

Rehabilitation - Aggregate Installation Guide { #rehab-installation }

This document includes an installation guide for the Rehabilitation aggregate package.

System default language: English

Available translations: French, Spanish, Portuguese, Arabic

Overview

=== "DHIS2.35"

```
```json
"package": {
 "DHIS2Build": "834b25f",
 "DHIS2Version": "2.35.8",
 "code": "RHAG00",
 "description": "",
 "lastUpdated": "20220221T181704",
 "locale": "en",
 "name": "RHAG00_DHIS2.35",
 "type": "AGG",
 "version": "1.0.0"
}
```
```

=== "DHIS2.36"

```
```json
"package": {
 "DHIS2Build": "aa38c7f",
 "DHIS2Version": "2.36.8",
 "code": "RHAG00",
 "description": "",
 "lastUpdated": "20220222T131957",
 "locale": "en",
 "name": "RHAG00_DHIS2.36",
 "type": "AGG",
 "version": "1.0.0"
}
```
```

=== "DHIS2.37"

```
```json
"package": {
 "DHIS2Build": "a7328e1",
 "DHIS2Version": "2.37.3",
 "code": "RHAG00",
 "description": "",
 "lastUpdated": "20220222T132004",

```

```

 "locale": "en",
 "name": "RHAG00_DHIS2.37",
 "type": "AGG",
 "version": "1.0.0"
 }
 ...

```

## Installation

Installation of the module consists of several steps:

1. [Preparing the metadata file with DHIS2 metadata.](#)
2. [Importing the metadata file into DHIS2.](#)
3. [Configuring the imported metadata.](#)
4. [Adapting the program after import](#)

It is recommended to first read through each section of the installation guide before starting the installation and configuration process in DHIS2. Identify applicable sections depending on the type of your import:

1. import into a blank DHIS2 instance
2. import into a DHIS2 instance with existing metadata.

The steps outlined in this document should be tested in a test/staging DHIS2 instance and only then applied to a production environment.

## Requirements

In order to install the module, an administrator user account on DHIS2 is required.

Great care should be taken to ensure that the server itself and the DHIS2 application are well secured, access rights to collected data should be defined. Details on securing a DHIS2 system is outside the scope of this document, and we refer to the [DHIS2 documentation](#).

## Metadata files

While not always necessary, it can often be advantageous to make certain modifications to the metadata file before importing it into DHIS2.

## Preparing the Metadata File

### Default data dimension

In early versions of DHIS2, the UUIDs 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 UUIDs of these defaults can be different. Later versions of DHIS2 have hardcoded UUIDs for the default dimension, and these UUIDs 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 from the instance 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

Object	UID	API endpoint
Category option combination	H1lvX50cXC0	../api/categoryOptionCombos.json? filter=name:eq:default

Identify the UIDs of the default dimensions 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)	kHy61PbChXr	../api/indicatorTypes.json? filter=number:eq:true&filter=factor:eq:1

## Visualizations using Root Organisation Unit UID

Visualizations, event reports, report tables and maps that are assigned to a specific organisation unit level or organisation unit group, have a reference to the root (level 1) organisation unit. Such objects, if present in the metadata file, contain a placeholder `<OU_ROOT_UID>`. Use the search function in the .json file editor to possibly identify this placeholder and replace it with the UID of the level 1 organisation unit in 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. All codes included in the metadata objects in the current package match the naming conventions. It may occur that the codes of existing metadata objects used in the target database use lower case characters. In this case, it is important to update those values directly in the database.

#### Important

During the import, the existing option codes will be overwritten with the updated upper case codes. In order to update the data values for existing data in the database, 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:

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

## 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.

The Rehabilitation package contains one option set and two options:

```
```json
{
  "optionSets": [
    {
      "name": "YES/NO (numeric)",
      "id": "TdDqpX1kdd2",
      "code": "YES_NO_NUM",
      "valueType": "INTEGER_ZERO_OR_POSITIVE",
      "options": [
        {
          "id": "VavIEUmBv8j"
        },
        {
          "id": "Xu8ieCbS7jH"
        }
      ]
    }
  ],
  "options": [
    {
      "name": "Yes",
      "id": "VavIEUmBv8j",
      "code": "1",
      "sortOrder": 1,
      "optionSet": {
        "id": "TdDqpX1kdd2"
      }
    },
    {
      "name": "No",
      "id": "Xu8ieCbS7jH",
      "code": "0",
      "sortOrder": 2,
      "optionSet": {
        "id": "TdDqpX1kdd2"
      }
    }
  ]
}
```
```

This Yes/No option set is based on "INTEGER\_ZERO\_OR\_POSITIVE" option values that are evaluated in predictors in order to determine **Rehabilitation essential package availability at PHC level** and count number of facilities offering essential packages.

Organisation unit groups and group sets { #rehab-orgunitgroups }

The package includes the following Organisation Unit Groups:

| Name                         | UID         | Description                                                               | Purpose                        |
|------------------------------|-------------|---------------------------------------------------------------------------|--------------------------------|
| PHC                          | aT5pkgRLbw5 | Primary health care facilities                                            | data set assignment, analytics |
| REHAB - Master Facility List | Uvefj6bDfzo | Includes all facilities reporting on rehabilitation                       | data set assignment, analytics |
| REHAB - PHC                  | bbsxlCu3Vya | Includes all primary health care facilities reporting on rehabilitation   | Analytics                      |
| REHAB - SHC                  | wZJCB2cj9jg | Includes all secondary health care facilities reporting on rehabilitation | Analytics                      |
| REHAB - THC                  | Re0iJ3vtBzE | Includes all tertiary health care facilities reporting on rehabilitation  | Analytics                      |
| Rehab inpatient ward         | AGK6o0K4ncb | Includes all facilities with a dedicated rehabilitation ward              | Analytics                      |
| Hospital district            | Y9lBaYVm9j7 | Includes all district hospitals                                           | Analytics                      |

and the following Organisation Unit Group Sets:

| Name                          | UID         | Description                                                               | Purpose   |
|-------------------------------|-------------|---------------------------------------------------------------------------|-----------|
| Administrative levels of care | dSwpdHITQ85 | Administrative levels of care eg. PHC, SHC, THC                           | Analytics |
| REHAB - Administrative levels | wkjpdklq0It | Rehabilitation levels of care eg. REHAB PHC, REHAB SHC, REHAB THC         | Analytics |
| Type                          | VQT2m5uMawR | Includes types of facilities eg. District Hospitals, Health centres, etc. | Analytics |

These metadata objects have to be configured.

1. If the target instance does not contain any organisation unit groups that match the description of the groups included in the package, follow the steps below during configuration and import:
  1. Import the package together with the included organisation unit groups.
  2. Assign applicable facilities to the new organisation unit groups in the Maintenance app.
2. If the target instance already contains organisation unit groups that match the description for the given metadata objects, follow the steps below during configuration and import:
  1. Note the UIDs of the matching organisation unit groups in the target instance.
  2. Replace all occurrences of the organisationUnitGrop UIDs in the metadata json file with the corresponding UIDs noted in step 1.
  3. Remove the organisationUnitGroup metadata objects from the metadata json file before import. This step is very important, otherwise the current assignment of the organisation units to existing groups in the target instance will be overwritten.
  4. Proceed to importing the package if no other pre-configuration / editing is required.

3. If the target instance does not contain organisation unit group sets that match the description provided, these organisation unit groups have to be imported into the target instance. The applicable organisation unit groups have to be added to the organisation unit group sets in the user interface or using the API.
4. If the target instance already contains organisation unit group sets that match the description provided, follow the steps below during configuration and import:
  1. Replace the UUIDs of the matching organisation unit group sets in the metadata file with the corresponding UUIDs of the organisation unit group sets from the target instance.
  2. Remove the organisation unit group set objects from the metadata file before import. This step is very important, otherwise the current assignment of the organisation unit groups to existing group sets in the target instance will be overwritten.
  3. Add the newly imported organisation unit groups to the organisation unit group sets. (See tables above).

### Example

The target instance may already contain the PHC organisation unit group. In this case, replace the UUID **aT5pkgRLbw5** of the group and all its occurrences in the json file with the corresponding UUID from the target instance before import. Then, remove the orgUnitGroup "PHC" metadata object from the json file. You will find it under **"organisationUnitGroups":**

