DHIS 2 configuration guide for the Android capture app

Applicable to version 2.0.0

DHIS 2
January 2020
Copyright © 2006-2020 DHIS 2

January 2020

 Revision History
master@

**Warranty:** THIS DOCUMENT IS PROVIDED BY THE AUTHORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHORS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS MANUAL AND PRODUCTS MENTIONED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

**License:** Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the source of this documentation, and is available here online: [http://www.gnu.org/licenses/fdl.html](http://www.gnu.org/licenses/fdl.html)
Table of Contents

1 About this guide
2 Introduction
3 Recommendations for a DHIS 2 mobile deployment
   3.1 Mobile device specifications
   3.2 Testing and Piloting
   3.3 How to migrate to Android Capture App
4 Contribute to the App
   4.1 How to report a bug
   4.2 How to suggest Icons to the library
   4.3 How to add translations
5 Visual configurations
   5.1 Meta data sync error
   5.2 Personalize your Icon
   5.3 Personalise the colour of the App
   5.4 Icon library for metadata
   5.5 Colour palette for metadata
   5.6 Rendering Types for Program Sections
   5.7 Calendar
6 Settings
7 Generic features
   7.1 Login
   7.2 Account Recovery
   7.3 Blocking session (PIN)
   7.4 Instructions/ information buttons
   7.5 Filter
   7.6 Sync Information
      7.6.1 Granular Sync
      7.6.2 SMS Sync
   7.7 Org unit
   7.8 Data Sets
   7.9 Differentiating Data Sets, Tracker and Event programs
   7.10 Sharing Data
   7.11 Capture Coordinates
      7.11.1 TEI coordinates
      7.11.2 Polygons
   7.12 Images
   7.13 Display events and TEIs in maps
8 Programs Specific features
   8.1 Program with registration specific features
      8.1.1 Complete/Re-open event
      8.1.2 Complete/Deactivate Enrollment
      8.1.3 Reset Search Fields
      8.1.4 Search screen for all Tracked Entity Type
      8.1.5 TEI Dashboard across programs
      8.1.6 TEI enrollment history and new enrollment
      8.1.7 Delete TEI's & Enrollments
   8.2 Program without registration specific features
      8.2.1 Complete event and Field completion percentage
9 Features supported
   9.1 Data Sets features
   9.2 Program features
   9.3 Program stage features
Table of Contents

9.4 Program with registration: Tracked entity dashboard features
9.5 Program without registration: Single event program specific features
10 Program rules supported in Android Capture App
   10.1 Program rule Variable source types supported
   10.2 Program rule Actions supported (Data element in current event)
   10.3 Program rule Actions supported (Other variables)
   10.4 Functions to use in program rule expressions
   10.5 Standard variables to use in program rule expressions
11 Program Indicators
   11.1 Variables to use in a program indicator expression or filter
12 Value types supported
1 About this guide

The DHIS 2 documentation is a collective effort and has been developed by the development team and users. While the guide strives to be complete, there may be certain functionalities which have been omitted or which have yet to be documented. This section explains some of the conventions which are used throughout the document.

DHIS 2 is a browser-based application. In many cases, screenshots have been included for enhanced clarity. Shortcuts to various functionalities are displayed such as Data element > Data element group. The “>” symbol indicates that you should click Data element and then click Data element group in the user interface.

Different styles of text have been used to highlight important parts of the text or particular types of text, such as source code. Each of the conventions used in the document are explained below.

- **Note**
  A note contains additional information which should be considered or a reference to more information which may be helpful.

- **Tip**
  A tip can be a useful piece of advice, such as how to perform a particular task more efficiently.

- **Important**
  Important information should not be ignored, and usually indicates something which is required by the application.

- **Caution**
  Information contained in these sections should be carefully considered, and if not heeded, could result in unexpected results in analysis, performance, or functionality.

- **Warning**
  Information contained in these sections, if not heeded, could result in permanent data loss or affect the overall usability of the system.

- **Complete**
  Information contained in these sections, will indicate that these are issues that have been fully implemented.

- **Incomplete**
  Information contained in these sections, will indicate that these are issues that are not implemented and will be ignored.

- **N/A**
  Not_applicable
N/A

Information contained in these sections, will indicate that these are issues not applicable.

[ ] Work_in_progress

Information contained in these sections, will indicate that these are issues or features not completely implemented or with unexpected behaviour already reported.

Program listings usually contain some type of computer code. They will be displayed with a shaded background and a different font.

Commands will be displayed in bold text, and represent a command which would need to be executed on the operating system or database.

Links to external web sites or cross references will be displayed in blue text, and underlined like this.
2 Introduction

This guide is designed to assist you in configuring DHIS 2 for use with the Android Capture App.

It is not a tutorial or training manual; it is designed to complement the core DHIS 2 documentation and training materials with specific reference material for the Android application.

The document is divided into sections, which are intended to describe all the features that have been implemented in the app, those that have not yet been implemented and those that will not be supported. Where applicable, an explanation of problems with the use of a particular feature in Android app will be provided. Recommendations or tips will also be presented for a better understanding and use of the app.

In addition, the document has sections for new features in the login, visual configurations, synchronization management and a list of known important errors.
3 Recommendations for a DHIS 2 mobile deployment

If you plan to deploy the DHIS2 Android App in the field, we strongly recommend you to read the Mobile Implementation Guidelines published by UiO. It includes chapters on technology requirements, security and configuration aspects, and testing and roll out recommendations. Below you will find some key aspects briefly introduced, we recommend reading the extended document.

