Informatics for Integrating Biology and the Bedside

User Guide

NCBO Extraction Tools

Document Version: 1.1.1

i2b2 Software Version: 1.1

Table of Contents

DOCUMENT MANAGEMENT

Revision Number	Date	Author	Description of change
1.0.1	08/03/11	Lori Phillips	Original document
1.0.2	11/04/11	Lori Phillips	Fixed typo in section 2.2 command
1.1.0	05/29/12	Lori Phillips	Version 1.1 updates
1.1.1	08/06/12	Lori Phillips	Root node clarification

ABSTRACT

This is a User's Guide for the NCBO Extraction Tools. This guide will help you run the NCBO Extraction tools and assist you in creating metadata files for ontologies originating from NCBO bioportal.

1. BEFORE YOU BEGIN

1.1 NCBO bioportal details

1.1.1 Community files

NCBO has set up a web site for posting extracted vocabularies. It is possible that someone in the i2b2 community has already extracted the vocabulary you are looking for. Please see section 4 for details.

1.1.2 Overview

Please read the document titled "NCBO Extraction Overview". It gives an overview of what the extraction tool does, how the information from NCBO bioportal is identified and how to set up for i2b2.

1.1.3 API KEY

This tool makes extensive use of the NCBO bioportal REST services. These services require an API key assigned from NCBO. If you do not already have one, login to BioPortal (http://bioportal.bioontology.org/login) and obtain one. Your key will appear in the 'Account' area.

1.1.4 ONTOLOGY ID

The ontology you wish to extract from NCBO bioportal has a unique acronym. The extraction tool refers to this as the 'ontology id'. The document titled "NCBO Extraction Overview" gives details on how to find this id.

As an example:

The ontology id for ICD-9 is ICD9CM.

The link http://bioportal.bioontology.org/ontologies/ICD9CM?p=classes shows a navigation tree for ICD-9.

☞ **Rule of thumb: If you cannot visualize the ontology within bioportal, you should not try to extract it.**

1.1.4.1 ROOT NODES

Unless you specify otherwise, the extraction tool will have the same number of root nodes as shown within bioportal. For example if you extract ontology id ICD0CM, you will end up with 4 root nodes in your final table as shown above. If you would prefer to have a single root node, e.g. "ICD-9" with these four categories as child nodes, be sure to specify the –rootNodeName parameter in section 2.3.2.

Some ontologies cannot be visualized in bioportal. As a rule these ontologies have no root nodes and as such will not render correctly using this tool.

1.2 Ontology Identification

1.2.1 Scheme/Prefix

Each distinct vocabulary and its associated codes is called a scheme. A distinction may be made between codes from different sources by pre-pending a unique prefix to each code. You need to determine what, if any, prefix you wish to prepend to the basecode provided by NCBO bioportal. Common choices for this prefix are the UMLS RSAB codes. (http://www.nlm.nih.gov/research/umls/knowledge_sources/ metathesaurus/release/source_vocabularies.html)

1.3 Limitations

1.3.1 Configurable concept path length

Some users are very concerned with the maximum concept path (c_fullname) length of their metadata. Others don't care. To support each user's different needs we have made the concept path length configurable. The concept_path is made up of terms from each level of the path: \Level 1 term\Level 2 term\ …. and so on.

We allow:

"As short as possible"   term at each level is reduced to a 4-char hash code. [S]

"Readable"   term at each level each term is reduced to 32-char   [R]

"Something in between" term at each level each term is reduced to 20-char   [M]

1.3.2 Ontology depth

It is necessary to be very aware of the depth (number of levels) of the ontology you wish to extract. This information is available on the bioportal metrics page for your ontology   Since the c_fullname overall length is limited; its important to note that the path length you choose is going to affect the ontology depth you can reasonably expect to support.

In general:

R[eadable]   will support a depth up to 20 levels.

M[edium]   will support a depth up to 33 levels.

S[hort]   will support over 33 levels.

The Short format is recommended for ontologies with 100,000 terms or more. While in theory, the Short format will support over 100 levels, we question the usability of an ontology with that many levels. If you decide to use the Short format be sure to read the note on page 12 about the possibility of duplicate non-synonymous c_fullname entries.

