Click on the **Run** button from within the **Run Configuration window**.



 Tip

See section called *Accessing Run Configurations Window* if you need to know how to access this window.

METHOD 2: Using the *right mouse button* to display the pop-up menu and selecting **Run As** and then **Eclipse Application**.

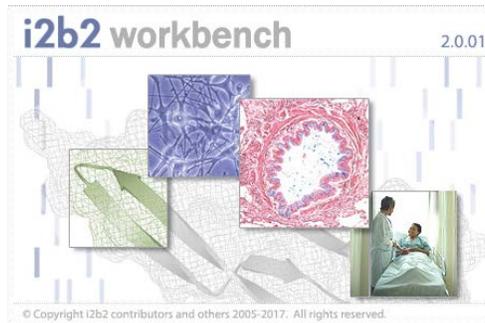


METHOD 3: Click on the **Run** button on the Eclipse toolbar (🟢).

Running the i2b2 Workbench

The previous section defines the different methods you can use to launch the i2b2 Application. This section assumes you have already selected to run the application using one of those methods.

1. The **splash screen** for the i2b2 Workbench will appear.



2. The **Login** window will open.

A screenshot of a dialog box titled "Enter User ID and Password". The dialog box has a light gray background and contains the following fields and controls:

- "Target location" dropdown menu with "i2b2demo" selected.
- "User name" text box containing "demo".
- "Password" text box containing "*****".
- A checkbox labeled "Start as demonstration only" with a question mark icon to its right.
- "Login" and "Cancel" buttons.
- A URL at the bottom: "http://services.i2b2.org/i2b2/services/DMService/".

3. Enter your **user name** and **password**. For the **i2b2demo** site you can enter the following user credentials.

User name: demo

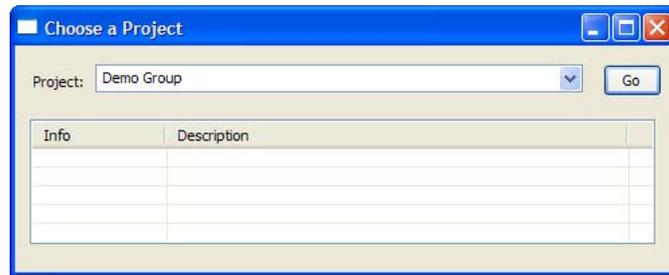
Password: demouser

4. Click on the **Login** button.

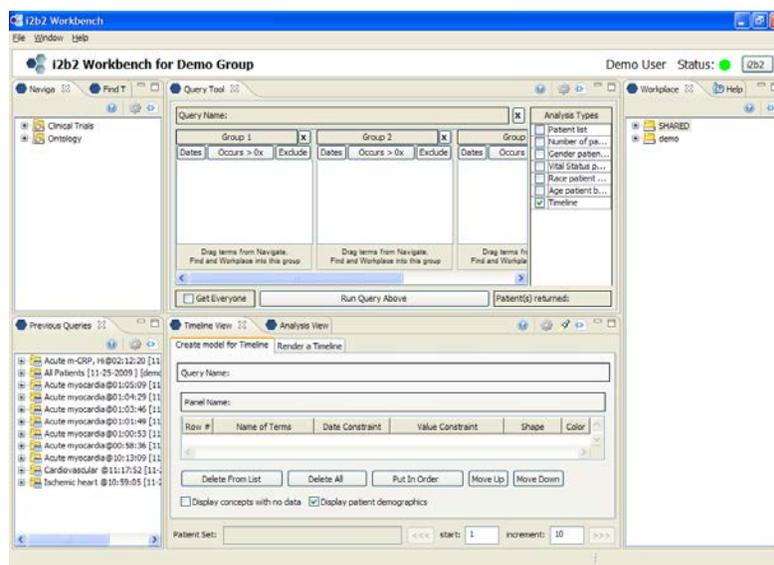
5. The **Choose a Project** dialog window will open.

Note

The project dialog box will only appear if your user has access to multiple projects. If you only have access to one project you will be brought to the i2b2 Workbench desktop.



6. Accept the default project by simply clicking on the **Go** button.
7. The **desktop** of the i2b2 Workbench will open.



Reconfiguring the Default Layout

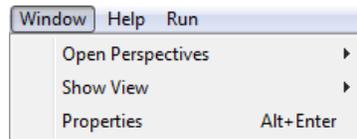
The default layout can be reconfigured to the user's needs.

