

WHO Cause of Death - Tracker Installation Guide { #crvs-cod-installation }

Introduction

This document is intended to guide administrators through the process of installing the Cause of Death module in a DHIS 2 database. Good understanding of DHIS 2, including the Tracker (case-based) functionality is required.

The Cause of Death module consists of a file with DHIS 2 metadata. More specifically, it includes:

For Tracker program:

- A tracker program with data elements, option sets, custom data entry form, program rules, program indicators etc.
- A standard report for validation of data.
- A dashboard with charts and tables for key data.
- This document describes how to install this module, and perform routine maintenance. This includes the critical task of adapting and improving the dictionary of diagnosis/medical terms, which is essential for the functioning of the module.

In addition, a [node.js](#) script for linking the Cause of Death module in DHIS 2 with the [IRIS software](#) for semi-automatic coding of causes of death to full ICD-10 and sending the result back to the DHIS 2 database. Outputs are provided to export this again data to the ANACOD data analysis tool. This functionality is described in a separate document.

Description of the module

This section gives an overview of the design and structure of the module.

The data collection form

The data collection form is created based on the International Medical Certificate of Causes of Death. As such, it uses a custom form that mimics the layout of this certificate. However, the logic built into the form for coding and validation also works without the custom form, allowing the module to be used on e.g. mobile devices with the DHIS 2 android applications.

Dealing with "Age"

For collecting data on the age of the deceased, the data collection form supports use of both date of birth and an estimated age. To facilitate analysis of the data, an additional data element for Age in years has been added to the form, but as a disabled/read-only field. Based on what is entered either as date of birth or estimated age, program rules calculate the age in years and assigns that value to the Age in years data element.

Dictionary

A critical component of the module is a dictionary (ICD-SMoL - local dictionary). The dictionary functions as a link between the medical terms used in data entry, and the ICD-SMoL code used for reporting and analysis of causes of death. In DHIS 2, the dictionary is implemented as an option set. The medical term is the option name, and the option code is made up of three components: the ICD-SMoL code, the ICD-10 code, and an ID. The ID is necessary because several terms could refer to the same ICD-SMoL and ICD-10 codes, and codes need to be unique. The ID itself consists of two parts: a country code (or INT for the ones provided with the module) and a counter. Keeping track of the international and country-specific terms will make it easier in the future to e.g. identify terms used in different countries which could be proposed for the international dictionary.

Fully localised French "Cause of Death" module uses a local dictionary that is different from the English ICD-SMoL - local dictionary. This dictionary is maintained in a different option set that is only included in the French package file.

The dictionary is sorted by length of the term, with the shortest terms appearing first. This ensures that as users search the dictionary, the terms where the largest proportion of the medical term matches with the search term is displayed first.

Selecting underlying cause of death

Selecting the underlying cause of death is a key component of the module. This is done through four checkboxes, which indicate which of the (up to) four causes is the underlying cause. Program rules are used to ensure that only one of the checkboxes can be selected, and that each checkbox can only be used if a diagnosis has been selected for that line/cause.

Once a cause has been selected as underlying, two read-only fields are assigned with the ICD-SMoL name and code for the selected cause.

Installation

Requirements and preparation

In order to install the module, an administrator user account on DHIS 2 is required. The procedure outlined in this document should be tested in a test/staging environment before being performed on a production instance of DHIS 2.

Great care should be taken to ensure that the server itself and the DHIS application is well secured, to restrict access to the data being collected. Even if using this module without any direct patient identifiers like ID or names, the age, sex and health facility could be sufficient to identify an individual. Details on securing a DHIS 2 system is outside the scope of this document, and we refer to the DHIS 2 documentation.

Choosing a version of the module

Importing metadata

The module is imported through a `.json` file with DHIS 2 metadata, using the Metadata import feature of the DHIS 2 Import/Export app. When importing the metadata in a DHIS 2 database with existing metadata, you might see errors if you look at the detailed import summary. These could require action, for example if a data

element with the same name or code already exists in the database your are importing the module into. In this case, you will either have to change the identifier (name, code) in the database, or modify the name/code in the import file. Any such modifications should be done first in a test or staging environment, to make sure the changes do not affect other parts of the system.

