

# CHIS Installation Guide { #chis-agg-installation }

---

## Overview

---

This document is intended to guide administrators through the process of installing configuration packages for DHIS2 for aggregate reporting. Good understanding of DHIS2 is required.

The configuration packages for DHIS2 consists of a metadata file in [JSON](#) format which can be imported into a DHIS2 instance, as well as accompanying documentation. The package provides pre-defined, standard content according to WHO recommendations. This document is concerned with complete packages that include configurations of data collection (data sets, data elements, validation rules) as well as analysis and dashboards (chart, map and pivot table favourites). This is intended for use where there is no data collection configured in DHIS2, or if replacing/revising existing content to align with WHO recommendations.

Configuration packages consist of the following components:

- data sets with data elements;
- package-specific metadata objects (predictors, constants, organisation unit groups, etc);
- a set of indicators;
- analytical outputs (pivot tables, data visualizations and GIS favourites);
- a set of dashboards.

These components are all linked, i.e. the indicators are based on the included data elements, the analytical outputs are based on the included indicators, and the dashboards are made up of the included analytical outputs.

## Installation

---

Installation of the module consists of two steps:

1. [Preparing](#) a metadata file with DHIS2 metadata.
2. [Importing](#) a metadata file into DHIS2.
3. [Configuring](#) the imported metadata.
4. [Performing](#) package-specific modifications.

This section deals with the first two steps of preparing and importing the metadata file into DHIS2, whilst the configuration procedure is discussed in the next section. It is recommended to first read through both sections before starting the installation and configuration process in DHIS2. In addition to the general steps described here, some of the configuration packages have annexes to the installation guide, describing particular issues. These are listed in the appropriate section [here](#).

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.

Multiple configuration packages

Some configuration packages have overlapping metadata, for example indicators. This means that in some situations, changes to metadata to configuration packages that have been imported previously may be overwritten when importing a different package. This can be avoided by importing "new only" metadata rather than "new and updates", but note that with either approach manual modifications will be needed. At a minimum, you must ensure that metadata used by multiple programmes are [shared](#) with the appropriate user groups for both programmes.

## Requirements

---

In order to install the configuration package, an administrator user account in DHIS2 is required, with authorities to add/edit (public) metadata objects, including (but not limited to):

- data elements (including categories)
- data sets
- indicators
- indicator types
- dashboards
- data visualizer, pivot table and GIS favourites

The full list of objects included in the module (for which the administrator needs authorities to manage) can be found in the Metadata reference document for the particular configuration package.

In cases where modifications to the .json metadata file of the configuration package are necessary ([see below](#)), a [text editor](#) is needed - these modifications should not be done with a word processor such as Microsoft Word.

The configuration package can be installed in DHIS2 through the DHIS2 Health App, or manually through a .json file with DHIS2 metadata using the [Import/Export](#) app of DHIS2. The procedure described in the rest of this section applies to the process of manually importing metadata.

The [Configuration](#) section applies for both methods.

## 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 on "[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 UID of the default data dimension 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 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 UUIDs of the .json file with the UUIDs of the database in which the file will be imported. Table 1 shows the UUIDs which should be replaced, as well as the API endpoints to identify the existing UUIDs.

Object	UUID	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

For example, if importing a configuration package into <https://play.dhis2.org/demo>, the UUID of the default category option combination could be identified through <https://play.dhis2.org/demo/api/categoryOptionCombos.json?filter=name:eq:default> as bRowv6yZ0F2 . You could then search and replace all occurrences of HllvX50cXC0 with bRowv6yZ0F2 in the .json file. 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 simply by their factor and whether or not they are simple numbers without denominator, they are unambiguous and can be replaced through a search and replace of the UUIDs. This avoids potential import conflicts, and avoids creating duplicate indicator types. Table 2 shows the UUIDs which could be replaced, as well as the API endpoints to identify the existing UUIDs.

Object	UUID	API endpoint
Numerator only (number)	kHy61PbChXr	../api/indicatorTypes.json?filter=number:eq:true&filter=factor:eq:1
Percentage	hmSnCXmLYwt	../api/indicatorTypes.json?filter=number:eq:false&filter=factor:eq:100

**Note 1:** by replacing the UUIDs as described and importing the .json file, the name (including any translations) of the indicator types in question will be updated to those included in the configuration packages.

**Note 2:** An alternative approach to re-using the existing indicator types is to import those included in the configuration package, and removing the existing ones that overlap. This would require all indicators that use these existing indicator types to be updated to the newly imported indicator types, before the indicator types can be deleted.