3.1 Mobile device specifications

The Android App is compatible and supported for DHIS 2 versions 2.30, 2.31, 2.32, and 2.33. And has no breaking changes with 2.29.

It requires a device that is running Android v4.4 or higher.

In this link you will find recommendations for new mobile device acquisitions for a Dhis2 Android deployment.

3.2 Testing and Piloting

If you plan to deploy the DHIS 2 Android App in the field, you should first do a full round of testing of the app in your own configuration.

The app has been extensively tested with the demo servers, and during Beta testing it was tested against some real configurations as well. We know, however, that every DHIS 2 configuration is special in many senses, and might cause inconsistencies that we have not being able to identify.

It is strongly advised to carry out a comprehensive testing of the app in your own server before piloting it.

3.3 How to migrate to Android Capture App

If you are ready for deploying the new Android App in the field and your users are already using Event Capture or Tracker Capture, you should follow these steps:

1. Sync data of the current app you are using

   **Warning**
   Deleting the app without syncing can cause information loss.

2. Download and install the DHIS 2 Android App

3. Login using your credentials and all data will be synced.
4 Contribute to the App

4.1 How to report a bug

First of all, please check the [List of already reported bugs](#).

The list is always updated as you add new bugs and we work on them. You need an active, free DHIS 2 Jira account.

When reporting a bug please specify together with your description:

- DHIS 2 version and build revision
- DHIS 2 Android App version
- Android OS version
- Device (when possible)
- Error log (when possible)

4.2 How to suggest Icons to the library

DHIS 2 and the new DHIS 2 Android App allows you to select icons for programs, datasets and the options of an optionset. DHIS 2 2.30 incorporates 125 icons. We plan to expand the collection of icons on each future version of DHIS 2- for that we need your help: we are seeking ideas for new icons - please submit yours using [this form](#).

4.3 How to add translations

The steps to contribute to the translation of the app are:

1. Create an account in [Transifex](#)
2. Join Existing Organization
3. Search for HISP UiO and select the organization not the projects
4. Select ANDROID: Capture project
5. Join the team for the selected language

Please let us know if it doesn't work well or if you need to add a new language by sending a message in the community to the user dhis2-android.
5 Visual configurations

5.1 Meta data sync error

In case of errors during the sync process, a message is displayed in the settings menu (‘Sync data’ or ‘Sync configuration’ section). Also, a red sync icon is displayed next to the program in the Home screen. The sync error log gives details about the error and is prepared to be shared with admins.

You can also open the sync errors log from Settings:

5.2 Personalize your Icon

You can set your icon in the wallpaper using the DHIS 2 App Widget. The icon will be the flag configured in your server.
5 Visual configurations

5.3 Personalise the colour of the App

You can set the generic theme of the app by configuring the theme in our server:

Note
How to add a widget:

- Long press in your wallpaper
- Select Widgets
- Look for the DHIS 2 Widget
- Select

It will add a shortcut to your app.

Note
How to set the server theme and flag:
Go to System Settings > Appearance > Style

- Select your style
  - Green,
India (Orange),
Light Blue,
Myanmar (red),
Vietnam

• Select your flag

5.4 Icon library for metadata

There are currently 134 icons available that you can choose from. You can search by name in the icon picker widget. Icons are available in positive, negative and outlined shapes. We will be adding many more icons in future releases.

Icons can be associated to programs, stages, data elements and the options of an option set. This is configurable through the Maintenance App.

Note
This feature is available from DHIS 2 2.30 onwards

In the Android App icons render in the home screen to illustrate all the programs available to a user, or while adding an stage. For data elements and options, the icons render for single event programs, when a section contains a single DE, showing the option set as a matrix or sequence of images.
Note

Icons will initially only render in the new Android app. The new web-based Capture App will incorporate them in the near future.

We plan to expand the collection of icons on each future version of DHIS2- for that we need your help: we are seeking ideas for new icons - please submit yours using this form.

5.5 Colour palette for metadata

Tracker and Event capture now have the ability to render colours for programs, stages, data elements and options in option sets. A colour picker is integrated in the Maintenance App, which shows as a palette, except for options which allows the selection of any colour.

In the Android App, the color will be rendered as background color for programs and stages combined with an icon (if selected). For options it renders as the background colour during data entry form for single event programs. When the user selects a program with an assigned colour, that colour becomes the background theme for all screens in the domain of the program.
5.6 Rendering Types for Program Sections

**Program Section:** when used in combination with icons, a Program Section with a single data element and associated Option Set can render the options in sequential or matrix layout (see screenshots).

**Note**
Colours will first be available in the new Android app, followed in future releases by the new web-based Capture App.
Note

Render type for sections will first be available in the Android app, followed in future releases by the new web-based Capture App UI.

5.7 Calendar

Now it is possible to use two different calendars. The one on the left is the first to appear but you can change it by clicking on the calendar icon on the lower left corner.

This is available when:

1. Making a new enrollment.
2. Creating a new event (Programs with and without registration).
3. Using period filters in data sets and programs with and without registration.
Note
For Android versions 4.4, 5.1 and small devices, the Accept option is not visible in the second calendar view.
6 Settings

Data sync process: User is able to choose from a drop-down list if the data entered in the app is uploaded to the server manually or scheduled every 15 min, 1 hour or 1 Day. By default the app will sync every 24h. This kind of Syncing includes new and updated events and TEI's.