Additional configuration

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

1. The programme must be assigned to a new or existing user role before users can start entering data. In addition, users need to have the necessary authorities in order to enter and/or see tracker data. Refer to the DHIS 2 Documentation for more information on configuring user roles.
2. The programme must be assigned to the organisation units for which data will be entered.
3. Optionally, you might also want to use the Sharing functionality of DHIS 2 to configure which users (user groups) should see the metadata and data associated with the programme. Sharing can be configured for the program itself, data elements, program indicators, aggregate indicators, option sets, and the analytical outputs.

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.
- For the tracker programme, linking the programme to an existing tracked entity type and/or modifying the tracked entity attributes. A placeholder tracked entity is included in the module.
- Adapting data element/option names according to national conventions.
- Adding translations to variables and/or the data entry form.
- However, it is strongly recommended not to change or remove any of the included form/metadata. This is both because the included variables are the standard recommended variables from the international death notification form, and because there is a danger that modifications could break functionality, for example program rules and program indicators. For example almost all variables in the form are linked to one or more program rules, which would cease to function if the variable is removed or replaced. Changing the default variables could also complicate updates to the module in the future.

Maintenance

Maintenance of the module has two main aspects: maintenance of the dictionary of medical terms, and maintenance related to new versions of DHIS 2. Each of these aspectes will be described here.

Dictionary maintenance

The dictionary of medical terms is a key component of the module, and is critical in order to produce meaningful data. This section describes how to check the integrity of the dictionary (e.g. that the codes used are valid), list terms that have been collected which are not in the dictionary and should thus be considered for

inclusion, and how to add new terms to the dictionary. Several SQL-views are included in the module to facilitate these tasks.

Checking integrity of dictionary

As described above, the dictionary is implemented as an option set in DHIS 2, with the medical term as the option name and a special format for the option code that consists of the ICD-SMoL code, ICD-10 code and an ID separated by the | (pipe) symbol. SQL-views are included to verify that all entries in the dictionary follow this format:

- Dictionary entries with invalid code format - list entries where the code does not follow the format of `[content]|[content]|[content]`.
- Dictionary entries with invalid ICD-SMoL reference - list entries where the ICD-SMoL-part of the code is not in the ICD-SMoL list, which is provided as a separate option set.
- Dictionary entries mixing main- and sub-categories - list entries where, for the same ICD-SMoL code, some entries link to the main ICD-SMoL code and some to the specific code (e.g. some to 5-4 and 5-4.1). This can create misleading outputs when analysing data within DHIS 2 as deaths might be double counted.

Adding terms to the dictionary

An SQL view "Terms not found in dictionary" is included that will list all medical terms that have been entered as free text, i.e. which were not found in the dictionary during data entry, and the number of times they have been entered. These are terms that should be considered for inclusion in the dictionary if they occur frequently and are meaningful.

Before a term can be added to the dictionary, the correct ICD-SMoL and ICD-10 code must be identified. Together with the ID, these make up the code to be used in the dictionary. To find the ID, an SQL view Next dictionary index number is included that will show the next ID to use. It has two columns, `termtype` and `nextvalue`. As discussed above, it is recommended that each country uses a county code for local additions to the dictionary, in which case that code will be listed under `termtype` and the next number to use in the `nextvalue` column. If no local terms have yet been added, use `[country code]00001` as the first ID, e.g. `GH00001` if using GH as country code. Once the first local term has been added, the SQL view will show the next ID to use at any time.

To ensure that the right terms are displayed first in the dictionary, it must be sorted by the length of the term. After adding new terms to the dictionary, it must be re-sorted so that the new terms are not always last in the list. This can be done with the following SQL query (for PostgreSQL databases):

```
with ordered as (  
  select  
    uid,  
    name,  
    code,  
    row_number() OVER (ORDER BY length(name),name) AS sort_order  
  from optionvalue  
  where  
    optionsetid =  
      (select optionsetid from optionset where code = 'ICD_SMOL_DICTIONARY')  
)
```

```
update optionvalue ov
  set sort_order = ord.sort_order
from ordered ord
where
  ov.uid = ord.uid and
  ov.optionsetid =
    (select optionsetid from optionset where code = 'ICD_SMOL_DICTIONARY');
```

Note that this is an UPDATE query that must be executed directly in the database, not through the DHIS2 user interface.

Upgrading DHIS

When new versions of DHIS2 are released, it is important to test all functionality of the module on a staging/test server before upgrading any production instances of system.