i2b2 Installation Guide (Linux)

Data Repository (CRC) Cell

Document Version:	*1.5.1*
i2b2 Software Version:	*1.5*

Table of Contents

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. This guide with the provided source code will help you deploy the i2b2 CRC cell.

1. PREREQUISITES

1.1 Required Software

If you installed the prerequisite software from the Hive Installation Guide you may skip this section.

1.1.1 Java JDK

JDK 6.0 is recommended and can be downloaded from the java website: http://java.sun.com/products/

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

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

1.1.2 JBoss 4.2.2GA

Download JBoss (jboss-4.2.2.GA.zip) from the jboss website: http://labs/jboss.com/jbossas/downloads

1. Unzip jboss-4.2.2.GA.zip into a directory of your choice.

    Example*:* /opt/jboss-4.2.2.GA or *YOUR_JBOSS_HOME_DIR*

2. Set JBoss JVM to run with 1GB of extended memory.

3. Edit '*YOUR_JBOSS_HOME_DIR*/bin/run.conf' and change the JAVA_OPTS memory setting to those shown below. (-Xms512m -X, x1024m)

```
#
# Specify options to pass to the Java VM.
#
if [ "x$JAVA_OPTS" = "x" ]; then
    JAVA_OPTS="-Xms512m –Xmx1024m
    -Dsun.rmi.dgc.client.gcInterval=3600000
    -Dsun.rmi.dgc.server.gcInterval=3600000"
```

4. If default port 8080 is unavailable because another application is using this port then edit the server.xml file ('*YOUR_JBOSS_HOME_DIR*/server/default/deploy/jboss-web.deployer/server.xml') to reconfigure the following two items:

    a. The non-SSL HTTP/1.1 Connector to another port such as 9090

```
< ! –Define a non-SSL HTTP/1.1 Connector on port 9090 - >
<Connector port="9090" address="${jboss.bind.address}"
        maxThreads="250" maxHttpHeaderSize="8192"
        emptySessionPath="true" protocol="HTTP/1.1"
        enableLookups="false" redirectPort="8443"
        acceptCount="100" connectionTimout="20000"
        disableUploadTimeout="true" />
```

    b. The AJP 1.3 Connector to another port such as 9009

```
< ! –Define an AJP 1.3 Connector on port 9009 - >
<Connector port="9009" address="${jboss.bind.address}"
        maxThreads="250" maxHttpHeaderSize="8192"
        protocol="AJP/1.3" emptySessionPath="true"
        enableLookups="false" redirectPort="8443" />
```

1.1.3 Apache Ant 1.6.5

Download *Apache Ant version 1.6.5* (apache-ant-1.6.5-bin.zip) from the following Apache website: http://archive.apache.org/dist/ant/binaries/

1. Unzip the file into a directory of your choice.

   Example: /opt/apache-ant-1.6.5 or *YOUR_ANT_HOME_DIR*

1.1.4 Apache Axis2 1.1

Download *Apache Axis2 version 1.1* from the following Apache website: http://ws.apache.org/axis2/download/1_1/download.cgi

Select the download type of WAR (Web Archive) Distribution (axis2.war).

1. Create a folder called **i2b2.war** inside 'YOUR_JBOSS_HOME_DIR/server/default/deploy' folder.

2. Unzip axis2.war inside
   'YOUR_JBOSS_HOME_DIR/server/default/deploy/i2b2.war' folder.

1.1.5 Oracle Express Edition

Download *Oracle Database 10g Express Edition (Universal)* (oracle-xe-univ-
10.2.0.1-1.0.i386.rpm) from the following Oracle website:
http://www.oracle.com/technology/software/products/database/xe/htdocs/102xelinsof
t.html

1. Run *rpm –i oracle-xe-univ-10.2.0.1-1.0.i386.rpm* as root.

   a. Configure the database by running '/etc/init.d/oracle-xe configure' as root

   b. Select HTTP and listener ports (use defaults 8080/1521 if they are
      available).

   c. Select 'Y'es to start on boot when asked.

2. Verify Oracle was properly installed.

   a. Open a browser

   b. Enter http://*yourHost:yourPort*/apex

   c. You should see an Oracle Database Express Edition login screen.

1.1.6 Update Environment Variables

Be sure to set the JAVA_HOME, JBOSS_HOME, and ANT_HOME variables to the
JAVA, JBOSS, and ANT home directories you set up in the previous sections.

