TB Case Surveillance Tracker Installation Guide { #tb-cs-tracker-installation }

Package Version 1.0.1

System default language: English

Available translations: French, Spanish, Portuguese

Overview

The package metadata json files contain a "package" component that provides technical details on package version and content. The files available in the current version of the package are listed below.

DHIS2.35

=== "Complete Package"

```
```json
"package": {
    "DHIS2Build": "35d663a",
    "DHIS2Version": "2.35.11",
    "code": "TBCS00",
    "description": "Case Surveillance",
    "lastUpdated": "20220119T174325",
    "locale": "en",
    "name": "TB_CS_TKR_1.0.1_DHIS2.35.11-en",
    "type": "TKR",
    "version": "1.0.1"
}
```
```

DHIS2.36

=== "Complete Package"

```
```json
"package": {
    "DHIS2Build": "5d136cb",
    "DHIS2Version": "2.36.6",
    "code": "TBCS00",
    "description": "TB Case Surveillance",
    "lastUpdated": "20220120T140039",
    "locale": "en",
    "name": "TB_CS_TKR_1.0.1_DHIS2.36.6-en",
    "type": "TKR",
```
```

```
    "version": "1.0.1"
}
```

When importing this package into a new/blank instance, please refer to the DHIS2 installation guide.

Installation

Installation of the module consists of several steps:

1. Preparing the metadata file.
2. Importing the metadata file into DHIS2.
3. Configuring the imported metadata.
4. 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

In order to install the module, an administrator user account on DHIS2 is required. 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. Details on securing a DHIS2 system is outside the scope of this document, and we refer to the DHIS2 documentation.

Preparing the metadata file

NOTE: If you are installing the package on a new instance of DHIS2, you can skip the “Preparing the metadata file” section and move immediately to the 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

In early versions of DHIS2, the UIDs of the default data dimensions were 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 can be different. Later versions of DHIS2 have hardcoded UIDs for the default dimension, and these UIDs are used in the 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 from the instance in

which the file will be imported. Table 1 shows the UIDs which should be replaced, as well as the API endpoints to identify the existing UIDs

Object	UID	API endpoint
Category	GLevLNI9wkl	../api/categories.json?filter=name:eq:default
Category option	xYerKDKCefk	../api/categoryOptions.json?filter=name:eq:default
Category combination	bjDvmb4bfuf	../api/categoryCombos.json?filter=name:eq:default
Category option combination	HllvX50cXC0	../api/categoryOptionCombos.json?filter=name:eq:default

Identify the UIDs of the default dimesions in your instance using the listed API requests and replace the UIDs in the json file with the UIDs from the instance.

> **NOTE**
>
> Note that this search and replace operation must be done with a plain text editor, not a word processor like Microsoft Word.

Indicator types

Indicator type is another type of object that can create import conflict because certain names are used in different DHIS2 databases (.e.g "Percentage"). Since Indicator types are defined by their factor (including 1 for "numerator only" indicators), they are unambiguous and can be replaced through a search and replace of the UIDs. This method helps avoid potential import conflicts, and prevents the implementer from creating duplicate indicator types. The table below contains the UIDs which could be replaced, as well as the API endpoints to identify the existing UIDs:

Object	UID	API endpoint
Numerator only (number)	CqNPn5KzksS	../api/indicatorTypes.json?filter=number:eq:true&filter=factor:eq:1

Tracked Entity Type

Like indicator types, you may have already existing tracked entity types in your DHIS2 database. The references to the tracked entity type should be changed to reflect what is in your system so you do not create duplicates. The table below contains the UIDs which could be replaced, as well as the API endpoints to identify the existing UIDs:

Object	UID	API endpoint
Person	MCPQUTHX1Ze	../api/trackedEntityTypes.json?filter=name:eq:Person

Sort order for options

Check whether the sort order `sortOrder` of options in your system matches the sort order of options included in the metadata package. This only applies when the json file and the target instance contain options and option sets with the same UID.

After import, make sure that the sort order for options within an option set starts at 1. There should be no gaps (eg. 1,2,3,5,6) in the sort order values.

Sort order can be adjusted in the Maintenance app.

1. Go to the applicable Option Set
2. Open the "Options" section
3. Use "SORT BY NAME", "SORT BY CODE/VALUE" or "SORT MANUALLY" alternatives.