Configuration sync process (Metadata syncing): User is able to choose from a drop-down list if the data entered in the app is uploaded to the server manually or scheduled every day or week. By default the app will sync every 24h. This sync will update changes in programs or configurations in web.

Sync Parameters: these parameters allow the user to specify the maximum number of TEI's and events that can be stored in the local device. The user can also specify if limits apply per organisation unit or in total. Values can be set to default by clicking on “Reset to default”.

Reserved Values: this will specify the number of reserved Id's available in your device and will allow you to refill them.

Open sync errors log: The sync error log gives details about the error and is prepared to be shared with admins.

Delete Data: this action will delete local data without having to log out.

Reset App: this action will log out the user and delete all data and metadata. It is similar to resetting the App.
7 Generic features

7.1 Login

There are two ways to access the application:

1. Manual: The user must type the corresponding URL of the server to be used and enter the username and password.

   ![Image of manual login](image)

   **Note**
   Take note that the login only possible with 2.29, 2.30 and 2.31 servers.

2. QR: The user can use a QR code instead of typing the URL. Username and password must be entered manually.

   ![Image of QR login](image)

   **Note**
   After the first login, the app will suggest URL and username of all successful connections.
   You are able to make an *offline* login only if using the same user as the last online session.

![Image of QR login with warning](image)

**Warning**
If a user attempts an on-line login and their account has been disabled as explained in the [DHIS 2 Manual - Disable User](link) all the data will be wiped from the phone. Make sure that before disabling a user all the data has been synced or that you are using this feature to remotely wipe sensitive data in case of a device getting lost.

7.2 Account Recovery

The users will be able to restore their own password if they have the setting enabled: Enabled User Account Recovery.
7.3 Blocking session (PIN)

User is able to lock the session using a 4 digit PIN. This allows to move to other apps in your phone without deleting the local data. If the user forgets the PIN number, log in by entering the credentials is also available.

7.4 Instructions/ information buttons

A contextual guide opens the first time the user opens the App.
7.5 Filter

The app has new and improved filters for all listing screens (home, event listing, tei search and dataSets).

Filter by period, org. Unit, sync state, event status and category option combination.
Filters will adapt to the different programs and data sets. 1. Program without registration: Period, Org. Unit, Sync State, Event Status and, Category Combination. 2. Program with registration: Period, Org. Unit, and Sync State. 3. Data Sets: Period, Org. Unit, and Syn State.

7.6 Sync Information

Allows the user to check sync information for a specific program.
7 Generic features

7.6.1 Granular Sync

Allows to sync individual records (Program, Event, TEI, DataSet, DataValues) with the server.

7.6.2 SMS Sync

When there is no internet connection, it allows to send a record through several SMS messages. The record is marked as “SMS synced”.

Tip
Edit parameters related to SMS gateway in the SMS Settings (Settings Menu)
7.7 Org unit

The whole organisation unit tree is displayed. Organisation units not available for data entry will be colored in grey. User must check the box to select the org unit wanted.

Caution

Mobile users are not expected to access the org. unit hierarchy of a whole country. Maximum number of org units is difficult to set, as the App does not set the limit, but the resources on the device (memory, processor). We could say below 250 org units should be safe, but still believe that is a very big number for a mobile use case.

7.8 Data Sets

The user can now enter aggregate data for an organisation Unit, a period and a set of data elements and send it to the server.
7.9 Differentiating Data Sets, Tracker and Event programs

An easy way to differentiate them is by looking at the word at the bottom left corner. The word ‘Event’ will always be in event programs. In tracker will appear the name of the tracked entity type (person, patient, building, etc.). For data sets, the word ‘DataSets’ will be shown next to the number of records.

7.10 Sharing Data

7.11 Capture Coordinates

7.11.1 TEI coordinates

Capture the TEI coordinates in the registration form. Enable this feature in the TET feature type.
7.11.2 Polygons

The app now supports the geoJSON format and the user is able to capture polygons.

7.12 Images

Value Type image has been implemented app side. This allows to pick an image for a data element or attribute and send it to the server. For TEIs, the first data element/attribute with this value type and marked as displayed in list will be used as the TEI profile image.
7.13 Display events and TEIs in maps

When a program stage or tracked entity type have a feature type (and for programs with registration the option displayFrontPageList is enabled) the listings can be switched to display the information in a map.
8 Programs Specific features

8.1 Program with registration specific features

8.1.1 Complete/Re-open event

User must enter the event and then click the ‘END’ button in the bottom right corner.

Two options will be presented:

1. Finish and Complete
2. Finish

Note

To verify if an event is completed look at the color of the box, it must be grey; and also have the label ‘Event Completed’ on it. It applies the same legend set than the web UI.
The app needs to take into consideration if the user has the correct authority (‘Uncomplete events’) to reopen a completed event.

### 8.1.2 Complete/Deactivate Enrollment

To complete or deactivate an enrollment, click on ‘See details’ and actions will be available in the ‘Lock’ icon on the top right corner of the screen.

![Image of a user interface showing 'Complete' and 'Deactivate' options]

#### 8.1.3 Reset Search Fields

All tracker programs will take the user to the search screen. The search fields are used to look up for a specific entity, and the circled arrow at the top right corner to reset the search. All fields will be blank for the user to do a new search.

At first, the user is obligated to do a search. If there are no coincidences, the search button will change to an ‘Add’ button for the user to create a new enrollment.