1.3.3 Total number of terms

Equally as important as depth is an ontology's total number of terms. To date, the largest ontology we have extracted contained ~100,000 terms. In this particular case, the resulting metadata file contained 5.7 million entries (terms have synonyms and can appear on several paths). For this reason, we recommend that you exercise caution or avoid extraction of ontologies with more than 100,000 terms.

If your desired ontology exceeds this limit of 100,000 terms, it is recommended that you look for or create a bioportal view that contains a subset of the entire ontology and then extract the subset. SNOMED-CT is a prime example of this problem. The metrics page for SNOMED-CT, http://bioportal.bioontology.org/ontologies/1353, lists over 390,000 terms for SNOMED-CT. Further down on this same page a set of views (or subsets of SNOMED) are shown.

In addition, when extracting large ontologies, we recommend that you utilize an ontology depth format (Section 1.3.2) of Short.

☞ **Rule of thumb: Do not extract ontologies with more than 100,000 terms.**

1.4　Software

1.4.1　Java JDK

JDK 6.0 or later 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*

1.4.2　Update Environment Variables

Be sure to set the JAVA_HOME, home directories you set up in the previous sections.

Example:

```
# Sample environment variables
JAVA_HOME=/opt/java/jdk1.6.0
export JAVA_HOME
```

2. INSTALLATION AND PREPARATION

2.1 Preparing the database

This package contains a folder called DatabaseScripts. Locate it now

➢ cd NCBOExtractionTools/DatabaseScripts

Scripts are provided for both oracle and sqlserver to create the staging and final tables used by this program. Open create_{sqlserver/oracle}_metadata_tables.txt and

    a. create staging table NCBO_STAGING

    b. create final metadata table NCBO_I2B2

       (You may rename it to something meaningful).

Rather than reusing these tables, you will probably want to create a different staging and final table for each ontology you extract.

2.2 Run the Extraction command line utility

This package contains a folder called Release_2_0. Locate it now

➢ cd NCBOExtractionTools/Release_2_0/

The files generated by this tool can be very large. It is recommended that output be directed to an area with large storage capacity. If this area exists in a directory separate from the location of the tool software, be sure that the user of the tool has write permission to the final output directory. In the example below we are extracting ICD-9 (id=ICD9CM). It is highly recommended that you include the ontology id in the name of the output file as it identifies the version of the ontology you have extracted.

➢ java –classpath endorsed_lib/*:genlib/i2b2Common-
   core.jar:lib/commons/*:lib/log4j/*:lib/jdbc/*:lib/spring/*:*
   edu.harvard.i2b2.ncbo.extraction.NCBOOntologyExtractAll -ont ICD9CM –apikey
   YOUR_API_KEY -outputFileName /YOUR_DIR/ICD9CM_stagingFile.txt &

where   -ont   ncbo ontology id of the ontology (version specific) you wish to extract (see section 1.1.3)

      -apikey   api key assigned to you from bioportal (see section 1.1.2)

      -outputFileName   full path file name of your destination staging file

      Make sure directory exists and that user has write permission to this directory.

Status will appear on screen …

    Extracting nodes to file: /YOUR_DIR/ICD9CM_stagingFile.txt

    INFO [main] (?:?) - NCBO Extraction tool Version 2.0 Feb 2014

    INFO [main] (?:?) – Obtaining NCBO root nodes

    INFO [main] (?:?) - Writing page: 1 of 448

    INFO [main] (?:?) - Writing page: 2 of 448

     …

    INFO [main] (?:?) - Extraction completed

The program obtains data via a web service call one page at a time. It is possible that you will experience network timeouts while the extraction is running. Any given call that times out will be retried twice; after that the program assumes that the network is down and ends the extraction program. The program defaults to a 300 second timeout. You may choose to increase the timeout length through an optional '-timeout' parameter. Timeouts are specified in milliseconds. A 500-second timeout would be specified as "-timeout 500000' on the Extraction tool command line.

Status appears on the screen and also gets logged to the file hierarchy.log. Please ensure that you receive the "Extraction completed" message before proceeding to the next step. When extraction is complete, the output file will contain data for the staging table in a '|'-delimited format. The first row of the file contains the column headings. Strings are quoted with the ' " ' identifier. Load this file into the staging table you created in section 2.1.