Dashboards and visualizations

The dashboards and visualizations in each dashboard cannot be hidden/unhidden by constants. Therefore the unapplicable dashboards/visualizations have to be manually removed during package installation based on the selection of tests/drugs.

Visualizations in the metadata file may contain a placeholder <OU_ROOT_UID>. This placeholder in the metadata.json file has to be replaced by the Root Organisation unit UID from the target instance before the package can be imported.

Some visualizations and maps may contain references to organisation unit levels. Maps that consist of several map views may contain vaious Organisation unit level references based on the configuration of the map layer. The table below provides an overview of such items included in the package. Adjust the organisation unit level references in the metadata json file to match the organisation unit structure in the target instance before importing the metadata file.

Example:

UID	Name	Type	Organisation unit levels	Mapping guidance
`N6MAiuCeFF0`	TB case notifications by facility (all cases, all forms)	map (mapview)	1,2,3	Country, Region, District
`N6MAiuCeFF0`	TB case notifications by facility (all cases, all forms)	map (mapview)	4	Facility

Importing metadata

Use Import/Export DHIS2 app to import metadata packages. It is advisable to use the "dry run" feature to identify issues before attempting to do an actual import of the metadata. If "dry run" reports any issues or conflicts, see the import conflicts section below. If the "dry run"/"validate" import works without error, attempt to import the metadata. If the import succeeds without any errors, you can proceed to configuring the module. 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 import conflicts

> **NOTE**
>
> If you are importing the package into a new DHIS2 instance, you will not experience import conflicts, as there is no metadata in the target database. After import the metadata, proceed to the "Configuration" section.

There are a number of different conflicts that may occur, though the most common is that there are metadata objects in the configuration package with a name, shortname and/or code that already exist in the target database. There are a couple of alternative solutions to these problems, with different advantages and disadvantages. Which one is more appropriate will depend, for example, on the type of object for which a conflict occurs.

Alternative 1

Rename the existing object in your DHIS2 database for which there is a conflict. 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 left as is, which can be an advantage for example when updates to the package are released. The original package objects are also often referenced in training materials and documentation.

Alternative 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 is left as-is. 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.

Note that for both alternative 1 and 2, the modification can be as simple as adding a small pre/post-fix to the name, to minimise the risk of confusion.

Alternative 3

A third and more complicated approach is to modify the .json file to re-use existing metadata. 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. 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:

- it requires expert knowledge of the detailed metadata structure of DHIS2
- the approach does not work for all types of objects. In particular, certain types of objects have dependencies which are complicated to solve in this way, for example related to disaggregations.
- future updates to the configuration package will be complicated.

Additional configuration

Once all metadata has been successfully imported, there are a few steps that need to be taken before the module is functional.

Sharing

First, you will have to use the *Sharing* functionality of DHIS2 to configure which users (user groups) should see the metadata and data associated with the programme as well as who can register/enter data into the program. By default, sharing has been configured for the following:

- Tracked entity type
- Program
- Program stages
- Dashboards

There are four user groups that come with the package:

- TB admin
- TB access
- TB data capture
- TB lab data capture

By default the following is assigned to these user groups

Object	**User Groups**			
	TB access	TB admin	TB data capture	TB lab data capture
Tracked entity type	**Metadata:** can view **Data:** can view	**Metadata:** can edit and view **Data:** can view	**Metadata:** can view **Data:** can capture and view	**Metadata:** can view **Data:** can capture and view
Program	**Metadata:** can view **Data:** can view	**Metadata:** can edit and view **Data:** can view	**Metadata:** can view **Data:** can capture and view	**Metadata:** can view **Data:** can capture and view
Program Stages	**Metadata:** can view **Data:** can view	**Metadata:** can edit and view **Data:** can view	**Metadata:** can view **Data:** can capture and view	**Group access is limited to stages: TB Registration and Laboratory Results** **Metadata:** can view **Data:** can capture and view
Dashboards	**Metadata:** can view **Data:** can view	**Metadata:** can edit and view **Data:** can view	**Metadata:** can view **Data:** can view	**Metadata:** can view **Data:** can view

You will want to assign your users to the appropriate user group based on their role within the system. You may want to enable sharing for other objects in the package depending on your set up. Refer to the DHIS2 Documentation for more information on configuring sharing.