![Image of a search screen with fields and options]

#### 8.1.4 Search screen for all Tracked Entity Type

User is able to search across all program of one tracked entity type (TET). In the Search screen there is a drop down which shows all the programs available for the active TET (active TET is
defined by the selection of the program in the home screen). That drop down should also have an option with TET name. (for example: person)

When the user selects that option, the search fields available will only be the TET attributes (no program specific attributes). Search restrictions do not apply, because they belong to the programs.

The search will return the found TEI's in the local database and also those in the Search OU of the user (when user is online). For those found online, the user will have to select them and the whole record will be downloaded.

**Note**

When configuring search org. units, make sure that your capture org. units are contained in your search org. units, to do that capture org. units have to be selected as well as search org. units.

### 8.1.5 TEI Dashboard across programs

User can see the TEI dashboard without any program by selecting the TEI in the list if the search was without program.

The dashboards will show the list of active enrollments.
8.1.6 TEI enrollment history and new enrollment

User is able to see the complete historical record of the TEI. By clicking on the top right corner icon they will see a list of Active enrollments, followed by a list of past enrollments (completed or cancelled), followed by the programs in which the TEI could be enrolled. User can also return to the ‘TEI Dashboard without any program’ by selecting ‘All enrollments’.

Users should be able to navigate to the different enrollments from the list.

8.1.7 Delete TEI's & Enrollments

To delete a TEI or an enrollment, select in the TEI dashboard, the three dots menu.

Local TEI or Enrollment will be deleted from the database. Records that has been previously synced with the server will be marked for deletion if the user has the authority:

F_ENROLLMENT_CASCADEDELETE
F_TEI_CASCADEDELETE

They will show in the TEI search list, but will not be accessible.
8.2 Program without registration specific features

8.2.1 Complete event and Field completion percentage

The percentage of data completed in each event is shown in the top right corner of an event when it is opened after first registration.

To complete an event click on the ‘END’ button. Two options will be offered

1. Finish and Complete
2. Finish
Note

The percentages will be adapted according to the effects of the program rules in the forms.
9 Features supported

The following is a comprehensive list of all features available for Data Sets, Programs with and without registration in DHIS2, and notes on whether or not these have been implemented in the Android Capture app.

In the notes, ‘admin’ refers to someone who develops and configures a DHIS2 system, and ‘user’ refers to someone who uses apps to capture data, update it, and review reports.

<table>
<thead>
<tr>
<th>Legend</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Feature implemented</td>
</tr>
<tr>
<td>☐</td>
<td>Feature not implemented (will be ignored)</td>
</tr>
<tr>
<td>N/A</td>
<td>Not applicable</td>
</tr>
<tr>
<td>⌚</td>
<td>Work in progress. Feature not completely implemented yet or with unexpected behaviour already reported.</td>
</tr>
</tbody>
</table>