Example:

```
# Sample environment variables
JAVA_HOME=/opt/java/jdk1.6.0
JBOSS_HOME=/opt/jboss-4.2.2.GA
ANT_HOME=/opt/apache-ant-1.6.5
PATH=$PATH:$ANT_HOME/bin:$JAVA_HOME/bin
export JAVA_HOME
export JBOSS_HOME
```

```
export ANT_HOME
```

1.1.7 Data Installation or Upgrade

Data installation or upgrade has already been performed.

2. INSTALL

2.1 Installing the Data Repository (CRC) Cell

The next few sections outline the CRC installation process.

2.1.1 Download and Extract Source Code

The following outlines the steps to download and extract the core server source code to a target area.

☞ *Note: You can skip this step if the core server source code was downloaded during a previous installation (e.g. PM or ONT).*

Steps:

1. Set up a target source_directory.

2. Extract core server source code into the target source_directory.

2.1.2 Stop Services

In order to install the CRC properly it is important that certain services are **not** running. The services are further defined below.

2.1.2.1 JBOSS

Verify JBOSS is not running.

- ▪ ./$JBOSS_HOME/bin/shutdown.sh -S

2.1.3 Deploy edu.harvard.i2b2.common

The following outlines the steps to deploy **edu.harvard.i2b2.common**.

🖘 **Note**: *You can skip this step if edu.harvard.i2b2.common was deployed during a previous installation (e.g. ONT).*

Steps:

1. cd source_directory/edu.harvard.i2b2.common

2. Edit the **build.properties** file and set *jboss.home* and *axis2.war.name properties*.

   ```
   jboss.home=YOUR_JBOSS_HOME_DIR
   axis2.war.name=i2b2.war
   ```

3. Run *ant clean dist deploy jboss_pre_deployment_setup*

2.1.4 Build edu.harvard.i2b2.crc.loader

The following outlines the steps to build **edu.harvard.i2b2.crc.loader**.

🖘 **Note**: *The source for the uploader service resides in the edu.harvard.i2b2.crc.loader project and uses the CRC's data source setup.*

Steps:

1. cd source_directory/edu.harvard.i2b2.crc.loader

2. Edit the **build.properties** file and set *jboss.home* and *axis2.war.name properties*

   ```
   jboss.home=YOUR_JBOSS_HOME_DIR
   axis2.war.name=i2b2.war
   ```

3. Edit the **etc/spring/crc_loader_application_directory.properties** file and specify a location for the application properties directory. This location can be anything you desire but must be a directory path that your Linux user has access permission granted.

   ```
   edu.harvard.i2b2.crc.applicationdir=YOUR_JBOSS_HOME_DIR/server/default/conf/crcloaderapp
   ```

4. Edit the **etc/spring/edu.harvard.i2b2.crc.loader.properties** file and set the *project management properties*.

☞ **Note**: *the new location of the PM URL.*

```
edu.harvard.i2b2.crc.loader.ws.pm.url=http://localhost:9090/i2b2/rest/PMService
/getServices
# Flag to bypass project management cell
edu.harvard.i2b2.crc.loader.ws.pm.bypass=false
edu.harvard.i2b2.crc.loader.ws.pm.bypass.role=ADMIN
edu.harvard.i2b2.crc.loader.ws.pm.bypass.project=Demo
```

If you are using the **file repository cell** and want to connect the CRC loader to it then set the *FileRepository cell property* setting.

```
# # # # # # # # # # # # # # # # #
# File Management Cell
# # # # # # # # # # # # # # # # #

edu.harvard.i2b2.crc.loader.ws.fr.url=http://localhost:9090/i2b2/services/FRServi
ce/
```

5. Edit the ***etc/spring/CRCLoaderApplicationContext.xml***

    a. Specify the *jdbc properties* to locate the CRC_DB_LOOKUP table.

```
<bean id="LoaderLookupDS" class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url" value="jdbc.oracle:thin:@localhost:1521:xe"/>
    <property name="username" value="i2b2hive"/>
    <property name="password" value+"i2b2hive_pswd"/>
</bean>
```

    b. Update entry in *edu.harvard.i2b2.crc.loader.properties* to reflect the above change.

```
# # # # # # # # # # # # # # #
# Datasource Lookup properties
# # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.loader.ds.lookup.datasource=LoaderLookupDS
edu.harvard.i2b2.crc.loader.ds.lookup.servertype=ORACLE
edu.harvard.i2b2.crc.loader.ds.lookup.schemaname=i2b2hive
```

6. Run *ant -f build.xml clean dist*

☞ **Note**: *The 'deploy' target is not used. The loader will be deployed as part of the edu.harvard.i2b2.crc.project.*

2.1.5 Setup edu.harvard.i2b2.crc

The following outlines the steps to setup **edu.harvard.i2b2.crc**.

