

PopMedNet-i2b2 Transform documentation - for SCILHS sites

Version 6.10 - 12/10/15

Jeff Klann, PhD

[Included Files](#)

[Tasks](#)

[Preparing the ontology for the transform](#)

[Preparing the PopMedNet database](#)

[Running the Annotated Data Dictionary Reporting Tool](#)

[Known Issues](#)

[Other notes](#)

[Changelog](#)

Included Files

ADD_SCILHS_100\MSSQL*: The SCILHS Annotated Data Dictionary Reporting Tool: A SQL Server version of a Python script to populate a summary spreadsheet of your PopMedNet tables that we will need to submit to PCORnet in January 2015.

NEW! - PMNTransform_v540\MSSQL\PCORNetLoader_v6.1.sql: SQL Server versions of a script to re-create PopMedNet tables, prepare the transform, and load the stored procedures. Several bug fixes and speed optimizations (see changelog).

PMNTransform_v540\MSSQL\run_v61_ms.sql: Script that selects all patients with at least one fact since 1/1/2010 and transforms them into PopMedNet tables. (No longer restricted to loyalty cohort - these are flagged in the Enrollment table.) This is the same as v6 except the stored procedures are listed out here for easy editing.

Note: Oracle files are provided for v6.0 adapted by Wake Forest are included. This has not been tested by SCILHS Central and does not have the v6.1 speed optimizations at this time.

PMNTransform_v500\Oracle\pmn_v6_Ora_general.sql: Oracle version of V6.0 with prescribing/dispensing fixes (untested). Does not have speed optimizations made in 6.1. See the changelog for details.

PopMedNet-i2b2 Transform Documentation.pdf: This document describing the transform and the process of loading it.

Tasks

Preparing the ontology for the transform

1. You must be running at least **v2.0.2** of our CDM ontology for the transform to work.
2. You have probably performed the following steps when preparing for previous versions of the transform. If not, please check the following.
 - a. Remember, If you used the Mapping tool, you must use the script to provided in the prepare_XXX_mappings to fix your pcori_basecode column after. This might also be necessary if you manually added child nodes. In this case, please verify that the nodes you added have the parent's pcori_basecode.
 - b. The transform requires that labs are mapped to or are children of a specific LOINC code, not a lab category. If a matching LOINC code does not exist in our ontology, remember that you are supposed to add it and inform us.
3. **NEW!** You must run the PCORI_MEDS_SCHEMA_CHANGE script on the database with your mapped pcornet_med ontology. This will add a pcori_cui and pcori_ndc code based on your mapping.
4. Back up all of your pcornet ontology tables (if you have changed them since the last backup).

Preparing the PopMedNet database

Once you have run v6 of the transform, you can skip this section when you want to refresh your data (re-writing your old PopMedNet data), *unless* you have changed your demographics mapping. If you have, you will need to re-run the PCORNetLoader script. Start at step 4.

1. Create a database for your PopMedNet tables. Our transform requires that they be on the same server as your i2b2 instance - though you can move them later.
2. Make sure the database user that will be running the scripts can also read from your i2b2 data and ontology databases. Some Oracle code that changes the permission is below:

```
-- First change "username" to the execution account username and run this
in the i2b2metadata account:
connect i2b2metadata/demouser@xe
grant all on pcornet_demo to username;
grant all on pcornet_diag to username;
grant all on pcornet_proc to username;
grant all on pcornet_enroll to username;
grant all on pcornet_vital to username;
grant all on pcornet_enc to username;

connect i2b2demodata/demouser@xe
-- AND run this in the i2b2Demodata account:
grant all on observation_Fact to username;
```

```
grant all on patient_dimension to username;  
grant all on visit_dimension to username;
```

3. If you are upgrading, you might need to manually delete script-generated views. If you do not, you will get errors in step 4. In this case, go back and delete the views and rerun step 4. Contact us if this occurs so we can develop a version that does not require this manual step.
4. From your new PopMedNet database, open the PCORNetLoader_v6.sql script. Note that the entire transform (including create tables) is now in this single script. Edit the preamble to match your local setup, according to the instructions in the file. This now includes the following:
 - synonym names
 - the USE line
 - datamart parameters that will go in the Harvest table
 - pointer to loyalty_cohort_summary
 - (loyalty cohort date range - you will probably not need to change this - by the time the range has changed we will release a new version that does not require manual entry)
5. This script will delete your existing PopMedNet tables. If you do not want this behavior, please back them up.
6. Run the script.
7. Verify that you have 15 tables (pmndemographic, pmndiagnosis, pmnencounter, pmnenrollment, pmnprocedure, pmnvital, pmnlabresults_cm, pmndispensing, pmncondition, pmnprescribing, pmndeath, pmndeath_cause, pmnharvest, pmnpcornet_trial, pmnpro_cm), that there were no errors, and that the pcornet_codelist table has some data (demographics code lists).