## Population, incidence and personnel density data

The Rehabilitation package includes data elements, indicators and other metadata objects that use on **population, incidence** and **personnel density** data.

The organisation unit levels at which the population data is entered in the target instance may vary.

In the generic Rehabilitation package, this metadata is added to the to the facility level data sets listed in the table below.

| Data element                   | UID         | Data Set name                          | Data Set UID | Data set period type | Data Set organisation Unit Group |
|--------------------------------|-------------|----------------------------------------|--------------|----------------------|----------------------------------|
| GEN - Population               | DkmMEcubiPv | REHAB - personnel density              | Sm2fALTZR0S  | Yearly               | REHAB - Master Facility List     |
| REHAB - Amputation incidence % | jEc1P0VAPcs | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |
| REHAB - Burns incidence %      | rtYJ0Nzb70Y | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |
| REHAB - MMT incidence %        | j1S0RS2LplZ | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |
| REHAB - SCI incidence %        | Iy6ylb65g4V | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |
| REHAB - Stroke incidence %     | 0IADGq0kCHW | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |
| REHAB - TBI incidence %        | rzVRv6GpduW | REHAB - bed density and incidence data | giKizLegiUW  | Yearly               | Rehab inpatient ward             |

If the target instance already has metadata infrastructure, which is used for collecting **Population, personnel or incidence data**, please refer to the steps listed below:

1. Choose the strategy to align population metadata in the target instance and in the .json file.

- Alternative 1: Replace the UUIDs of the data elements and all their occurrences in the json file with the UUIDs from the target system
- Alternative 2: Consider replacing the UUIDs of these data elements in the target system with the UUIDs from the json file. GEN data elements are part of DHIS2 core metadata library and are used in other metadata packages.

2. Indicators that use the **Population, personnel or incidence data** will be aggregating data at the level/levels where the data is entered.

3. Additional mapping and configuration may be required after the package is imported. Refer to the [data set configuration section](#)

#### NOTE

When updating the UUID of a metadata element in the existing DHIS2 instance, you will need to run an SQL command in the database and additionally replace all occurrences and references of its UUID in other metadata objects: predictor, indicator, validation rule expressions, etc.

Predictors { #rehab-predictors }

The package includes the following predictors:

| Name                                           | UID         | Period type | Missing value strategy         | Output data element - name                     | Output data element - UID | Organisation unit levels |
|------------------------------------------------|-------------|-------------|--------------------------------|------------------------------------------------|---------------------------|--------------------------|
| REHAB - Availability (Essential package)       | zYbrCP7xGtk | yearly      | Skip if all values are missing | REHAB - Package availability (overall)         | bUJKsy9u2xv               | Facility level           |
| REHAB - Availability (occupational therapists) | MaWSMzXDLkm | yearly      | Skip if all values are missing | REHAB - Availability (occupational therapists) | RPKfAe7vo03               | Facility level           |
| REHAB - Availability (other personnel)         | aphqcwJiK5s | yearly      | Skip if all values are missing | REHAB - Availability (other personnel)         | qVvuda0dniT               | Facility level           |
| REHAB - Availability (physiotherapists)        | yd1TJLDcFBj | yearly      | Skip if all values are missing | REHAB - Availability (physiotherapists)        | N6ru55bPe1o               | Facility level           |
| REHAB - Availability (prosthetists/orthotists) | K3QZ2zs0opc | yearly      | Skip if all values are missing | REHAB - Availability (prosthetists/orthotists) | k1LHQzY0lw2               | Facility level           |
| REHAB - Availability (psychologists)           | RpxclhlYJxw | yearly      | Skip if all values are missing | REHAB - Availability (psychologists)           | pNNIXV0k0us               | Facility level           |
| REHAB - Availability (rehabilitation doctors)  | CbnJeF5K1cM | yearly      | Skip if all values are missing | REHAB - Availability (rehabilitation doctors)  | LQ10SQqGKyf               | Facility level           |

| Name                                              | UID         | Period type | Missing value strategy         | Output data element - name                        | Output data element - UID | Organisation unit levels |
|---------------------------------------------------|-------------|-------------|--------------------------------|---------------------------------------------------|---------------------------|--------------------------|
| REHAB - Availability (speech language therapists) | qHJTzQcMSd4 | yearly      | Skip if all values are missing | REHAB - Availability (speech language therapists) | SuZDPY0gFbN               | Facility level           |

Predictor metadata includes organisation unit levels used for aggregation of data values. The package metadata file contains placeholders that need to be replaced with the UIDs of the corresponding organisation unit levels in the target database.