Steps:

1. cd source_directory/edu.harvard.i2b2.crc

2. Edit the ***build.properties*** file and set *jboss.home and axis2.war.name properties*.

```
jboss.home=YOUR_JBOSS_HOME_DIR
axis2.war.name=i2b2.war
#jms persistance could be either oracle or mssql
jboss.jms.persistance=oracle
```

3. Edit the ***etc/spring/crc_application_directory.properties*** file and specify a location for the application properties directory. This location can be anything you desire but must be a directory path that your Linux user has access permission for.

```
edu.harvard.i2b2.crc.applicationdir=YOUR_JBOSS_HOME_DIR
/server/default/conf/crcapp
```

4. Edit the **etc/spring/crc.properties** file and set *project management properties*.

☞ **Note**: *the new location of the PM URL.*

```
# # # # # # # # # # # # # # # #
# Project Management Cell
# # # # # # # # # # # # # # # #
queryprocessor.ws.pm.url=http://localhost:9090/i2b2/rest/PMService/getServices
# Flag to bypass project management cell
queryprocessor.ws.pm.bypass=false
queryprocessor.ws.pm.bypass.role=ADMIN
queryprocessor.ws.pm.bypass.project=Demo
```

5. Edit the **Ontology property settings**; change the *hostname* and *port*

```
# # # # # # # # # # # # # # # #
# Ontology Cell
# # # # # # # # # # # # # # # #
queryprocessor.ws.ontology.url=http://localhost:9090/i2b2/rest/OntologyService/getTermInfo
```

6. Create the **CRC service user account** in the Project Management cell and assign the roles "*User*" and "*Data_Obfsc*" to the user. The CRC uses this PM user in the calls to the Ontology and PM cells.

```
# # # # # # # # # # # # # # # #
# CRC service account properties
# # # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.pm.serviceaccount.user=OBFSC_SERVICE_ACCOUNT
edu.harvard.i2b2.crc.pm.serviceaccount.password=demouser
```

7. Update **QT_BREAKDOWN_PATH** table: Based on your Ontology mapping, update the parent item key values used in the analysis break down calculation.

Name	Value

PATIENT_GENDER_COUNT_XML	\\i2b2\i2b2\Demographics\Gender\
PATIENT_RACE_COUNT_XML	\\i2b2\i2b2\Demographics\Race\
PATIENT_VITALSTATUS_COUNT_XML	\\i2b2\i2b2\Demographics\Vital Status\
PATIENT_AGE_COUNT_XML	\\i2b2\i2b2\Demographics\Age\

8. The following steps tell the CRC, where the "CRC_DB_LOOKUP" table is located.

code_lookup		
PK	**c_domain_id**	**varchar(255)**
PK	**c_project_path**	**varchar(255)**
PK	**c_owner_id**	**varchar(255)**
	c_db_fullschema	**varchar(255)**
	c_db_datasource	**varchar(255)**
	c_db_servertype	**varchar(255)**
	c_db_nicename	**varchar(255)**
	c_db_tooltip	varchar(255)
	c_comment	text
	c_entry_date	datetime
	c_change_date	datetime
	c_status_cd	char(1)

    a. Edit *etc/spring/CRCApplicationContext.xml*.

```
<bean id="CRCDataSourceLookup" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
    <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver"/>
    <property name="url" value="jdbc:oracle:thin:@localhost:1521:xe"/>
    <property name="username" value="i2b2hive"/>
    <property name="password" value="i2b2hive_pswd"/>
```

```
</bean>
```

b. Update the entry in *etc/spring/crc.properties* to reflect the above change.

```
# # # # # # # # # # # # # # # #
# Datasource Lookup properties
# # # # # # # # # # # # # # # #
query.processor.ds.lookup.datasource=CRCDataSourceLookup
query.processor.ds.lookup.servertype=ORACLE
query.processor.ds.lookup.schemaname=i2b2hive
```