Running the transform

1. Separately from the above steps, be sure you have run the loyalty cohort script on your i2b2 database (this is distributed separately). If your data has not changed since running it previously and you have not deleted its tables, you do not need to run it again.
2. Load the run_v6.sql script and prepare to run it on your PopMedNet database. Modify the database names marked by comments. Run the script. It could take several hours or days to complete (approximately 1 day per 500k patients). When first setting up the process, try testing on a subset of data by changing the "top 100000000" in the script to a smaller number. When finished, the only errors should be in some of the 'DROP' commands at the beginning of the script - these are expected. There are also some warnings being issued about key constraints being violated on demographics. This seems to be an issue in mapping but it generally does not cause problems. If this does occur, please send us an error report. However, it is not a problem at this point. You will also see some warnings about set aggregation - these are not errors and are expected.
3. Please look at the i2pReport table to verify that data was transferred to PopMedNet. The 'difference' column should be 0 (or null), and the 'in popmednet' column should be large

(likely >1 million for all but patients and enrollment). NULL is expected in the i2b2 column for some entries.

4. **Send us the data in the i2pReport table as verification that you have run the transform.**
5. **Troubleshooting:** Note that the transform expects mappings in your ontology according to our instructions in the ontology documentation. If you run into problems, look there first. Mapping problems are the most likely culprit. For example, check that all of your local children are children of valid observation types in the ontology (for example, LOINC codes in labs, not lab groupers) and that pcori_basecode is set for all added rows.

Running the Annotated Data Dictionary Reporting Tool

Once refreshes begin occurring, you will be required to run this at every refresh. At this time, it is not required.

This section describes how to use the Annotated Data Dictionary Reporting Tool to automatically populate the summary spreadsheet on the data in your PopMedNet tables. PCORI is requiring us to submit these by early January. If you choose not to use this tool, you can run the queries in ETL_DICT_Queries by hand. Note that this tool only runs on SQLServer, but adapting for Oracle should not be too difficult. See the note below.

Instructions:

1. Install python and pypi on your machine.
2. Run `pip install -r requirements.txt`
3. Edit `config.ini`. If you are using Oracle, change `db_type` to ORACLE and change `sql_file` to point to the ORACLE version of the scripts (*). Change the host and sid for your environment.
4. Edit the environment variables you selected in `config.ini` with your username (or domain\username) and password. In Windows, the SET command does this.
5. If you are using Oracle, uncomment the line 'import cx_Oracle' in `query.py`. No other Python changes should be needed, but you will need to alter the SQL Script.
6. Run `python populate_spreadsheet.py config.ini --` hopefully this will run without error, which should take 5-10 minutes.
7. The Excel file that ends with `_SITE` should now contain your summary data of the PopMedNet tables. **Send that to us by 3/30/15.**

(*) Note: We have not modified the Oracle scripts to work with SCILHS. The Greater Plains Collaborative version is included for reference, but you will need to make changes. In particular, our table names begin with 'pmn', which will have to be appended, and our enrollment start date column has a different name.

Known Issues and Limitations

The following features have not been developed yet:

- Further testing and optimization of Oracle version.

- Code translation in Encounters (other than DRG and inout_cd) - v6 assumes all other visit_dimension columns use PCORI codes.
- Vitals unit conversions - v6 assumes the fact table uses the units PCORI is expecting.
- The Demographics transform assumes all data is in the patient dimension and there are no local children added below the PCORI leaf nodes.
- The lab transform does not:
 - translate units_cd into PCORI's expected units
 - verify qualitative results (tval_char) match the metadata
- The prescribing transform is slower than desired or expected at present.

Other issues:

- There is also a known issue somewhere in the demographics transform where a handful of patients get inserted twice. This only happens on a small subset of patients.
- Encounter and Demographic transforms assume that dimcodes for dimension tables are single-quoted, comma-separated lists. Unquoted strings are not supported.

Other notes

- If you have another method of creating PopMedNet databases or generating the annotated data dictionary, that is fine. You will still need to send us reports similar to the i2pReport table and the annotated data dictionary.
- "PopMedNet database" refers to the database in the PCORnet CDM physical data model which will be queried by PopMedNet.

Changelog

10/8/14 - initial release of preparatory files

10/20/14 - initial release of demographic transform

12/16/14 - release of "version 5" for SQL Server, which transforms all sections of CDM v1.5 except for unit conversions and code translation on visit_dimension columns.

3/19/15 - release of loyalty cohort transform script

8/20/15 release of "version 5.1" for SQL Server which transforms lab results.

10/10/15 - release of "version 5" for Oracle, adapted by Wake Forest. Limited to CDM v1.5.2.

10/15/15 - release of "version 6" for SQL Server, which transforms:

- Vitals 2.0.2: Adds smoking and tobacco status
- Enrollment: Now adds a flag for loyalty cohort patients
- Labs v2.0.2
- Medications v2.0.2 into prescribing and dispensing tables
- Condition: Uses the CONDITION_SOURCE modifier in Diagnoses 2.0.2
- Various optimizations and bugfixes noted in the code, plus the following changes requested by the coordinating center:
 - Enrollment date ranges are now based on encounters not facts
 - Discharge disposition: forced to NI for AV
 - Discharge status: forced to NI on AV

- Encounter columns like admit source now translate null to NI
- dx_source forced to FI for AV
- vital_source set to HC

12/10/15 - release of “version 6.1” for SQL Server and “version 6” for Oracle:

- MSSQL and Oracle: Bug fixes in prescribing/dispensing - please re-run
- MSSQL: Speed optimizations in prescribing, dispensing, labs, condition, and diagnosis. Now each transform runs on 2m patients at Partners in <30m.
- Note that the Oracle version was adapted by colleagues at Wake Forest and has not been tested by SCILHS Central.