9.1 Data Sets features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description of feature</th>
<th>Status</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Period type</td>
<td>Determines the period covered by data-entry.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Expiry days</td>
<td>Sets the deadline (days after period) after which DHIS2 locks all data entry for the period (0 means no locks at all). Periods can still be opened, but cells will be greyed out.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Open future periods for data entry</td>
<td>This setting can be used to unlock the current period or all periods up to a certain point in the future.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Data input periods</td>
<td>Allows you to set a specific range of dates for a period's data-entry, and prevents data capture for periods outside of this date range.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Days after period to qualify for timely submission</td>
<td>Sets the deadline (days after period) after which DHIS2 treats data entry as ‘late’.</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>[Attribute] category combination</td>
<td>Allows an admin to attach a Category (set of Options) to the Data Set, generating a separate data entry screen for each Option (This is called an Attribute Category Combination in DHIS2).</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>[Attribute] Category Combination option restrictions</td>
<td>If Attribute Category Combinations (see above) are used, then this features gives admins the ability to restrict which specific options are available in the drop-down. Each option can be restricted to a particular range of dates and/or organisation units, and that option will not be shown if data is being captured outside of these dates or org units.</td>
<td>☑</td>
<td>ANDROAPP-1153 Restriction is only possible using dates.</td>
</tr>
<tr>
<td>Complete notification recipients</td>
<td>Sends a DHIS2 message to the selected User Group when the Data Set is marked 'complete'.</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>Send notification to completing user</td>
<td>Sends a DHIS2 message to the data-entry user when the Data Set is marked ‘complete’.</td>
<td>☐</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>All fields for data elements required</td>
<td>Where one or more Categories are used to disaggregate a data element, this setting forces the user to either complete all disaggregations, or to leave them all empty.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Complete only if validation passes</td>
<td>Only allows the Data Set to be marked complete if no Validation Rules are triggered.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Skip offline</td>
<td>Requires the user to add a ‘comment’ if a value is left blank (or Data Set cannot be ‘completed’). Only allows Data set to be selected for data entry while connected to the internet (although once selected, data entry can continue offline).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data element decoration</td>
<td>Shows the description of the Data Element when the mouse hovers over the Data Element Name.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Section forms - render sections as tabs</td>
<td>Displays each section in the form as a separate tab, instead of all together on the same page.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Section forms - render vertically</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Data Element - CatCombos</td>
<td>Enables an individual Data Element to be automatically disaggregated into one or more categories (e.g., both male/female and child/adult), with separate fields/values collected for each of these disaggregations.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Inline indicators / section form totals</td>
<td>Enables the default CatCombo for each data element to be overridden with a different CatCombo for this Data Set only.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Section forms - render sections as tabs</td>
<td>Adding indicators to Data Sets makes them available for use in section forms and custom forms; row and/or column totals can also be added to these forms. (Both are displayed on the screen alongside data capture cells, and update automatically as values are captured.)</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Organisation unit assignment</td>
<td>Ensures the Data Set is only available for those organisation units to which it has been assigned.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Compulsory data elements</td>
<td>This allows the marking of specific Data Elements/CatCombos as ‘compulsory’, which means users must enter a value (they cannot be left blank).</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Forms - default forms</td>
<td>DHIS2 automatically renders the form as table(s), with a new table started every time the Category Combinations change (= different column headings).</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Forms - section forms</td>
<td>Form sections and section titles can be specified, giving you more control over the grouping and layout of the form (but it is still rendered automatically). This section form automatically overrides the default form if implemented.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Forms - custom forms</td>
<td>A custom HTML form can be designed, giving total control over layout, and enabling JavaScript code to be included within the form. This custom form automatically overrides the default and section forms if implemented.</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Section forms - disable fields (grey)</td>
<td>With section forms, this enables you to individually 'grey out' fields (a whole data element, or specific CatCombo Options) so that users cannot enter data into it.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Multi-organisation unit forms</td>
<td>When this server setting is enabled, form layouts are changed to show multiple org units as rows, and all data elements/CatCombos as columns (ie very flat and wide form per org unit).</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Data value pop-up: mark value for follow-up</td>
<td>Enables the user to mark this specific data value for follow-up (marked values can be reviewed in the Data Quality web app).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data value pop-up: add comment to value</td>
<td>Enables the user to add a comment to this specific data value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data value pop-up: display data element history</td>
<td>Shows the history over time of this specific data element (i.e. previous 12 months' values).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data value pop-up: display audit trail</td>
<td>Shows a history of previous edits to this specific data value.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data value pop-up: min/ max ranges (also accessible via Data Administration app)</td>
<td>This enables users to set minimum and maximum expected values for a data element, enabling DHIS2 to highlight values outside this range during data entry (but it does not prevent saving or ‘completing’). You can set min/max ranges automatically/in-bulk (via the Data Administration app) or manually/individually (via the Data Entry app).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Print form / print blank form</td>
<td>Enables printing of a data-entry form, to permit data capture on paper, and data-entry later.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Save data</td>
<td>Data entered into the screen is not captured until ‘saved' - until then, it is only held in memory, and is lost if power is switched off etc.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Complete data set</td>
<td>This enables user to mark data-entry for a period/org unit/etc as ‘complete’. Note that this is just for data-entry tracking and timeliness purposes, and does not lock the data set or prevent further edits.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data elements: validation</td>
<td>Enables the creation of rules (at the data element level) to enforce data quality, based on comparing different values/collections of values. (Eg number of patients seen in the month must be less than the number of visits for the month.)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/Can capture data</td>
<td>Enables the user to add new values, edit values and delete values in the dataset.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/Can view data</td>
<td>Enables the user to see values within the dataset.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/No access</td>
<td>The user won’t be able to see the dataset.</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data approval workflow</td>
<td>If an admin selects a pre-configured Data Approval Workflow, this will be used to enforce an ‘approval’ or ‘acceptance and approval’ cascade, enabling users to sign-off and lock data.</td>