9. In the *etc/spring/crc.properties* file, the *PDO paging size* can be adjusted based on the runtime jvm setting.

```
# # # # # # # # # # # # # # # #
# PDO Paging properties
# # # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.pdo.paging.observation.size=75000
```

10. In the **etc/spring/crc.properties** file, you can tune the **default max job count** for the Analysis job queue

```
# # # # # # # # # # # # # # # #
# Analysis Queue properties
# # # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.analysis.queue.medium.timeoutmills=1800000
edu.harvard.i2b2.crc.analysis.queue.medium.maxjobcount=4
edu.harvard.i2b2.crc.analysis.queue.large.timeoutmills=3600000
edu.harvard.i2b2.crc.analysis.queue.large.maxjobcount=4
edu.harvard.i2b2.crc.analysis.queue.medium.jobcheck.timeoutmills=120000
edu.harvard.i2b2.crc.analysis.queue.large.jobcheck.timeoutmills=120000
```

11. If required, you can adjust the ***default query queues timeout*** in second in the ***etc/spring/crc.properties*** file.

```
# # # # # # # # # # # # # # # # # # # # # # # # #
# Setfinder JMS Queue transaction timeout properties
# # # # # # # # # # # # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.jms.small.timeoutsecs=180
edu.harvard.i2b2.crc.jms.medium.timeoutsecs=14400
edu.harvard.i2b2.crc.jms.large.timeoutsecs=43200
```

12. If required, you can change the default user's setfinder lockout parameter in the etc/spring/crc.properties file.

```
# # # # # # # # # # # # # # # # # # # # # # # # #
# CRC Setfinder query lockout parameter
# # # # # # # # # # # # # # # # # # # # # # # # #
edu.harvard.i2b2.crc.lockout.setfinderquery.count=7
edu.harvard.i2b2.crc.lockout.setfinderquery.day=30
```

13. Configure JBoss Datasource:

CRC supports both Oracle and SqlServer. Edit etc/jboss/crc-ds.xml and set the database connection properties. For SqlServer data source, please use the sample provided in the crc-ds.xml.

The following shows configuring the Oracle data source for two projects.

If your setup has multiple projects pointing to different data sources, then copy <local-tx-datasource> in ds-xml for each project data source.

```
<jndi-name>QueryToolDemoDS</jndi-name>
<connection-url>jdbc:oracle:thin:@localhost:1521:xe</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>i2b2demodata</user-name>
<password>i2b2demodata_password</password>
```

For example, project Demo2 would have this entry:

```
<jndi-name>QueryToolDemo2DS</jndi-name> with connection-url:
<connection-url>jdbc:oracle:thin:@localhost:1521:xe</connection-url>
<driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
<user-name>i2b2demodata2</user-name>
<password>i2b2demodata2_password</password>
```

14. Edit *etc/jboss/crc-jms-ds.xml* and set 'DefaultDS' JMS database connection properties If the data source is SQLserver, then comment the oracle section and uncomment sqlserver data source section.

```
<local-tx-datasource>
    <jndi-name>DefaultDS</jndi-name>
    <connection-url>jdbc:oracle:thin:@localhost:1521:xe</connection-url>
    <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
    <user-name>i2b2hive_uname</user-name>
    <password>i2b2hive_password</password>
    …
</local-tx-datasource>
```

2.1.6 Build and deploy edu.harvard.i2b2.crc and edu.harvard.i2b2.crc.loader

The following outlines the steps to build and deploy **edu.harvard.i2b2.crc** and **edu.harvard.i2b2.crc.loader**.

Steps:

1. Run *ant -f master_build.xml clean build-all deploy*

☞ **Note**: *If you are deploying on an existing JBoss instance with CRC 1.3, then please remove the QueryProcessor-EJB.jar from the $JBOSS_HOME/server/default/deploy directory.*

