Common metadata library - Installation Guide { #gen-lib-installation }

This document includes an installation guide for the common metadata library.

System default language: English

Available translations: French, Spanish, Portuguese, Arabic

Installation

The table below will help you identify which metadata file/s are relevant for your implementation:

Metadata file	Application area
Common metadata library (complete)	DHIS2 environment containing both tracker and aggregate modules, implementation of published metadata packages, development of custom programs and modules that utilise common metadata
Common tracker metadata library	Tracker-based DHIS2 environment, implementation of published tracker packages, development of custom tracker programs that utilise common tracker metadata
Core case profile	Tracker-based DHIS2 environment, implementation of published tracker packages, development of custom tracker programs that utilise core case profile data (Person)
Common aggregate metadata	Implementation of published aggregate modules, development of custom aggregate modules that utilise common aggregate metadata

For more information, refer to the Common metadata library design guide

Carefully read through each section of the installation guide before importing the metadata files into your DHIS2 instance!

Importing common metadata into a blank DHIS2 instance

The application area of the DHIS2 instance will help you identify, which metadata file/s to import. The metadata can be imported without any additional adjustments.

Importing common metadata into a working DHIS2 instance

Overwriting metadata in the target instance may change the configuration of the existing programs and may lead to errors. When you import the Common metadata library into a working DHIS2 environment, you need to make sure that you do not lose any existing configurations (ie. sharing settings, option sort order, etc).

If the metadata objects in the common metadata library and in the target instance share the same UIDs but are not identical (eg. different aggregation type, different assigned option set, different number of options), consider assigning new UIDs to the metadata objects in the target instance, before importing the common metadata library.

UIDs

Replacing UIDs for metadata elements in the target system to match the source metadata file can be a complex task, as the uid maybe referenced in other metadata objects, such as program rules, indicators, predictors, validation rules, etc. Make sure to replace all occurances of the old uid with the new uid in the system before importing the metadata object.

> **EXAMPLE**
>
> To update the uid of a category option use the following command:
>
> ```
> UPDATE dataelementcategoryoption SET uid = '<new-uid>' WHERE uid = '<old-uid>';
> ```

Tracked entity attributes, data elements, option sets and options

For option set based data elements and attributes, make sure that these option sets are identical in the source file and in the target instance. Check that the matching option sets contain the same options.

Check the aggregation and value types of the data elements and attributes in the source file and the target instance.

Option codes

According to the DHIS2 naming conventions, the metadata codes use capital letters, underscores and no spaces. Some exceptions that may occur are specified in the corresponding package documentation. If data values in the target instance contain lower case codes, it is important to update those values directly in the database.

The table below contains all option sets where codes were changed to upper case in the metadata package. Before importing metadata into the instance, check whether the option sets in the existing system match those in the package .json and use the same upper case option codes.

> **Important**
>
> During the import, the existing option codes will be overwritten with the updated upper case codes. In order to update the existing data values, it is necessary to update the values stored in the database using database commands. Make sure to map existing old option codes and new option codes before replacing the values. Use staging instance first, before making adjustments on the production server.

For data element values, use:

```
UPDATE programstageinstance
SET eventdatavalues = jsonb_set(eventdatavalues, '{"<affected-data-element-uid>","value"}', '"<
WHERE eventdatavalues @> '{"<affected-data-element-uid>":{"value": "<old-value>"}}'::jsonb
AND programstageid=<database_programsatgeid>;
```

For tracked entity attribute values, use:

```
UPDATE trackedentityattributevalue
SET value = <new-value>
WHERE trackedentityattributeid=<affected trackedentityattribute database_id> AND value=<old-val
```

Tracked entity type

If the tracked entity types in the Common metadata library and the target instance match, reviw the sharing settings, the assigned tracked entity type attributes and their settings, before importing the Common metadata library.