<td>✓</td>
<td>The process of the approval has to be done in web. Once a data set is approved, the data will not longer be editable in the app.</td>
</tr>
</tbody>
</table>
### 9.2 Program features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description of feature</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data entry method for option sets</td>
<td>Enables an admin to choose how options will be displayed on-screen across the entire program (ie either as drop-down lists or as radio buttons)</td>
<td></td>
<td></td>
<td>This will be replaced by the new rendering options.</td>
</tr>
<tr>
<td>Combination of categories (Attribute CatCombo)</td>
<td>Allows an admin to attach a Category (set of Options) to the Program, requiring users to categorize each enrolment. (This is called an Attribute Category Combination in DHIS 2.)</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Data approval workflow</td>
<td>If an admin selects a pre-configured Data Approval Workflow, this will be used to enforce an ‘approval’ or ‘acceptance and approval’ cascade, enabling users to sign-off and lock data.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display front page list</td>
<td>If this option is ticked, the landing page displays a list of active enrolments once an Org Unit and Program have been chosen. (Attributes shown are those ticked as ‘display in list’.)</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>First stage appears on registration page</td>
<td>When this option is chosen, then during Program enrolment, the screen for the first Program Stage is also shown (enrolment and the first event are captured together on one screen).</td>
<td>☑</td>
<td>N/A</td>
<td>In Android, this is implemented by opening automatically the event after enrollment is completed, instead of adding the form to the same screen.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Completed events expiry days</td>
<td>Enables admins to lock data-entry a certain number of days after an event has been completed.</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Expiry period type + expiry days</td>
<td>Enables admins to set a period (eg weekly, monthly), and to lock data-entry a certain number of days after the end of the period.</td>
<td>✔️</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Allow future enrolment dates</td>
<td>If ticked, this enables a user to enter future Enrolment dates during enrolment in a Program; otherwise users are restricted to today or past dates.</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Allow future incident dates</td>
<td>If ticked, this enables a user to enter future Incident dates during enrolment in a Program; otherwise users are restricted to today or past dates.</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Only enrol once (per tracked entity instance lifetime)</td>
<td>If ticked, prevents a TEI (eg person) from being enrolled in this Program more than once.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Show incident date</td>
<td>If ticked, both Enrolment and Incident dates are shown to the user for data capture; otherwise, only the Enrolment date is shown/captured.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Description of incident date</td>
<td>Allows an admin to customize the label that is used for the incident date.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Description of enrolment date</td>
<td>Allows an admin to customize the label that is used for the enrollment date.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Capture coordinates (enrolment)</td>
<td>Enables users to capture geographical coordinates during enrolment in the program.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Capture Polygon (enrolment)</td>
<td>Enables users to capture locations (enclosed areas) during enrolment in the program.</td>
<td>☑</td>
<td></td>
<td>N/A</td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>TEI Coordinates</td>
<td>Enables users to capture geographical coordinates for the TEI during the enrolment in the program.</td>
<td>✔️</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Relationships: create and update</td>
<td>Enables users to create and update relationships.</td>
<td>✔️</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Relationships - shortcut link to add a relative</td>
<td>This enables admins to add a link for one specific relationship to the Dashboard, enabling users to directly create a linked TEI (e.g., “child” patient).</td>
<td>❌</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Attributes: display in list</td>
<td>This setting determines whether an Attribute can be viewed in lists such as search results, and whether it can be seen in the shortlist of Attributes shown under “Profile” in the Dashboard.</td>
<td>✔️</td>
<td>N/A</td>
<td>The first three attributes will be shown</td>
</tr>
<tr>
<td>Attributes: mandatory</td>
<td>This enables an admin to mark an Attribute as “mandatory”, meaning the enrolment can’t be saved until a value is captured.</td>
<td>✔️</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>------------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Attributes: date in future</td>
<td>For date Attributes, this enables an admin to either prevent or allow future dates to be captured.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Registration form - default</td>
<td>The default data entry form simply lists all attributes defined for the TEI.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Registration form - custom</td>
<td>This enables an admin to define a custom layout (using HTML) for the registration form.</td>
<td>-</td>
<td>N/A</td>
<td>Custom layouts are not supported in the Android App</td>
</tr>
<tr>
<td>Program notifications</td>
<td>You can set up automated notifications for when program enrolments or completions occur, or at a set interval before/after incident or enrolment dates. These can be sent as internal DHIS 2 messages, emails or SMSs.</td>
<td>✓</td>
<td>✓</td>
<td>This functionality is executed on the server side, once data is received. Will not work when the app is working offline.</td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Activate/deactivate enrolment</td>
<td>Deactivating a TEI dashboard will cause the TEI to become “read-only”. This means you cannot enter data, enrol the TEI or edit the TEI profile.</td>
<td>✔</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Complete allowed only if validation passes</td>
<td>Select check box to enforce that an event created by this program is only completed when all validation rules have passed.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Org unit opening/closing dates</td>
<td>Enables an admin to set opening and closing dates for an Org Unit, which blocks users from adding or editing events outside of these dates.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/Can capture data</td>
<td>Enables the user to add new event, edit data and delete events in the program.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/Can view data</td>
<td>Enables the user to see list of events within the program.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Data sharing levels/No access</td>
<td>The user will not be able to see the program</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Event form - default</td>
<td>The default data entry form simply lists all attributes belonging to a program registration</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Event form - section forms</td>
<td>Sections forms allow you to split existing forms into segments</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Event form - custom</td>
<td>Define a custom event form as a HTML page.</td>
<td></td>
<td></td>
<td>Custom layouts are not supported in the Android App.</td>
</tr>
<tr>
<td>Program stage notifications</td>
<td>You can set up automated notifications for when the program stage is completed, or at a set interval before/after scheduled event dates. These can be sent as internal DHIS 2 messages, emails or SMS messages.</td>
<td>✓</td>
<td>N/A</td>
<td>This functionality is executed on the server side, once data is received. Will not work when the app is working offline.</td>
</tr>
<tr>
<td>Repeatable</td>
<td>If Repeatable is ticked, this stage can be repeated during one program enrollment. If it is not, then the stage can only happen once during a program enrollment.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Repeatable</td>
<td>If Repeatable is ticked, this stage can be repeated during one program enrollment. If it is not, then the stage can only happen once during a program enrollment.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Repeatable + Standard interval days</td>
<td>The system will suggest the due date as the calculation of the last event + standard interval dates.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Period type</td>