2.3   Run the Processing command line utility

Process the staging table content to produce the final table.

➢   cd NCBOExtractionTools/Release_1_1/

2.3.1 Configure database parameters

Configure the ExtractionApplicationContext.xml file to point to your staging table. This configuration shown below is for SQLServer .. Insert the driverClassName and url for your database here. Database drivers for sqlserver and oracle have been included in folder lib/jdbc/.

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource"
destroy-method="close">
    <propertyname="driverClassName"
            value="com.microsoft.sqlserver.jdbc.SQLServerDriver"/>
```

```
<property name="url" value="jdbc:sqlserver://your_db:port"/>

<property name="username" value="your u-name"/>

<property name="password" value="your password"/>

<property name="defaultAutoCommit" value="false"/>

<property name="defaultReadOnly" value="false"/>
```
    `</bean>`

Next, configure this for the schema that your staging table resides in and the staging table name. Example shown here is for sqlserver.

```
<bean id="database" class="edu.harvard.i2b2.ncbo.model.DBInfoType">

    <property name="db_fullSchema" value="i2b2metadata.dbo"/>

    <property name="stagingTable" value="NCBO_staging"/>

 </bean>
```

2.3.2   Run the staging table processing program

It is highly recommended that you include the ontology id in the name of the output file as it uniquely identifies the data you extracted from NCBO.

> java -classpath endorsed_lib/*:genlib/i2b2Common-
> core.jar:lib/commons/*:lib/log4j/*:lib/jdbc/*:lib/jdbc/sqlserver2005/*:lib/spring/*:*
> edu.harvard.i2b2.ncbo.extraction.NCBOOntologyProcessAll -outputFileName
> /YOUR_DIR/ncbo_ICD9CM_MED.txt -prefix YOUR_BASECODE_PREFIX   –
> pathFormat Medium –rootNodeName "YOUR ROOT NODE NAME"   &

  where -prefix is an optional parameter that specifies a prefix (eg ICD-9) to
        prepend to the basecode.(see section 1.2.1) If not specified, the basecodes
        will have no prefix.

    -pathFormat is an optional parameter that affects overall concept fullname
        length. (see sections 1.3.1 and 1.3.2) It may be set to:

      S[hort] results in shortest possible concept path length *(see Note on page 12)

       M[edium]   results in a somewhat readable concept_path

       R[eadable] results in a readable concept_path

       If not specified, the tool defaults to M[edium].

     -outputFileName is the full path of the final metadata file

        **Make sure directory exists and that user has write permission to this
        directory**

-rootNodeName [optional] is an optional singular root node whose name you specify. Often the data extracted from NCBO will have several root nodes. Our ICD-9 example has four. We recommend that you specify a single root node name. Names with more than one word should be quoted (e.g. "ICD-9 (NCBO)")

Status will appear on screen …

A single root node [has not] /[name of YOUR ROOT NODE NAME has] been specified.

Basecodes will [have no scheme prefix] [be prepended with YOUR_BASECODE_PREFIX:]

Processing data to file: /YOUR_DIR/ ncbo_ICD9CM_MED.txt

NCBO Processing tool   Version 2.0 Feb 2014

4 level 1 nodes found…

Writing category Diseases and injuries

…

Processing complete

When processing is complete, the output file will contain data for the final target metadata table in a '|' delimited format. The first row of the file contains the column headings. Strings are quoted with the ' " ' identifier. Load this file into the final table. Note that the output file will contain columns that are unnecessary for the target i2b2 metadata table. In addition, a report file is generated that lists all the settings used to create the output. In the example above, this file would be found at: /YOUR_DIR/ ncbo_ICD9CM_MED.report.

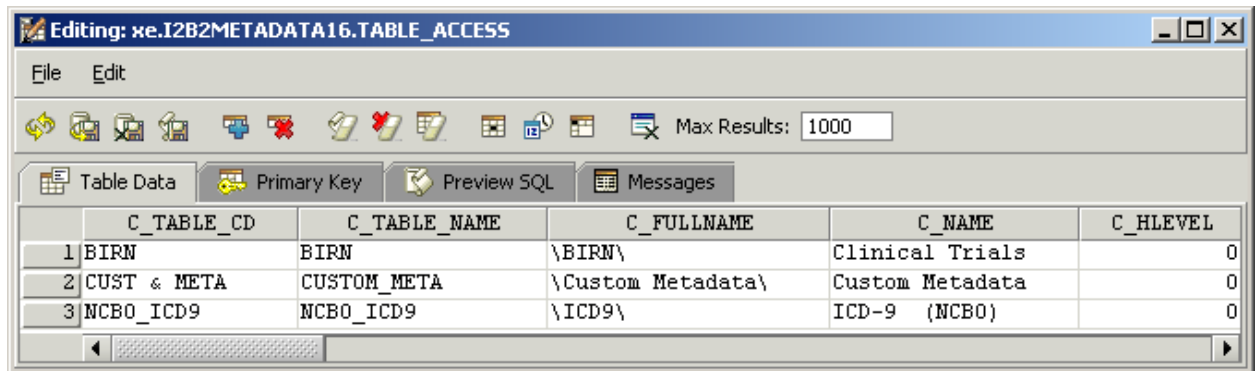⚠ *A note about pathFormat = S[hort]*

A pathFormat of S[hort] results in a 4-character symbol for each level of a c_fullname. While careful consideration was made in creating an algorithm that generates the 4-character symbol, it cannot absolutely guarantee a unique c_fullname for each term at a given level. We therefore recommend that you query for and manually edit duplicate non-synonymous c_fullname entries in your final metadata table.

