KNCV
TUBERCULOSISFOUNDATION

dhis2 PCF
People-Centred-Framework

DASHBOARD

TECHNICAL

DOCUMENTATION

2021

INTRODUCTION TO META
DATA PACKAGES

&

INSTALLING THE DHIS2 PCF
METADATA PACKAGE

Contents

1. Introduction to the TB metadata package

The KNCV TB metadata package consolidates the gold standard DHIS2 system designs into an installable file that can be implemented and customized within a new or existing DHIS2 instance.

"Metadata" is data that gives information about other data. Within DHIS2, metadata provides structural information that is used to classify data and define the configuration. This includes information on where, when and what data are collected, categorization and grouping of data elements, structure of forms and workflows, and analytical outputs such as indicators, charts, maps, and dashboards. Metadata in DHIS2 are fully customizable, which allows the platform to be used in a wide variety of contexts as each system owner can customize DHIS2 to meet their needs.

Note: The package has been developed under the 2.35 DHIS2 version. Using a latest version could impact the use of the package.

Package description

The KNCV TB metadata package is a downloadable, installable metadata (.json) files that can be adapted and customized according to country requirements. Metadata package can be downloaded and installed on a DHIS2 instance.
This package uses the aggregate DHIS2 data model. In addition to analytical outputs in the Dashboard / Analytics packages, Aggregate packages provide a standardized design for data inputs (data sets, data elements and category combinations) to ensure completeness and precision of systematic routine data capture.

PCF Datasets

The TB package contains a total of **6 data sets and 5 dashboards**, as described in tables 1& 2 below.

Table 1: Datasets

Data set name	Period	Comments
General TB Profile	Yearly	Data categorised as Epidemiology, People and System data
People not accessing the health system	Yearly	
People with TB diagnosed and notified but not successfully treated	Yearly	
People with TB seeking care either not diagnosed or notified	Yearly	
UNHLM achievements	Yearly	Enter annual country achievement
UNHLM targets	Yearly	Enter annual country targets

Table 2: Dashboards

Dashboard name	Comments
Summary PPA	Summary of the patient pathway analysis
Care Continuum 1 (CC1): People not accessing the health system	• People with TB infection who are at a high risk for disease • People with asymptomatic disease but not seeking care • People with symptomatic disease also not seeking care
Care Continuum 2 (CC2): People with TB seeking care either not diagnosed or notified	• People presenting to health facilities but not diagnosed with TB • People diagnosed by NTP but not notified • People diagnosed by non-NTP but not notified
Care Continuum 3 (CC3): People with TB diagnosed and notified but not successfully treated	• People diagnosed but not started on treatment • People notified in the system but not successfully treated • People successfully treated but not relapse free
UN HLM targets & achievements	Country achievements towards UN HLM yearly targets

PCF Dataset Structure

All datasets have the same structure with data categorized into 3 types and these are set up as tabs for each data sets:

I. Epidemiology data

II. People data

III. Systems data

Figure 1: Data types in a dataset

Data sets characteristics are structured in similar format:

- Each data type tab has a list of indicators that allows values to be entered as numbers and/or percentages

- All indicators have an interval confidence that allows lower and upper limits per indicator – *these are optional and can be blank*

- Indicators are further disaggregated by age, gender, TB type, urban/rural where there apply

Figure 2: Data disaggregation

- The UN HLM indicators have targets and achievements and are based on the standard UN HLM country data. The values can easily be imported into the system.

UNHLM targets	
Filter in section	**Value**
Annual UNHLM key targets for diagnosis and treatment of TB	5900
Annual UNHLM key targets for TB Preventative therapy targets in PHLIV	15894
Annual UNHLM key targets for TB Preventive Treatment targets (LTBI treatments) for child contacts	3000
Annual UNHLM key targets treatment and diagnosis for Childhood (0 - 14 years) TB	500
Annual UNHLM key targets treatment and diagnosis for MDR-TB	5542

Figure 3: UN HLM targets

Naming convention

For easy metadata management, a strict naming convention is used to organize the data elements.

Table 3: Data element naming convention

Name	Type	Comments
EPI 1.x **PPL1.x** **SYS 1.x**	Prefix	EPI = Epidemiology tab, PPL = People tab SYS = System tab The number following the prefix EPI refers to the data set (CC1, CC2, CC3) **Where x = 0**: General TB profile **Where x = 1**: CC1->People not accessing the health system **Where x = 2**: CC2 -> People with TB seeking care either not diagnosed or notified **Where x = 3**: CC3->People with TB diagnosed and notified but not successfully treated
_Abs val	Suffix	Absolute value / number value
_Perc val	Suffix	Percentage value
_Low	Suffix	Confidence interval lower limit
-Up	Suffix	Confidence interval upper limit
_Sex	Suffix	Sex/gender disaggregation

_Age	Suffix	Age disaggregation

Examples of data element names:

PPL 2.5 *Location of Diagnosis: Proportion of people diagnosed at different sectors and levels of the heath system_Percentage value*

- Data set: CC2 - People with TB seeking care either not diagnosed or notified and
- Data tab: People
- Data value: Percentage