<td>Enables an admin to configure a set of periods (e.g. weeks or months) for each event in the program stage, instead of just a date. When creating events, users are then asked to choose a period (instead of a date) for each new event they create within that program stage.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Auto-generate event</td>
<td>If ticked, a “booking” is generated for this Program Stage upon enrolment, based on the “Scheduled days from start”.</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Generate events based on enrolment date</td>
<td>Check on it for auto-generating due dates of events from program-stages of this program based on the enrollment date. If it is not checked, the due dates are generated based on the enrolment date.</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Open data entry form after enrolment + date</td>
<td>If selected, once an enrolment is complete, an event’s data entry form should open directly afterwards.</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Ask user to complete program when stage is complete</td>
<td>If selected, upon completing the program the user should be asked to complete the program. (This setting is ignored if “Ask user to create new event” is also ticked.)</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Ask user to create new event when stage is complete</td>
<td>If selected, when the Program Stage is completed the user is prompted to book.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Hide due date</td>
<td>Only shows the actual date for events, hiding the due date.</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Capture coordinates (event)/ Feature Type-Point</td>
<td>Enables the user to capture geographical coordinates when each event is created particularly useful in devices that have GPS (eg Android), as instead of having to type in coordinates, the user can automatically populate them with the press of a button.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Capture Polygon (event)/ Feature Type-Polygon</td>
<td>Enables users to capture locations (enclosed areas) when each event is created. A Polygon must contain at least 4 points.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Description of report date</td>
<td>Allows an admin to customize the label that is used for the event's date.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Data elements- compulsory</td>
<td>This enables an admin to mark a data element as “compulsory”, meaning an event can not be saved until a value is captured.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data elements- allow provided elsewhere</td>
<td>On the form, this places a tick-box next to the selected data element, and enables previous data to be pulled into the data element.</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Data elements- display in reports</td>
<td>Displays the value of this data element into the single event without registration data entry function.</td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data elements- date in future</td>
<td>For date Data Elements, this enables an admin to either prevent or allow future dates to be captured.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Data elements- render options as radio</td>
<td>Enables an admin to choose how options will be displayed on-screen for each Data Element (i.e. either as drop-down list or as radio buttons).</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### 9.4 Program with registration: Tracked entity dashboard features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description of feature</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block entry form after completed</td>
<td>Prevents all edits to events after they have been completed.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Event comments</td>
<td>Enables the user to add overall comments to an event. These comments are cumulative (new comments are added below existing comments).</td>
<td></td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Messaging</td>
<td>Enables users to send ad-hoc free-text messages to TEIs (e.g. patients) via SMS or email.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark for follow-up (button with exclamation triangle)</td>
<td>Enables a user to mark a TEI (e.g. patient) as requiring follow-up.</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Display TEI audit history</td>
<td>Enables a user to see a history of all edits to Attributes for this TEI (e.g. patient).</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inline Program Indicators</td>
<td>If a program indicator “display in form” box is ticked, the indicator appears on the Tracker Capture dashboard, and is updated live as data capture occurs.</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete events</td>
<td>Enables the user to delete an event.</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>Schedule events</td>
<td>In the event generation dialogue, the user should also see the option to schedule an event. The process is like creating an event, but the user will be sent back to the TEI dashboard after the event is scheduled.</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Referral of patients</td>
<td>In the event generation dialogue, the user should also see the option to refer a patient. The process is like creating/scheduling an event, but the user can change the org unit and has to specify if it is a one-time or permanent referral. One time will just create the event in the specified OU.</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reset search fields</td>
<td>User is able to clean up the search fields by pressing on the rounded arrow icon on the top right corner of the search screen.</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------</td>
<td>--------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>Search screen for all TE Type</td>
<td>User is able to search across all program of one tracked entity type (TET). In the Search screen there is a drop down which shows all the programs available for the active TET (active TET is defined by the selection of the program in the home screen). That drop down should also have an option with TET name. (Person in our server). When the user selects that option, the search fields available will only be the TET attributes (no program specific attributes). All search restrictions do not apply, because they belong to the programs.</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TEI Dashboard without program</td>
<td>User can see the TEI dashboard without any program by selecting the TEI in the list if the search was without program. The dashboards will show the TET attributes in the details card followed by a list of active enrollments.</td>
<td>✔️</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>TEI enrollment history and new enrollment</td>
<td>User is able to see the complete historical record of the TEI. By clicking on the top right corner icon they will see a list of Active enrolments, followed by a list of past enrolments (completed or cancelled), followed by the programs in which the TEI could be enrolled. Users should be able to navigate to the different enrolments from the list.</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 9.5 Program without registration: Single event program specific features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description of feature</th>
<th>Status</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Events listing (grid)</td>
<td>A listing of existing events that is displayed once a program is selected.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Sort and filter events in grid</td>
<td>Allows the user to sort listed events, or to filter events based on keywords or specific ranges of dates/numbers.</td>
<td>✔</td>
<td>Events are sorted chronologically. The user can filter by period and organisation unit.</td>
</tr>
<tr>
<td>Edit events in grid</td>
<td>Allows the user to directly edit the data elements shown in the events listing/grid.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>View event audit history</td>
<td>Enables the user to see a history of all changes to the event's data elements.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show/hide columns (in event list/grid)</td>
<td>Enables the user to modify the data elements shown in the event listing/grid (applies to that user only).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature</td>
<td>Description of feature</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Field completion percentage</td>
<td>The percentage of data completed in each event is shown in the top right corner of an event when it is opened after first registration. The percentages should be adapted to the effects of the program rules in the forms.</td>
<td></td>
<td>The percentage of completion does not take into account the not-supported value types in the forms.</td>
</tr>
<tr>
<td>Delete events</td>
<td>Enables the user to delete an event.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10 Program rules supported in Android Capture App

