ACT Sites: i2b2 upgrade from 1.7.09c to 1.7.12A

Upgrading i2b2 from version 1.7.09c to 1.7.12A

Version Information

Current Version:	1.7.12A
Release Date:	May 2020
License:	Mozilla 2 Open Source License

Download

Description	i2b2 Released Version	Minimum Version Required	Download Link	Requirements
Files to upgrade the i2b2 server to 1.7.12A	1.7.12A	1.7.09c	i2b2core-upgrade-1712a.zip	You are running at least i2b2 1.7.09c in the ACT production network. Download the zip file to the left, and follow the instructions below to upgrade to i2b2 1.7.12A.

Notes

Release 1.7.12 contains changes to the i2b2 core Server, database and Web Client.

- The upgrade process is now made easier using a pre-built war file and the properties files are now stored in the database.
- The prebuilt-war file has all the core, web and database files packaged together and available as single download zip file.
- **NEW:** For further troubleshooting issues, please refer to the Troubleshooting Tips for i2b2 & SHRINE page

Upgrade Instructions

Below are Step-by-Step Instructions after you have upgraded to Wildfly.

Example Paths to i2b2.war

WildFly 10 path
`/opt/wildfly-10.0.0.Final/standalone/deployments/i2b2.war/`

Step-by-Step Instructions for WildFly

In the following instructions the variable { version } refers to your version of Wildfly installed. We have tested with Wildfly 10, 14, and 17.

1

Stop WildFly

Linux Example

```
$ /opt/wildfly-{version}.Final/bin/jboss-cli.sh --connect command=:shutdown
```

2

Backup your existing **i2b2.war directory**

Linux

On **Linux**, move the */opt/wildfly-{version}.Final/standalone/deployments/* and save it to a different location outside of */opt/wildfly-{version}.Final/*

Windows

On **Windows**, move the *c:\opt\wildfly-{version}.Final\standalone\deployments* and save it to a different location outside of *c:\opt\wildfly-{version}.Final*

3

Extract the download file i2b2core-war-1712.zip to a folder outside of wildfly folder.

4

Copy all the files from the download war file **deployment** into your existing **WildFly** **deployment** directory (e.g. /opt/wildfly /standalone/deployment/)

5

Upgrade i2b2 Databases to 1.7.12A

Run the following upgrade scripts on your i2b2 database instance, where {db} can be Oracle, sqlserver or postgresql. **Note:** The example scripts below are examples ONLY. The actual .sql scripts are provided in the zip file download above.

In **/i2b2/data/** folder

CRC	Open the file: /i2b2/data/Crcdata/scripts/crc_create_query_{db}.sql and run the last three relevant sections (upgrading from 1.7.09c all the way to 1.7.12): For example, for SQL server: --== -- Database Script to upgrade CRC from 1.7.09c to 1.7.10 --== -- New column added to support new SQL breakdowns - roles based access alter table QT_QUERY_RESULT_TYPE add USER_ROLE_CD VARCHAR(255); --== -- Database Script to upgrade CRC from 1.7.10 to 1.7.11 --== insert into QT_PRIVILEGE(PROTECTION_LABEL_CD, DATAPROT_CD, HIVEMGMT_CD) values ('SETFINDER_QRY_PROTECTED','DATA_PROT','USER') --== -- Database Script to upgrade CRC from 1.7.11 to 1.7.12 --== alter table QT_QUERY_RESULT_TYPE add CLASSNAME VARCHAR(200) ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientSetGenerator' where NAME='PATIENTSET' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultEncounterSetGenerator' where NAME='PATIENT_ENCOUNTER_SET' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientCountGenerator' where NAME='PATIENT_COUNT_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder.QueryResultGenerator' where NAME='PATIENT_GENDER_COUNT_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder.QueryResultGenerator' where NAME='PATIENT_VITALSTATUS_COUNT_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder.QueryResultGenerator' where NAME='PATIENT_RACE_COUNT_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder.QueryResultGenerator' where NAME='PATIENT_AGE_COUNT_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientSQLCountGenerator' where NAME='PATIENT_LOS_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientSQLCountGenerator' where NAME='PATIENT_TOP20MEDS_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientSQLCountGenerator' where NAME='PATIENT_TOP20DIAG_XML' ; update QT_QUERY_RESULT_TYPE set CLASSNAME='edu.harvard.i2b2.crc.dao.setfinder. QueryResultPatientSQLCountGenerator' where NAME='PATIENT_INOUT_XML' ;

Hive	Open the file: /i2b2/data/Hivedata/scripts/upgrade_{db}_i2b2hive_tables.sql and run the last required section:
	
	For example, for SQL server:
	
	`CREATE TABLE HIVE_CELL_PARAMS (`
	`ID INT NOT NULL,`
	`DATATYPE_CD VARCHAR(50) NULL,`
	`CELL_ID VARCHAR(50) NOT NULL,`
	`PARAM_NAME_CD VARCHAR(200) NOT NULL,`
	`VALUE VARCHAR(MAX) NULL,`
	`CHANGE_DATE DATETIME NULL,`
	`ENTRY_DATE DATETIME NULL,`
	`CHANGEBY_CHAR VARCHAR(50) NULL,`
	`STATUS_CD VARCHAR(50) NULL,`
	`CONSTRAINT HIVE_CE__PK PRIMARY KEY(ID)`
	`);`
	