2.1.7 Start Services

2.1.7.1 START JBOSS

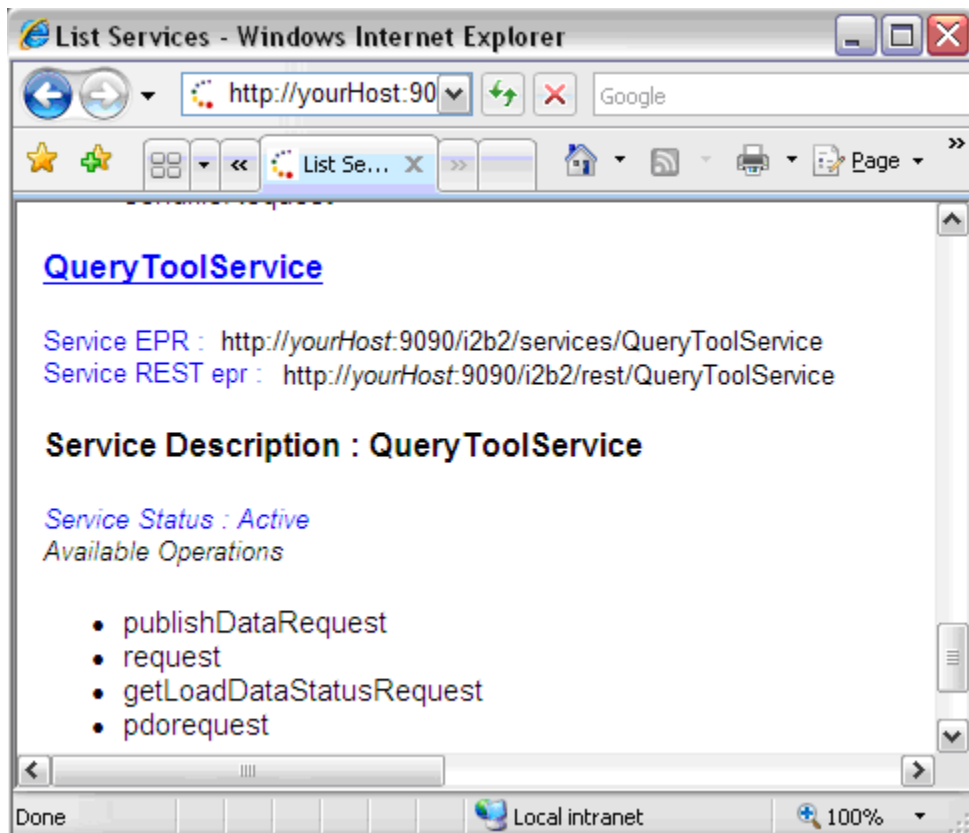To start JBOSS run the following: *$**JBOSS_HOME**/bin/run.sh -b 0.0.0.0*

2.1.7.2 WEB SERVICE

The following are the steps to take to verify the web service is running.

1. In a browser, check the following URL:

    *http://**yourHost**:9090/i2b2/services/listServices*

2. Verify QueryToolService is listed as active.

2.2   Changing Server log level

By default JBOSS log will be in **DEBUG** mode, changing it to **INFO** mode will increase server performance.

1. Edit *$JBOSS_HOME/server/default/conf/jboss-log4j.xml* file and *add* the '*Threshold*' *<param>*

```
<appender name="FILE" class="org.jboss.logging.appender.DailyRollingFileAppender">

    <errorHandler class="org.jboss.logging.util.OnlyOnceErrorHandler"/>

    <param name="File" value="${jboss.server.home.dir}/log/server.log"/>

    <param name="Append" value="false"/>

    <param name="Threshold" value="INFO"/>

    <!-- Rollover at midnight each day -->

    <param name="DatePattern" value="'.'yyyy-MM-dd"/>
```

```
<layout class="org.apache.log4j.PatternLayout">

    <!-- The default pattern: Date Priority [Category] Message\n -->

    <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>

    . . .

</layout>

</appender>
```