```
select c_fullname, count(1) from ncbo_i2b2 where c_synonym_cd = 'N'
          group by c_fullname having count(1) > 1
```
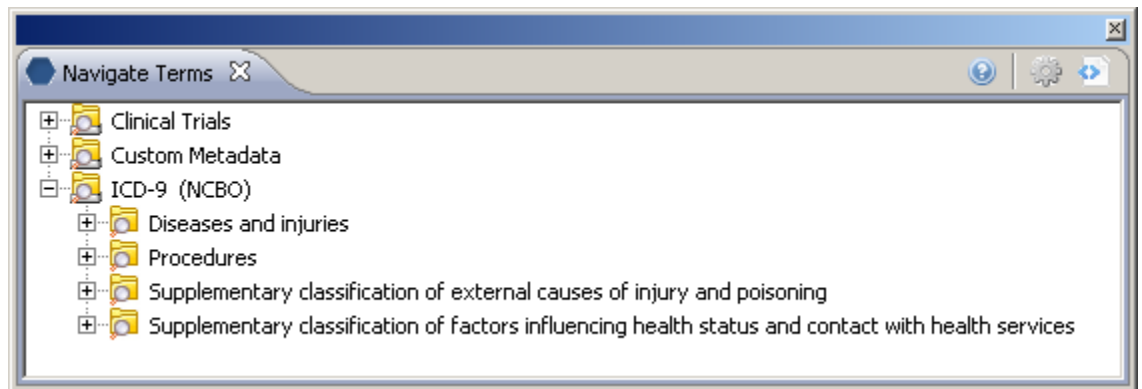
3. CONFIGURING I2B2 TO USE YOUR NEW METADATA

3.1 TABLE_ACCESS

Reconfigure table_access to include the root nodes of your new metadata. If a single root node was specified, it will appear in your final metadata table with c_hlevel = 0. If not, root nodes are entries in your final metadata table with c_hlevel = 1. There should be one entry per root node. The following example shows a configuration for for our ICD-9CM example. In this example our final target i2b2 metadata table was named 'NCBO_ICD9', as shown in c_table_name column.
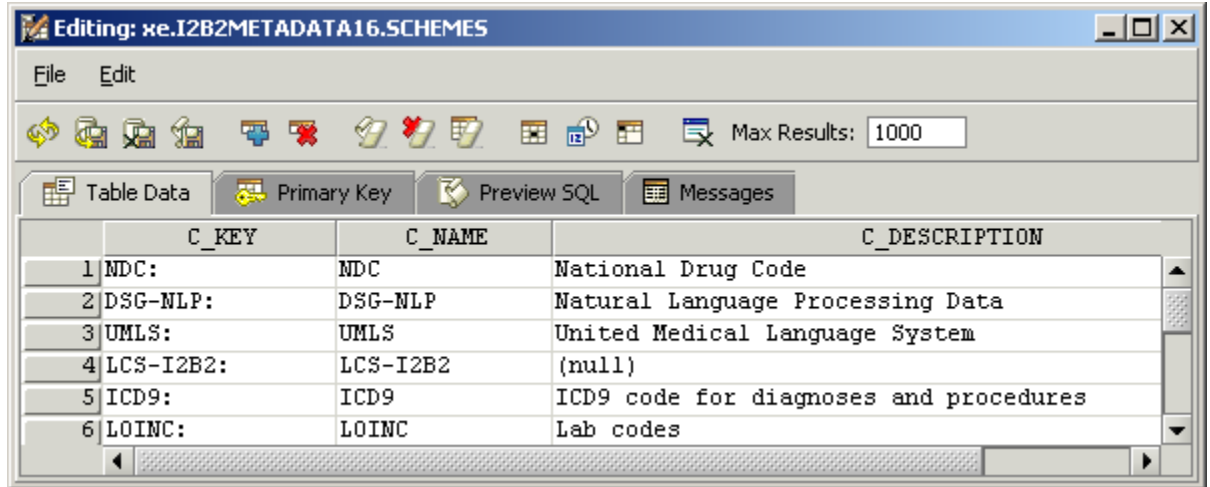

☞ **Note that in our example above we specified a rootNodeName of 'ICD-9 (NCBO)'**

3.2 SCHEMES

Reconfigure the SCHEMES table to include your new ontology's scheme or *prefix*. There should be one entry per scheme. The following example shows a configuration including the ICD9 prefix.
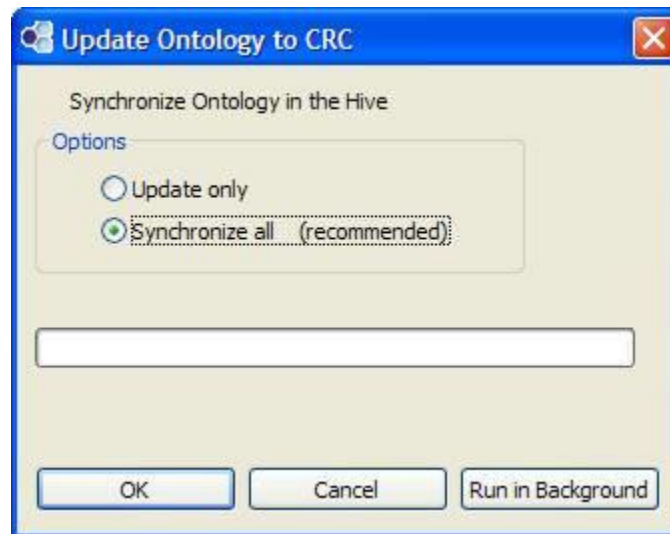
	C_KEY	C_NAME	C_DESCRIPTION
1	NDC:	NDC	National Drug Code
2	DSG-NLP:	DSG-NLP	Natural Language Processing Data
3	UMLS:	UMLS	United Medical Language System
4	LCS-I2B2:	LCS-I2B2	(null)
5	ICD9:	ICD9	ICD9 code for diagnoses and procedures
6	LOINC:	LOINC	Lab codes

3.3   CONCEPT_DIMENSION

You will need to synchronize your concept_dimension table so it contains the terms in your new metadata. Synchronization of metadata and concept_dimension is a feature found in the Edit View tool of the workbench. A user must have roles = EDITOR, ADMIN in order to perform the synchronization process.

1.   Click on the synchronize icon ()or () at the top of the view.

2.   The **Update Ontology window** will open.

3.   Click on the **OK button** to start the process.

4. CONTRIBUTE TO THE I2B2 COMMUNITY

4.1 Upload your final data file to NCBO

NCBO has provided a web site to upload your final metadata files for others to use: http://i2b2.bioontology.org/ . Note that the report output file is also uploaded to identify the parameter settings used to extract the ontology.

While not a requirement, it is highly recommended that the metadata file you upload contain the ontology_id /ontology_acronym in the file name as it uniquely identifies the ontology.