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(33, 'T', 'CRC', 'queryprocessor.jndi.queryinfolocal', 'ejb.querytool.QueryInfoLocal', NULL,`
	`NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(31, 'T', 'CRC', 'queryprocessor.jndi.querymanagerlocal', 'ejb.querytool.QueryManagerLocal',`
	`NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(37, 'T', 'CRC', 'queryprocessor.jndi.querymanagerremote', 'ejb.querytool.QueryManager', NULL,`
	`NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(61, 'T', 'ONT', 'applicationName', 'Ontology Cell', NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(63, 'T', 'CRC', 'applicationName', 'CRC Cell', NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(62, 'T', 'ONT', 'applicationVersion', '1.7', NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(64, 'T', 'CRC', 'applicationVersion', '1.7', NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(16, 'T', 'CRC', 'edu.harvard.i2b2.crc.analysis.queue.large.jobcheck.timemills', '60000',`
	`NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(14, 'T', 'CRC', 'edu.harvard.i2b2.crc.analysis.queue.large.maxjobcount', '1', NULL, NULL,`
	`NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(13, 'T', 'CRC', 'edu.harvard.i2b2.crc.analysis.queue.large.timeoutmills', '43200000', NULL,`
	`NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(15, 'T', 'CRC', 'edu.harvard.i2b2.crc.analysis.queue.medium.jobcheck.timemills', '60000',`
	`NULL, NULL, NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`VALUES(12, 'T', 'CRC', 'edu.harvard.i2b2.crc.analysis.queue.medium.maxjobcount', '4', NULL, NULL,`
	`NULL, 'A');`
	`INSERT INTO HIVE_CELL_PARAMS(ID, DATATYPE_CD, CELL_ID, PARAM_NAME_CD, VALUE, CHANGE_DATE,`
	`ENTRY_DATE, CHANGEBY_CHAR, STATUS_CD)`
	`... (shortened)`

Ontology	Open the file: /i2b2/data/Metadata/scripts/ont_{db}.sql and run the script: For example, for SQL server: alter table TABLE_ACCESS add C_ONTOLOGY_PROTECTION VARCHAR(MAX); **Note:** For ACT, you'll want to apply this statement to *both* your local i2b2 ACT ontology, and your SHRINE ACT ontology (if you use two separate i2b2 projects).
PM	Open the file: /i2b2/data/Pmdata/scripts/upgrade_{db}_i2b2pm_tables.sql and run the last relevant section (upgrading from 1.7.09c): For example, for SQL server: ALTER TABLE PM_USER_LOGIN DROP PRIMARY KEY; CREATE INDEX PM_USER_LOGIN_IDX ON PM_USER_LOGIN(USER_ID, ENTRY_DATE);

Update the datasource (*-ds.xml) files in your **deployment directory**

6

Copy ***-ds.xml** files from the backup folder to **wildfly-1#.0.1.Final/standalone/deployments**

Edit the following files **crc-ds.xml**, **ont-ds.xml**, **pm-ds.xml**, **work-ds.xml**
and replace ALL the **<driver>{something}.jar</driver>** with

Oracle	SQL Server	PostgreSQL
<driver>ojdbc8.jar</driver>	<driver>mssql-jdbc-7.4.1.jre8.jar</driver>	<driver>postgresql42.2.8.jar</driver>

7

Migrate cell properties: In 1.7.12, cell properties have been moved to the database, in a new table called HIVE_CELL_PARAMS. If any cell properties were previously changed, they will need to be manually updated in the database. After this, the properties files can be deleted to prevent confusion.

More documentation on setting cell properties is available at this page. **Most commonly, the AGG_SERVICE_ACCOUNT password will need to be updated.** Generally, the cell URLs do not need to be configured anymore, as the hostname and port is now auto-detected.

Example
Properties files are stored at */opt/wildfly-{version}.Final/standalone/configuration/* on **Linux** and c:\opt\wildfly-{version}.Final\standalone\configuration\ on **Windows**. The table can be edited with a SQL editor in hive_cell_params.

Note: any changes made in your HIVE_CELL_PARAMS table will require you to restart Wildfly/i2b2 for the changes to take affect.

8

Backup your existing **i2b2_config_data.js** from your web server

On your webclient backup your existing **i2b2_config_data.js** to a folder outside of your web server.

Linux Example

On **Linux**, *move the folder /var/www/html/webclient/* and save it to a different location outside of */var/www/html/webclient*

Windows

On **Windows**, move the folder *c:\inetpub\webclient* and save it to a different location outside of *c:\inetpub\webclient*

9

Install the new webclient on your web server

Linux

On **Linux**, extract the i2b2webclient-1712.zip to the folder */var/www/html/webclient/* and copy the backup **i2b2_config_data.js** to */var/wwww/html/webclient*

Windows

On **Windows**, *extract the i2b2webclient-1712.zip c:\inetpub\webclient* and copy the backup **i2b2_config_data.js** to *c:\inetpub\webclient*

10

Start WildFly

Linux Example

```
$ /opt/wildfly-{version}.Final/bin/standalone.sh -b 0.0.0.0 &
```

11

Verify your upgrade

Test Web Client

Navigate to your local i2b2 Web Client in your preferred browser and verify you are able to log on and perform standard i2b2 functions. (e.g. Run queries, retrieve previous queries, view breakdowns, etc.)

Need help? If you run into any issues, please check out the Troubleshooting Tips for i2b2 & SHRINE page, and/or post your issues to the ACTtecdh mailing list