EPI 3.2 *Number of people diagnosed with TB in the general population_age*

- Data set: CC3 - People with TB diagnosed and notified but not successfully treated
- Data tab: Epidemiology
- Disaggregation: Age

Reference

The PCF Dashboard metadata can be accessed by clicking the links below:
- [General TB profile](#)
- [People not accessing the health system](#)
- [People with TB seeking care either not diagnosed or notified](#)
- [People with TB diagnosed and notified but not successfully treated](#)

2. Package installation

The installation of the package can be done in 3 steps:
I. Preparing the metadata file.
II. Importing the metadata file into DHIS2.
III. Adapting the program after being imported

It is recommended to first read through each section before starting the installation and configuration process in DHIS2. Sections that are not applicable have been identified, depending on if you are importing into a new instance of DHIS2 or a DHIS2 instance with metadata already present. The procedure outlined in this document should be tested in a test/staging environment before either being repeated or transferred to a production instance of DHIS2.

Requirements

An administrator user account on DHIS2 is required to install the module. The procedure outlined in this document should be tested in a test/staging environment before being performed on a production instance of DHIS2. Great care should be taken to ensure that the server itself and the DHIS2 application is well secured, to restrict access to the data being collected.

Note: To open and edit the Json files must use *a plain text editor,* not a word processor like Microsoft Word.

Preparing the metadata file

If the package will be installed on a new instance of DHIS2, you can skip this section and move to the next section '*Importing a metadata file into DHIS2'*. While not always necessary, it can often be advantageous to make certain modifications to the metadata file before importing it into DHIS2.

Default data dimension (if applicable)

In the early versions of DHIS2, the default data dimension UID was auto generated. Thus, while all DHIS2 instances have a default category option, data element category, category combination and category option combination, the UIDs of these defaults could be different. More recent DHIS2 versions have hardcoded UIDs for the default dimension, and these UIDs are used in the meta data configuration packages.

To avoid conflicts when importing the metadata, it is advisable to *search and replace the entire .json file for all occurrences of these default objects, replacing UIDs of the .json file with the UIDs of the database in which the file will be imported.*

Object	API call
Category	api/categories.json?filter=name:eq:default

Category Option	api/categoryOptions.json?filter=name:eq:default
Category Combos	api/categoryCombos.json?filter=name:eq:default
Category Option Combo	api/categoryOptionCombos.json?filter=name:eq:default

For example, you can search for the UID of the default Category Option Combos on the system the package will be imported into, with this api call:

api/categoryOptionCombos.json?filter=name:eq:default

Then do a "search and replace all occurrences" in the .json file, to update the UID of the default Category Option Combo. The UID of the default CoC in the package is **HllvX50cXC0.**

Each data set is assigned to an Organisation Unit (OU). In the package, the OU UID is **FsJ8fnaDBz0**. You must update this UID before importing the data. We advise you to import the package against the root Organisation Unit of the DHIS2 instance.

Importing metadata

The .json metadata file is imported into your DHIS2 instance through the Import/Export app of DHIS2. It is advisable to use the "dry run" feature to identify issues before attempting to do an actual import of the metadata. If the "dry run"/"validate" import works without error, attempt to import the metadata. In some cases, import conflicts or issues are not shown during the "dry run", but appear when the actual import is attempted. In this case, the import summary will list any errors that need to be resolved.

Handling imports conflicts

If importing into a new DHIS2 instance, you will not have to worry about import conflicts, as there no existing database to conflict with. Follow the instructions to import the metadata then proceed to the "Additional configuration" section.

There are several different conflicts that may occur. The most common is existing metadata objects in the configuration package with a name, shortname and/or code that already exists in the target database. There are a couple of alternative solutions to these problems, with pros and cons to each one. The more appropriate option is dependent on the type of object for which a conflict occurs. Note that for options 1 and 2, the modification can be as simple as adding a simple prefix/suffix to the object name, this minimizes any confusions with the existing naming convention used.

- **Option 1: Rename the existing object in the DHIS2 database for which a conflict occurred.**
  - The advantage of this approach is that there is no need to modify the .json file, as changes are instead done through the user interface of DHIS2. This is likely to be less error prone. It also means that the configuration package is not changed, which can be an advantage for example when training material and

documentation based on the configuration package reference the object name.

- **Option 2: Rename the object for which there is a conflict in the .json file. The advantage of this approach is that the existing DHIS2 metadata remains unchanged.**
  - o This can be a factor when there is training material or documentation such as SOPs of data dictionaries linked to the object in question, and it does not involve any risk of confusing users by modifying the metadata they are familiar with.

- **Option 3: modify the .json file to re-use existing metadata.**
  - o This is a more complicated approach. For example, in cases where an option set already exists for a certain concept (e.g., "sex"), that option set could be removed from the .json file and all references to its UID replaced with the corresponding option set already in the database.
  - o The big advantage of this (which is not limited to the cases where there is a direct import conflict) is to avoid creating duplicate metadata in the database. There are some key considerations to make when performing this type of modification. Note that this requires expert knowledge of the detailed metadata structure of DHIS2.