Adding a View

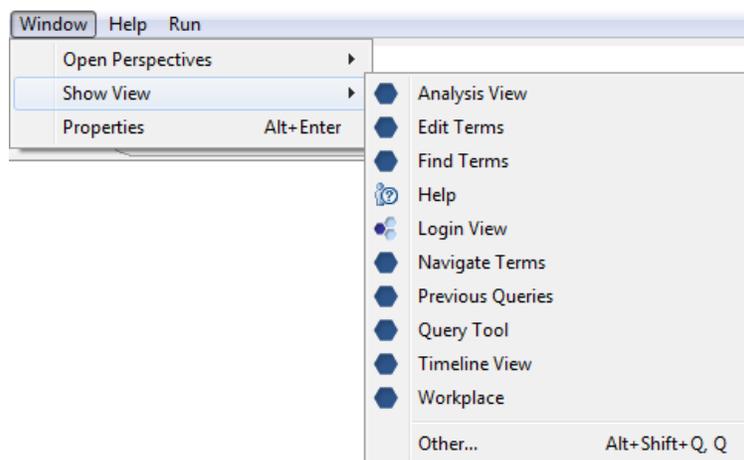
The following steps explain how to add a view to the desktop of the i2b2 Workbench.

1. On the toolbar, click on **Window**.

2. A drop-down menu will display.



3. Using your *mouse*, hover over **Show View** to display the sub-menu listing all the core views.



4. Click on the name of the view you want to add.

5. The view will be added to the desktop area of the i2b2 Workbench.

Note

Core views are those views that are associated with the i2b2 cells that are part of the core hive. Select *Other* to see a list of all views including the optional ones.

Views	Description
Analysis View	Provides a visualization of the analysis results for queries that users run in the i2b2 Workbench.

Edit Terms	Allows users to enter new containers, folders, concepts, modifier containers, modifier folders, and modifiers as well as edit existing ones provided they are not locked.
Find Terms	Allows users to search for concepts and modifiers.
Help	Displays the help documentation for the active view.
Navigate Terms	Displays the i2b2 vocabulary concepts and modifiers that are maintained in the Ontology Management Cell of the i2b2 Hive .
Previous Queries	Displays the most recent queries run by a user. The results associated to the query are also available for viewing.
Query Tool	The Query Tool view is designed to simplify the process of retrieving information from the database associated to the i2b2 Workbench.
Timeline View	Displays the information in a graphical representation of a chronological sequence of events
Workplace	Information in the Workplace view is related to the most common concepts and queries that an individual uses and in essence becomes their personal workspace. The main objectives of this view are: <ul style="list-style-type: none"> ▪ Save and organize user specific items. ▪ Share information with other project users. ▪ Project managers can view the workplace for individual team members.

Moving a View

Move the position of any view by clicking on the view's tab and dragging it to another area of the i2b2 Workbench (desktop area).

Removing a view

At the top of each view is a tab that contains the name of the view and a close button. Simply click on the close button (✕) to remove a view from the desktop.

Package structure

The **archive file** contains the projects listed below. The project / directory **edu.harvard.i2b2.eclipse** is considered the *base plug-in* for the i2b2 Workbench and contains the property and configuration files.

Directory	Description
BuildAssist	Files to build the i2b2 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 user interface for the top panel)
edu.harvard.i2b2.plugins.adminTool	User interface for the <i>Managers Tool</i> view.
edu.harvard.i2b2.plugins.analysis	User interface for the <i>Analysis</i> view.
edu.harvard.i2b2.plugins.common.apache	Third party jars needed for common functions.
edu.harvard.i2b2.plugins.exporer	User interface for the <i>Timeline</i> view.
edu.harvard.i2b2.plugins.jdnc_support	Third party jars needed for the user interface support
edu.harvard.i2b2.plugins.log	Third party jars needed for logging
edu.harvard.i2b2.plugins.ontology	User interface for the <i>Ontology</i> views. This includes: Navigate Terms Find Terms Edit Terms
edu.harvard.i2b2.plugins.patientMapping	User interface for the <i>Patient Mapping</i> view.
edu.harvard.i2b2.plugins.patientSet	User interface for the <i>Find Patient Set</i> view.
edu.harvard.i2b2.plugins.previousQuery	User interface for the <i>Previous Query</i> view
edu.harvard.i2b2.plugins.query	User interface for the <i>Query</i> view
edu.harvard.i2b2.plugins.querytool	User interface for the <i>Temporal Query</i> view
edu.harvard.i2b2.plugins.webservicesAxis2	Third party jars needed for web service support
edu.harvard.i2b2.plugins.workplace	User interface for the <i>Workplace</i> views. This includes: Workplace Find in Workplace
edu.harvard.i2b2.plugins.xml_support_jaxb	Third party jars needed for jaxb / message processing
edu.harvard.i2b2.plugins.xml_support_jdom	Third party jars needed for jdom / message processing
edu.harvard.i2b2.plugins.xml	XSD for the hive

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.