2. To switch back to DEBUG mode, *comment out* the 'Threshold' <param> and wait a minute.

☞ ***Note****: There is no need to restart JBOSS.*

3. VERIFY INSTALLATION

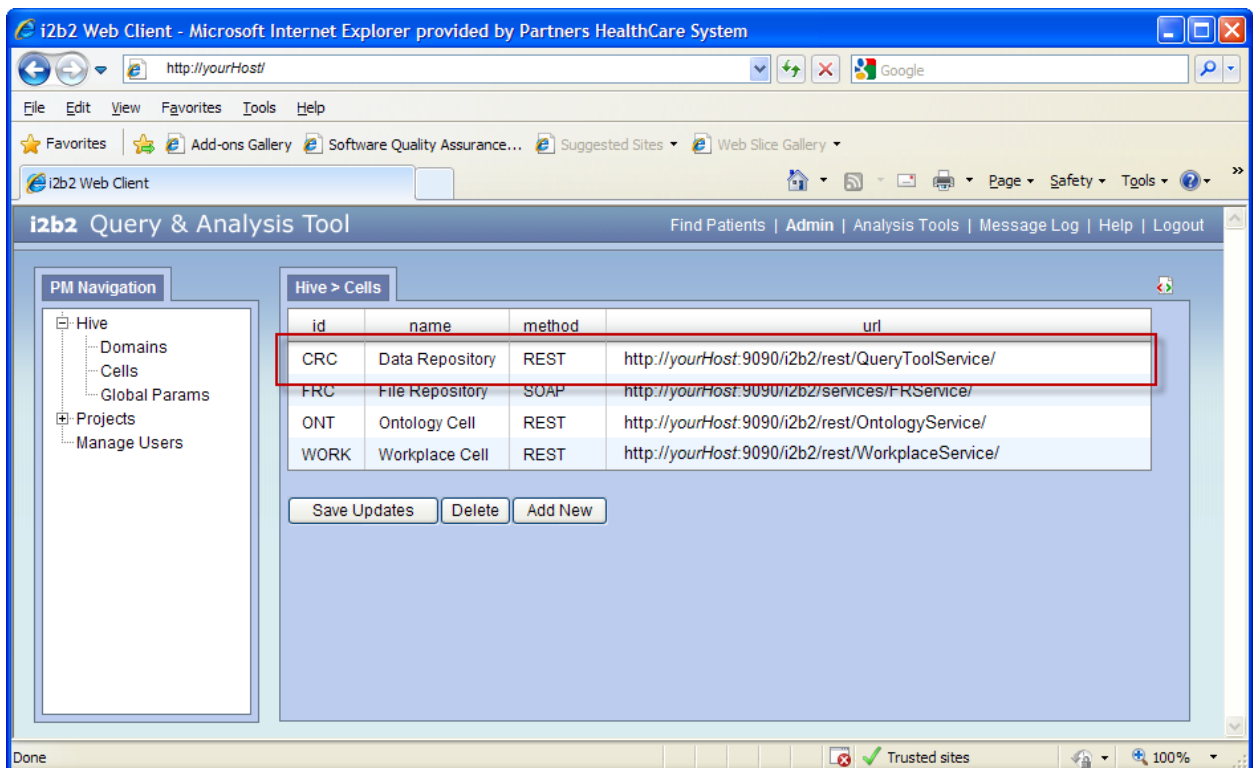## 3.1 CRC Cell Sanity Test via the i2b2Workbench

3.1.1 Configure the i2b2Workbench to communicate with your CRC cell.

Cell configuration is addressed in the Project Management (PM) Cell installation and set up. Please refer to this document if the CRC Cell has not yet been configured.

To verify the data, go to the web client site *http://host/webclient*.

Once logged on, select "*Cells*" under the "*Hive*" primary navigation tab. If the CRC Cell has been configured it will be listed as an existing, registered cell:
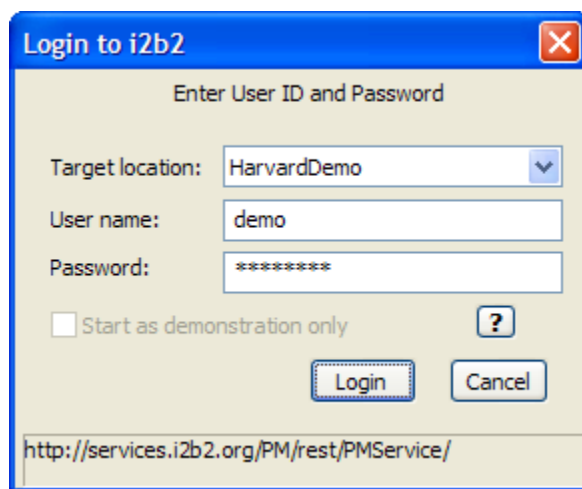
- To verify cell data, click on the cell name (CRC).

- To edit the cell data, click on the field you want to edit. Click on "*Save Updates*" to save the changes you have made.

3.1.2 Launch the i2b2Workbench

The following outlines the steps to log into the i2b2 Workbench.