User Roles

Users will need user roles in order to engage with the various applications within DHIS2. The following minimum roles are recommended:

1. Tracker data analysis : Can see event analytics and access dashboards, event reports, event visualizer, data visualizer, pivot tables, reports and maps.
2. Tracker data capture : Can add data values, update tracked entities, search tracked entities across org units and access tracker capture

Refer to the DHIS2 Documentation for more information on configuring user roles.

Organisation Units

You must assign the program to organisation units within your own hierarchy in order to be able to see the program in tracker capture.

Duplicated metadata

NOTE: This section only applies if you are importing into a DHIS2 database in which there is already meta-data present. If you are working with a new DHIS2 instance, you may skip this section.

Even when metadata has been successfully imported without any import conflicts, there can be duplicates in the metadata - data elements, tracked entity attributes or option sets that already exist. As was noted in the section above on resolving conflict, an important issue to keep in mind is that decisions on making changes to the metadata in DHIS2 also needs to take into account other documents and resources that are in different ways associated with both the existing metadata, and the metadata that has been imported through the configuration package. Resolving duplicates is thus not only a matter of "cleaning up the database", but also making sure that this is done without, for example, breaking potential integrating with other systems, the possibility to use training material, breaking SOPs etc. This will very much be context-dependent.

One important thing to keep in mind is that DHIS2 has tools that can hide some of the complexities of potential duplications in metadata. For example, where duplicate option sets exist, they can be hidden for groups of users through sharing.

Constants

TB Case Surveillance Tracker package includes a set of tests and a list of drugs that can be modified by the implementing country according to national context (e.g. which drugs and tests are used/available in country). The use of constants and corresponding program rules enables a system admin in an implementing country to easily ‘turn on’ or ‘turn off’ types of drugs and tests depending on availability in country. When the complete package is installed into a DHIS2 instance, all data elements for all tests and drugs included in this package are included in the system. The constant displaying lists of drugs in the Treatment Stage is set to **2** while all other constants are set to **1** (enabling the related data elements for data entry) and can be set to **2** by an implementer or system admin according to country context if not

needed (disabling the related data elements for data entry). If a test or drug later becomes available in the country, an admin can simply re-enable the data elements by changing the constant from a value of **2** to a value of **1**.

Constant (Laboratory test)	**UID**
Sputum Smear Microscopy	q1ah12sKfG3
TB-LAMP	yF0xR5nDSXa
LF-LAM	riDIK5mvDzW
Xpert MTB/RIF	H4ObQDbhnTA
Xpert MTB/RIF Ultra	cFoFDkXKcXC
Truenat	zBicQdPbHfj
Culture in Solid Media (e.g. LJ)	iSKEqcuVAui
Culture in Liquid Media (e.g. MGIT)	qpAseG5vJyS
Initial Phenotypic DST in Solid Media (e.g. LJ)	HjN2Bgnusyy
Initial Phenotypic DST in Liquid Media (e.g. MGIT)	OQxeAIyQUeB
Subsequent Phenotypic DST in Solid Media (e.g. LJ)	BpRfvWQcvTo
Subsequent Phenotypic DST in Liquid Media (e.g. MGIT)	W8Fm1pJuJPL
LPA (Rifampicin / Isoniazid)	ESUffSPwmju
LPA (Fluoroquinolones / Second-line Injectables)	govArZqiFzY

For Phenotypic DST, the applicable drugs can also be pre-configured according to the country standards during the initial setup by adjusting the values of constants. The package includes the following first- and second-line drugs in both Initial and Subsequent DST sections:

TB Drug	**Initial Phenotypic DST in Solid Media (e.g. LJ)**	**Initial Phenotypic DST in Liquid Media (e.g. MGIT)**	**Subsequent Phenotypic DST in Solid Media (e.g. LJ)**	**Subsequent Phenotypic DST in Liquid Media (e.g. MGIT)**
Rifampicin - **R**	Q67DDsupq7v	uOJYEwV7XfN	zJoEjvstp2d	lelJEADYpeI
Isoniazid critical concentration - **H CC**	dAPAScvFPfX	cRldkiBh5xv	eM3UfUO7W8O	ZBpqH7xJLBO
Isoniazid clinical breakpoint **H CB**	anpSc1tmlft	lqaf0KfpHGd	s8WPR943gGS	F9DTb5zl8rS