The steps to prepare the predictors for import are described below:

1. Identify the organisationUnitLevel UID of the Facility level at which the data for the predictors will be aggregated. Use the following API endpoint to identify the required UID: `./api/organisationUnitLevels.json?fields=id,name`
2. Find the following organisationUnitLevel placeholders in the json file: `<OU_LEVEL_FACILITY_UID>`
3. Replace the placeholders with the UID of the identified facility level in the target instance.
4. Identify the number of the Facility level in the target instance.
5. Find the properties of the **output data elements** in the json file, using the UIDs provided in the "Output data element - UID" column.
6. Look for property: `"aggregationLevels": [4]`
7. If the level matches the level in the target instance, keep the number as is. If the Facility level number in the target instance is different, then adjust the number accordingly.

Validation rules { #rehab-validation-rules }

Validation rules included in the package are grouped by data sets for which they have been configured.

All validation rule groups and the corresponding validation rules are listed in the appendix to this installation guide:

#### Rehabilitation - validation rules

The organisation unit groups for all validation rules are set to facility level. The facility level value is located in the `"organisationUnitLevels"` property of each validation rule. It is set to `4` by default. Adjust these levels in the metadata file to match the facility level in the target instance before importing the package.

## 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.

## 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 program as well as who can register/enter data into the program. By default, sharing has been configured for the following:

- Dashboards
- Visualizations, maps, event reports and report tables
- Data sets
- Category options

Please refer to the [DHIS2 documentation](#) for more information on sharing.

Three core user groups are included in the package:

- Rehab access (view metadata/view data)
- Rehab admin (view and edit metadata/no access to data)
- Rehab - (view metadata/capture and view data)

The users are assigned to the appropriate user group based on their role within the system. Sharing for other objects in the package may be adjusted depending on the set up. Refer to the [DHIS2 Documentation on sharing](#) for more information.

### 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 unit assignment

The data sets must be assigned to organisation units within existing hierarchy in order to be accessible via capture app. The table below provides information about organisation unit assignment for Rehabilitation data sets:

| Data set                              | UID                         | Data set form type | Data collection period | Facility types                                                                                             |
|---------------------------------------|-----------------------------|--------------------|------------------------|------------------------------------------------------------------------------------------------------------|
| Bed Density and Incidence Data        | <a href="#">giKizLegiUW</a> | default            | yearly                 | Rehabilitation facilities with a dedicated rehabilitation inpatient ward                                   |
| Essential Package Availability at PHC | <a href="#">MGzqZDWvPhL</a> | section            | yearly                 | Primary Healthcare Facilities reporting on Rehabilitation (Rehab PHC)                                      |
| Personnel Density                     | <a href="#">Sm2fALTZR0S</a> | section            | yearly                 | All facilities reporting on Rehab (Master Facility List)                                                   |
| Inpatient Report                      | <a href="#">WjN1YoDt10d</a> | custom             | monthly                | All facilities with an inpatient ward (not dedicated Rehab ward) reporting on Rehab (Master Facility List) |
| Rehab Ward Report                     | <a href="#">tP8et8TNWgF</a> | custom             | monthly                | All facilities with a dedicated rehabilitation inpatient ward reporting on Rehab (Master Facility List)    |
| Outpatient Report                     | <a href="#">zInFVXb98JD</a> | custom             | monthly                | All facilities with an outpatient department reporting on Rehab (Master Facility List)                     |

## Organisation unit group assignment

The organisation units in the target system have to be assigned to the Rehab organisation unit groups based on the overview in the [Organisation unit groups](#) section.

Data sets { #rehab-dataset-config }

The following data sets require additional configuration after import:

### Bed density

If the annual data for total number of rehabilitation beds is already collected in the existing HMIS, the Rehab Bed density data set is not needed. Make sure to replace all occurrences of the data element **"Available rehabilitation beds (total)"** [K0Y941ADtGw](#) with the existing data element UID in all metadata objects, where this data element is referenced:

| Metadata object UID          | Metadata object type | Details                                     |
|------------------------------|----------------------|---------------------------------------------|
| <a href="#">V0dQ2YRmSz f</a> | Indicator            | Data element is referenced in the numerator |