1. Go to the location (folder) in which the i2b2 Workbench was installed.

2. Double click on **i2b2Workbench.exe**.
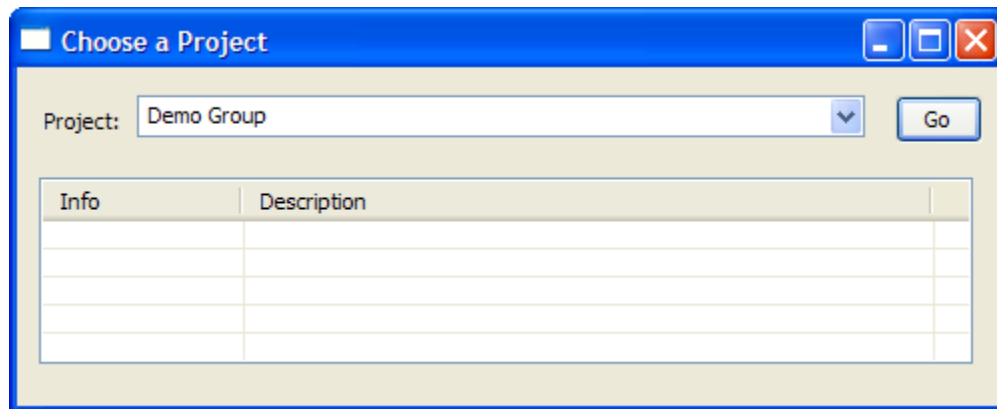
3. The **Login dialog box** will open.

4. Select your **target location** (YourSite)

5. Enter you **User name** (ID) and **Password**. (demo/demouser)

6. The *URL at the bottom of the login screen* should be the address of your PM cell. If it isn't then edit the *i2b2Workbench.properties* file to point to the correct location of your PM cell.

   i2b2.1=**YourSite**,REST,http://*jbossHost*:port/i2b2/rest/PMService/
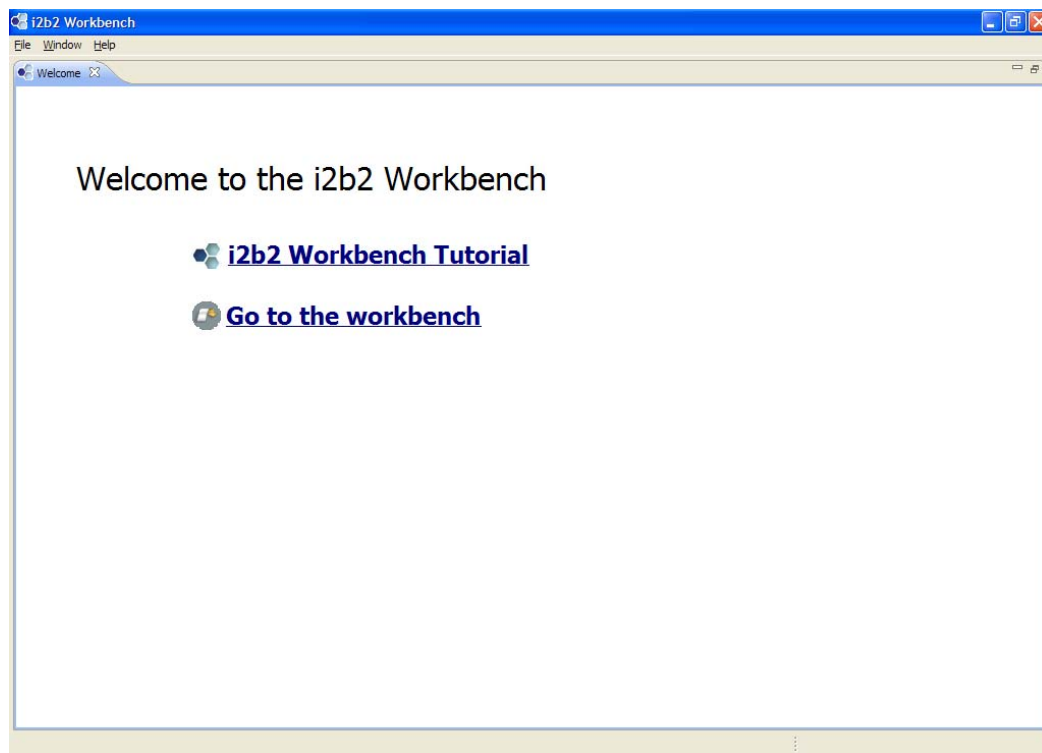
7. The **Choose a Project** dialog box will open.