TB Drug	Initial Phenotypic DST in Solid Media (e.g. LJ)	Initial Phenotypic DST in Liquid Media (e.g. MGIT)	Subsequent Phenotypic DST in Solid Media (e.g. LJ)	Subsequent Phenotypic DST in Liquid Media (e.g. MGIT)
Pyrazinamide **Z**	aCaNdqUIDZl	FDtk4otxDse	bHPYGfrFNV2	VY15auehssY
Ethambutol - **E**	yxPHZFwMTN6	n9zOOsLO0QP	wNxsAieLWYc	lc5asMdbpH8
Levofloxacin - **Lfx**	NlOX3oV4gWe	t7Okdm8VgDV	Fw8wOlTETPt	b1KgVOel21P
Moxifloxacin critical concentration - **Mfx CC**	RJNE1o9cw7Y	dgjpdQO2Iva	Czx5FuOqF9i	HhKitGif8RV
Moxifloxacin clinical breakpoint - **Mfx CB**	m4c79OHEKCG	UL61U78GWCg	NvA1K4hFJbc	xfltmyqC3p0
Amikacin - **Am**	XO1V9o5C95J	x6v8O6EZo0U	dTwKm1u2VhY	a4PpEfSutcV
Bedaquiline - **Bdq**	UxcTzMRQsfa	KjvwSybuWJU	r70ONQRhaHY	QaioqMbO0TX
Delamanid - **Dlm**	VUlsMrm4D8k	BJTzi3WWjhT	RFqmLP5W5di	Xt5GU9kBD7q
Linezolid - **Lzd**	CVQR2ZtZWxk	aZq11UuXPP6	KIeI2d8xalG	mU5iEABeF2K
Clofazimine - **Cfz**	ZDpLleSK08x	GL5JTtBvbEC	bywBn5DxVdi	Nb6oHqjCCZq

For LPA (Fluoroquinolones / Second-line Injectables) the use of Ethambutol can be preconfigured using the following constant:

TB Drug	UID
Ethambutol	AiyTLOJHMkl

The use of list of first-line and second-line drugs in the treatment stage can be preconfigured using the following constant:

Constant	UID
Lists of First-line and Second-line drugs	GxOlFoUKbHB

Configuring tracker capture interface, widgets and top bar

You must configure tracker capture dashboard after the package has been installed. This configuration includes data entry forms, widgets and top bar.

Data entry forms

- After registering the first (test) case, access the **Settings** menu in the tracker capture form and select **Show/Hide Widgets**
- Switch from **Timeline Data Entry** to **Tabular Data Entry**
- Make sure that **Enrollment**, **Feedback** and **Profile** widgets are selected. Click **Close**.
- Adjust the widgets on the screen as shown below:

Enrollment	**Profile**
Feedback	
Tabular Data Entry	

Top Bar

- Access the **Settings** menu and select **Top bar settings**
- Select **Activate top bar**
- Select required information fields and assign their **Sort order**

Recommended fields	**Sort order**
Attributes	
TB Registration Number	2
Indicators	
Case classification	8
Patient's age (months)	4
Treatment regimen	9
Resistance classification	7
Months since diagnosis	3
Patient's age (years)	5
Date of diagnosis	1
Resistance	10
HIV Status	6

- Click **Save**
- Return to the **Settings** menu. Click **Saved dashboard layout as default**. Lock layout for all users.

Reporting case-based data into aggregate TB reports

The TB case-based surveillance tracker captures data that can be fed into standard, aggregate reporting (i.e. monthly, quarterly, or more frequently as determined by the country). An aggregate TB system design in DHIS2 can be accessed at who.dhis2.org/documentation/#tb Mapping of **program indicators** in TB Case Surveillance tracker with **data elements** and **category option combinations** in the aggregate package is required for the reporting process.

Adapting the tracker program

Once the programme has been imported, you might want to make certain modifications to the programme. Examples of local adaptations that *could* be made include:

- Adding additional variables to the form.
- Adapting data element/option names according to national conventions.
- Adding translations to variables and/or the data entry form.
- Modifying program indicators based on local case definitions.

However, it is strongly recommended to take great caution if you decide to change or remove any of the included form/metadata. There is a danger that modifications could break functionality, for example program rules and program indicators.