### Personnel density

If the annual population data is already collected in the existing HMIS, the **"GEN - Population"** data element [DkmMEcubiPv](#) may be removed from the data set [Sm2fALTZR0S](#). Make sure to replace all occurrences of the **"GEN - Population"** data element with the existing data element UID in all metadata objects, where this data element is referenced:

| Metadata object UID | Metadata object type | Details                                       |
|---------------------|----------------------|-----------------------------------------------|
| 0nxT9nXB9yB         | Indicator            | Data element is referenced in the denominator |
| hLkZBsoxgwG         | Indicator            | Data element is referenced in the denominator |
| V0dQ2YRmSz f        | Indicator            | Data element is referenced in the denominator |
| n0cE7LiP4j8         | Indicator            | Data element is referenced in the denominator |
| peWxNUcIjZw         | Indicator            | Data element is referenced in the denominator |
| dXNfY2I7umm         | Indicator            | Data element is referenced in the denominator |
| hpP5GW43n1J         | Indicator            | Data element is referenced in the denominator |
| s9SRcnMtI0K         | Indicator            | Data element is referenced in the denominator |
| RRCtatVRlI0         | Indicator            | Data element is referenced in the denominator |
| PuSDjaFs2we         | Indicator            | Data element is referenced in the denominator |
| tsIeJwq6x8L         | Indicator            | Data element is referenced in the denominator |
| U5tL3Eqq3Vj         | Indicator            | Data element is referenced in the denominator |
| NcA1znaVgFH         | Indicator            | Data element is referenced in the denominator |
| M0UPequfEYf         | Indicator            | Data element is referenced in the denominator |
| U5WwSS3zx1X         | Indicator            | Data element is referenced in the denominator |
| fhZ9MI3qTaA         | Indicator            | Data element is referenced in the denominator |
| YEjkkya4JCJ         | Indicator            | Data element is referenced in the denominator |
| xW6TcvEMhwG         | Indicator            | Data element is referenced in the denominator |
| TNjjTJr7fLe         | Indicator            | Data element is referenced in the denominator |
| uKK11dDx8HH         | Indicator            | Data element is referenced in the denominator |
| qTq20E3B08y         | Indicator            | Data element is referenced in the denominator |
| ePjfu6Fr4Jq         | Indicator            | Data element is referenced in the denominator |
| zNzm3AUiQ3B         | Indicator            | Data element is referenced in the denominator |
| Vq98oh9BIB1         | Indicator            | Data element is referenced in the denominator |
| k1NqjksyNAL         | Indicator            | Data element is referenced in the denominator |
| bW75ZyPq9aZ         | Indicator            | Data element is referenced in the denominator |
| ME2YCnift7x         | Indicator            | Data element is referenced in the denominator |
| HTZ7STQR648         | Indicator            | Data element is referenced in the denominator |
| Z2f5wDvVxUL         | Indicator            | Data element is referenced in the denominator |
| X1ISfeHbzxc         | Indicator            | Data element is referenced in the denominator |
| NVbwb4XlTVo         | Indicator            | Data element is referenced in the denominator |
| t260bhmhj0b         | Indicator            | Data element is referenced in the denominator |
| FjVJNnV0u6S         | Indicator            | Data element is referenced in the denominator |
| fDj7xDywd5C         | Indicator            | Data element is referenced in the denominator |
| BLUTcTXPhTs         | Indicator            | Data element is referenced in the denominator |

| Metadata object UID | Metadata object type | Details                                       |
|---------------------|----------------------|-----------------------------------------------|
| RFVOIDIULVO         | Indicator            | Data element is referenced in the denominator |

The level of data collection for **incidence** data has to be the same or lower than the level of data collection for **population** data.

The organisation unit assignment of the **Personnel density** data set should remain at facility level for the purpose of analytical outputs.

## 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, please skip this section and go to [Adapting the tracker program](#). If you are using any third party applications that rely on the current metadata, please take into account that this update could break them"

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](#).

## Adapting the program

Once the program has been imported, you might want to make certain modifications to the program. 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 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.

## Indicator mapping

For partial implementation of the Rehabilitation package, i.e. implementation of a customized WHO core indicator set, please refer to the [WHO-to-DHIS2 indicator mapping table](#).

When adapting metadata, make sure to identify visualizations and dashboards where applicable indicators are used, as well as data elements and category combinations used in the corresponding data sets.

## Removing metadata

In order to keep your instance clean and avoid errors, it is recommended that you remove the unnecessary metadata from your instance. Removing unnecessary metadata requires advanced knowledge of DHIS2 and various dependencies.