Mapping with CAMP FHIR

# Welcome!

This file is meant to be more verbose documentation for running CAMP FHIR than the basic readme provides. You can choose to read straight through it—or, if you want to skim, you can go directly to the red headers (“Action items”), which contain just the critical information you’ll need to get moving.

# Step 1: Pick your source data model and map your codeset(s)

FHIR mapping involves two separate, but equally important steps: (1) mapping the variables in your source data to the corresponding variable in FHIR, and (2) mapping any enumerated value sets in your source data to acceptable FHIR values. In this step, we’re handling #2.

## What’s meant by “map a codeset”?

Because “enumerated value set” is a mouthful, we’ll refer to that concept as “codeset” from here on out. An example of a codeset is the set of values used to describe a patient’s ethnicity. In your source system, you may use “HISP” for Hispanic, “NHISP” for non-Hispanic, etc. You could put these local values directly in the FHIR structure, but you would lose out on the benefits of adopting FHIR’s preferred value set (“2135-2” for Hispanic, “2186-5” for non-Hispanic, etc.), which can be made consistent across organizations. For this reason, we start the CAMP FHIR process by creating a table in your source database that maps your local codesets to the preferred FHIR values.\*

\**Minor disclaimer*: The codeset mappings we provide in our PCORnet script are the subjective opinions of the informatics team at NC TraCS Institute. Though we have been careful and thoughtful in our mapping decisions, we absolutely encourage others to review our choices and make any changes that you see fit. Moreover, we welcome your feedback if you think we’ve got something wrong.

## **Action item #1: Create and populate your mapping table**

Log on to your source database and run scripts/1\_Create\_and\_populate\_mapping\_tbl\_PCORnet41.sql. This will create a table in your database called PCORNET\_FHIR\_MAPPING, and will populate the table with mappings from PCORnet 4.1 codesets to appropriate FHIR values.

**If your source system is not PCORnet**: Your source data model may not be PCORnet. That’s fine! That does mean that our mappings won’t do you much good, though. In this case, create the table, but just don’t run the INSERT statements. Instead, run different INSERT statements with your local mappings. *You may need to map concepts that we didn’t have to map*—for example, if you use a codeset for something that is not coded in PCORnet. It is fine to map new things—you’ll just want to account for these new mappings by adding new JOINS in the view creation scripts in the next step.

A couple caveats:

1. Our script assumes you’re using Oracle. If you’re using another RDBMS, you may need to tweak the syntax.
2. Our mappings are valid as of PCORnet 4.1. Future versions of PCORnet CDM may necessitate changes or additions.

Finally, here’s a dictionary for the table you’ll be creating:

|  |  |
| --- | --- |
| Column | Function |
| TABLE\_CD | The name of the source DB table containing the variable that uses this codeset |
| COLUMN\_CD | The name of the source DB column containing the variable that uses this codeset |
| LOCAL\_IN\_CD | The local code used to describe a member of this codeset (e.g., “M” for male). |
| FHIR\_OUT\_CD | The preferred FHIR value used to describe a member of this codeset (e.g., “male” for male) |
| FHIR\_OUT\_CHAR | Spelled-out (or display) version of the FHIR code. Optional. |
| COMMENTS | Any comments about this mapping |
| FHIR\_OUT\_COLUMN | The name of the target column in the view where this variable will ultimately end up |
| FHIR\_SYSTEM | URL for the documentation of the FHIR-preferred codeset |

# Step 2: Run (or modify and run) view creation scripts

Once you’ve got your codesets mapped, it’s time to create views of your data that CAMP FHIR will use as input. These views transform your data into a format that CAMP FHIR can easily work with, and also incorporate your new mappings where they exist.

Each view in our script represents a single FHIR resource. The current version of CAMP FHIR supports Patient, Encounter, Condition, Procedure, Observation, Medication Request, and Practitioner. More resources will be added over time.

The comment block above each view create statement details the exact FHIR element that each view column maps to. In order to compare the variables in our views against the FHIR documentation, open the appropriate FHIR resource page (e.g., *Patient* at <https://www.hl7.org/fhir/patient.html>), and follow the “chain” of elements detailed in the comment block. Note that we do not (yet) support all variables available in every FHIR resource. Over time, we expect to flesh out each of these domains to support many more variables.

We have provided view creation scripts for PCORnet 4.1 as a source system. Once again, Oracle is the assumed RDBMS.

**If your source system is i2b2**: We also have scripts on hand for i2b2, but as i2b2 is so highly variable between installations, we have opted not to provide our i2b2 scripts as part of this git repository. If you believe our scripts would be helpful to you in mapping i2b2, please feel free to email epfaff at email dot unc dot edu, and we’d be happy to chat.

**If your source system is something else**: All CAMP FHIR cares about is that the final views have the same structure as those created by our scripts. If you’d like to map from a totally different source system, just use our scripts as “inspiration,” and manually map from your source data to our view structure. In terms of the names of the views themselves—you may change the view names or column names, so long as you record your choices in table.hbm.xml (see Action item #3).

**Regarding size/performance:** If your database is nicely optimized, you could technically convert the data for all data in each view all at once. For us, we prefer to identify a cohort of patients first, and then inner join a table containing those patients to the views so that they just contain data for those patients. This cuts down on processing time and number of files output, particularly if you don’t really need every patient in your data source to be FHIR-ized.

## **Action item #2: Create your views**

Log on to your source database and run scripts/2\_Create\_CAMPFHIR\_views\_PCORnet41.sql. If you’re not using PCORnet as your source data model, read the text above to determine how to proceed.

# Step 3: Run CAMP FHIR to output FHIR files

## **Action item #3: Run CAMP FHIR from the command line**

1. Download the following files:
   1. CAMPFHIR.jar
   2. config.properties
   3. table.hbm.xml
2. Place config.properties and table.hbm.xml in a folder named /opt/ or C:\opt\ (for windows)
3. Change the database configuration setting in config.properties to point to your database.
4. Change the name of the views in table.hbm.xml to match the names you’ve given each of your views. For example, the default file contains the following name for our Condition view: <class name="Condition" table="**DEV\_CONDITION\_2FHIR**">. But, you are free to change this to: <class name="Condition" table="**MYAWESOMECONDITIONVIEW**">.
5. To run the application, enter the following command:

java –jar CAMPFHIR.jar <domain> <destination-folder> <record-partition-number>

* 1. Domain: Name of the FHIR domain. Example: Patient, MedicationRequest, etc. (Any of the domains in table.hbm.xml are options for this argument.)
  2. Destination-folder: folder where transformed data will be written.
  3. Record-partition-number: Result sets can be very large in size, which can require large quantities of heap space. To reduce this, you can partition your result set into chunks which are written in multiple files. The number you put here will be the number of chunks into which you partition your results—so, if you use 1000, your output for that domain will be chunked into 1000 files.