☞ *Note: 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.*
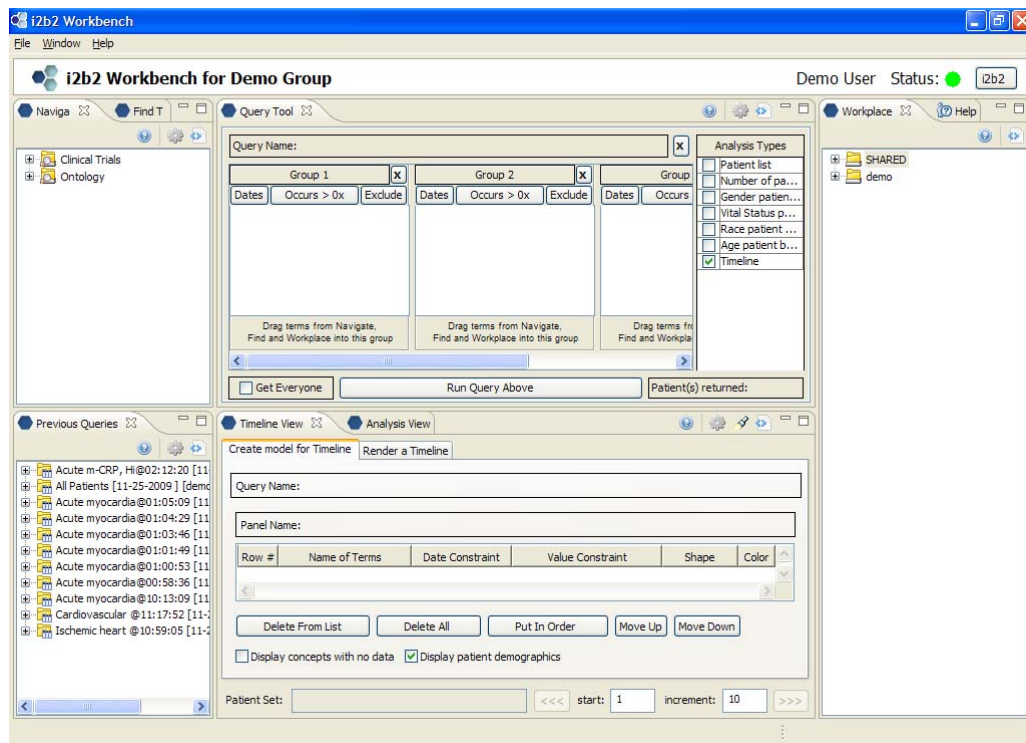
8. Select the project you want to log into.

9. The **Welcome** page will open.

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

10. Click on Go to the workbench.

11. The **desktop** of the i2b2 Workbench will open.

3.1.3 Possible problems

One of the following error messages may appear in view window.

1. CRC cell is unavailable

    The possible cause for this error message may be one of the following:

    ▪ CRC cell address in the PM webclient is incorrect.

a. See the section labeled "*Configure the i2b2Workbench to communicate with your CRC Cell*" (section 3.1.1) to verify the CRC cell address is correct.

- CRC cell may be down. Follow these steps to check the status of the CRC cell.

    a. In a browser, check the url
       http://**yourHost**:9090/i2b2/services/listServices

    b. Verify that *QueryToolService* is listed as active.

2. Remote server is unavailable

   The possible cause for this error message may be one of the following:

   - Server may be down. Follow these steps to check the status of the server.

       a. In a browser, check the url
          http://**yourHost**:9090/i2b2/services/listServices

       b. Verify that *QueryToolService* is listed as active.

3. PM service is not responding

   The Project Management Cell is down or its address was not configured properly in the following sections:

    a. *Section*: Build edu.harvard.i2b2.crc.loader (section 2.1.4)
       Step 4: Edit the *etc/spring/edu.harvard.i2b2.crc.loader.properties* file

    b. *Section*: Setup edu.harvard.i2b2.crc (section 2.1.5)
       Step 4: Edit the etc/spring/crc.properties file

4. Database error

   The possible cause for this error message may be one of the following:

   - Problem connecting to the database.

   - Verify configuration parameters in the following sections:

       a. *Section*: Build edu.harvard.i2b2.crc.loader (section 2.1.4)

Step 5: Edit the *etc/spring/CRCLoaderApplicationContext.xml*

    b.   <u>*Section*</u>: Setup edu.harvard.i2b2.crc (section 2.1.5)

       Step 8a: Edit *etc/spring/CRCApplicationContext.xml*

       Step 8b: Update the entry in *etc/spring/crc.properties*

       Step 13: Configure Jboss Datasource

       Step 14: Edit *etc/jboss/crc-jms-ds.xml*

LICENSE

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