## Importing a metadata file into DHIS2

The .json metadata file is imported 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 the metadata. If "dry run" reports any issues or conflicts, see the [Handling 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 [configure](#) the module. In some cases, import conflicts or issues are not shown during the "dry run", but appears 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 into a new DHIS2 instance, you will not have to worry about import conflicts, as there is nothing in the database you are importing to conflict with. Follow the instructions to import the metadata then please proceed to the "[Additional 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 exists 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 for which there is a conflict. The advantage of this approach is that there is no need to modify the .json file, as changes is instead done through 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 training material and documentation based on the configuration package will be used.

### 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. At the same time, modifying object of the .json file means that using the associated documentation and training material might become more complicated.

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 indicator already exists for a certain concept (e.g. "ANC 1 coverage"), that indicator could be removed from the .json file and all references to its UID replaced with the corresponding indicator 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. However, it is generally not recommended for several reasons:

- it requires expert knowledge of the detail 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.
- as with alternative 2, it means that the result of the installation is not entirely according to the standard configuration, and training material and documentation developed for the configuration might not be usable without modifications.
- future updates to the configuration package will be complicated.

## Additional Configuration

---

Once all metadata has been successfully imported, the module should be useable with only a few minor additional adjustments. In addition, depending entirely on the DHIS2 instance the module has been imported into, some additional configuration and modification might be necessary or advantageous. This section will first go through the necessary configuration steps, then mention some of the other types of modifications or configuration that might be relevant.

## Required configuration

---

Before the configuration packages can be used for data collection, there are two steps that are required.

- Assign the data set(s) to appropriate organisation units
- Share the data set(s) with appropriate user groups
- Add your user(s) to the appropriate user groups

The data sets should be assigned to the organisation units (facilities) which are expected to report on that particular data set.

The data sets and category options should also be shared with the relevant user group(s) of the personnel who are responsible for doing data entry for those data sets.

## 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. Sharing should be configured for the following:

- Data elements/Data element groups
- Indicators/Indicator groups
- Data Sets
- Dashboards

The user groups that usually come with the package include:

- {Package Name} access
- {Package Name} admin
- {Package Name} data capture

We recommend to provide the following access to these groups

Object	User Group		
	<i>{Package Name}</i> access	<i>{Package Name}</i> admin	<i>{Package Name}</i> data capture
<i>Data Elements/Data element group</i>	Metadata : can view Data: can view	Metadata : can edit and view Data: can view	Metadata : can view Data: can view
<i>Indicators/Indicator group</i>	Metadata : can view Data: can view	Metadata : can edit and view Data: can view	Metadata : can view Data: can view
<i>Data sets</i>	Metadata : can view Data: can view	Metadata : can edit and view Data: can view	Metadata : can view Data: can capture and can view
<i>Dashboards</i>	Metadata : can view Data: can view	Metadata : can edit and view Data: can view	Metadata : can view Data: can view

## Duplicated metadata

Even when metadata has been successfully imported without any import conflicts, there can be duplicates in the metadata - whether it's a chart, indicator, data element or data element category that already exists. As was noted in the section above on resolving conflict, an important issues 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 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 data element categories exists these duplications can be hidden to end users through category option group sets, or certain metadata objects can be hidden for groups of users through sharing.

It is recommended not to remove or replace the indicators included in configuration packages even though the same indicator already exists, so that the analytical outputs (which as based exclusively on indicators) can be kept. This will allow training material based on the configuration packages to be re-used.

## Examples of other modifications

In addition to the required configuration, there are a number of other modifications and configuration that might be relevant, depending on the specific situation. It will not be possible to provide guidance and recommendations that cover all the different scenarios, but a brief discussion of some common problems are presented here.

## Translations

Additional translations might have to be added, if other languages than those included are needed.

## Adding additional variables

The configuration packages includes key recommended data elements and indicators. However, it might in some cases be necessary to add additional variables to address country-specific information needs. These could be added to the included data sets.

## Updating layout of reporting forms

To facilitate the work of personnel doing data entry in DHIS2, it is sometimes desirable to add a custom data entry form that match the format of paper forms used at the primary level.

## Maintenance

---

No particular maintenance is required for the configuration packages, since they are made up of standard DHIS2 metadata objects. However, before upgrading to new versions of DHIS2, it is important to test all functionality of the system in general on a staging/test server before upgrading any production instances of the system.