The following is a comprehensive list of all Program rule components (variable types and actions) available in DHIS 2, and notes on whether or not these have been implemented in the Android Capture app.

Any issues around using a particular feature with Android are highlighted with an exclamation mark !.

<table>
<thead>
<tr>
<th>legend</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Value type implemented</td>
</tr>
<tr>
<td>☐</td>
<td>Value type not implemented, but will be safely ignored (if not compulsory)</td>
</tr>
<tr>
<td>N/A</td>
<td>Not applicable</td>
</tr>
<tr>
<td>🚧</td>
<td>Work in progress. Feature not completely implemented yet or with unexpected behavior already reported</td>
</tr>
</tbody>
</table>

10.1 Program rule Variable source types supported

<table>
<thead>
<tr>
<th>Variable type</th>
<th>Description of variable type</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data element from the newest event for a program stage</td>
<td>This source type works the same way as “Data element from the newest event in the current program”, except that it only evaluates values from a specific program stage.</td>
<td>✔️</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Data element from the newest event in the current program (with registration)</td>
<td>This source type is populated with the newest data value collected for the specified data element within the enrolment.</td>
<td>✔️</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Variable type</td>
<td>Description of variable type</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Data element from the newest event in the current program (without registration)</td>
<td>This program rule variable will be populated with the newest data value found within the 10 newest events in the same organization unit.</td>
<td>N/A</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Data element in current event (with registration)</td>
<td>Variable takes the data element's value from the current event.</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Data element in current event (without registration)</td>
<td>Contains the data value from the same event that the user currently has open.</td>
<td>N/A</td>
<td>☑</td>
<td></td>
</tr>
<tr>
<td>Data element from previous event (with registration)</td>
<td>Program rule variables with this source type will contain the newest value from all previous events for the specified data element. The event currently open is not evaluated.</td>
<td>☑</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Variable type</td>
<td>Description of variable type</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Data element from previous event (without registration)</td>
<td>This program rule variable will be populated with the newest data value found within the 10 events preceding the current event date (i.e. not including the current event).</td>
<td>N/A</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Tracked entity attribute</td>
<td>Populates the program rule variable with a specified tracked entity attribute for the current TEI (e.g. current patient).</td>
<td>✓</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>Calculated value</td>
<td>Calculated value.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

### 10.2 Program rule Actions supported (Data element in current event)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description of action</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Field</td>
<td>Hides an individual data element if the rule is true.</td>
<td>✓</td>
<td>✓</td>
<td>! If you change the value after the field is hidden, it will revert the action depending on the value type rule engine default value. We recommend its use combined with the hasvalue function.</td>
</tr>
<tr>
<td>Hide Section</td>
<td>Hides a whole section and its data elements if the rule is true.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of action</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>Hide Option</td>
<td>Hide a single option for an option set in a given data element/tracked entity attribute. When combined with <strong>show option group</strong> the <strong>hide option</strong> takes precedence.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hide Option Group</td>
<td>Hide all options in a given option group and data element/tracked entity attribute. When combined with <strong>show option group</strong> the <strong>hide option</strong> takes precedence.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show option group</td>
<td>Used to show only options from a given option group in a given data element/tracked entity attribute. To show an option group implicitly hides all options that is not part of the group(s) that is shown.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assign Value</td>
<td>Assigns a value to a specified data element or attribute if the rule is true.</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of action</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>----------------------------</td>
<td>-----------------------------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>Show Warning</td>
<td>Shows pop-up warning to the user if rule is true; does not prevent the user from continuing.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Warning on Complete</td>
<td>Shows a pop-up warning to the user if, at the point ‘complete’ is clicked, a rule is true; this does not prevent the user from continuing.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Show Error</td>
<td>Shows a pop-up error message to the user as soon as a rule is true, and prevents user from continuing until rule is no longer true.</td>
<td>✔</td>
<td>✔</td>
<td>The field will be cleared out so that the user is unable to store a value unless it meets the criteria of the program rule.</td>
</tr>
<tr>
<td>Error on Complete</td>
<td>Shows a pop-up warning to the user if, when “complete”; is clicked, a rule is true, and prevents user from continuing until rule is no longer true.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Make Field Mandatory</td>
<td>Sets a data element as “mandatory”; if rule is true.</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of action</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Display Text (Event Programs)</td>
<td>Used to display information that is not an error or a warning, for example feedback.</td>
<td>✓</td>
<td>✓</td>
<td>Independently from the source variable type, text will be displayed in the form as the last element of the last section. Text will be displayed as the messages in the indicators tab.</td>
</tr>
<tr>
<td>Action</td>
<td>Description of action</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Display Text (Tracker Programs)</td>
<td>Used to display information that is not an error or a warning, for example feedback.</td>
<td>✔</td>
<td>✔</td>
<td>1. Program rule configured as “Trigger rule only for program stage”. Text will be displayed ONLY in the form as the last element of the last section. Text will be displayed as the messages in the indicators tab. -&gt; If the program rule uses any variable type which is not from the current stage, the rule will not be able to evaluate and the message will not be shown. 2. Program rule NOT configured as “Trigger rule only for program stage”. Text will be displayed ONLY in the indicators tab and NOT in the form. -&gt; If the program rule uses any variable of type Current event, the rule will not be able to evaluate and the message will not be shown.</td>
</tr>
<tr>
<td>Action</td>
<td>Description of action</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| Display Key Value/Pair (Event Programs) | Used to display information drawn from a data element. | ✓ | ✓ | Variable Type:  
* Data element from the newest event in the current program  
* Data element from previous event  
* Data element in current event  
* Built-in variable  
Key/Value Pair will be displayed in the form ONLY in the specified section. |
| Display Key Value/Pair (Traker Programs) | Used to display information drawn from a data element. | ✓ | ✓ | 1. Variable Type:  
* Data element in current event  
Key/Value Pair will be displayed in the form ONLY in the specified section.  
2. Variable Type:  
* Data element from the newest event in the current program  
* Data element from previous event  
* Data element from the newest event for a program stage  
* Tracked entity attribute  
* Built-in variable  
Key/Value Pair will be displayed ONLY in the indicators tab and NOT in the form. |
<table>
<thead>
<tr>
<th>Action</th>
<th>Description of action</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Program Stage</td>
<td>Hides a whole program stage from the user if the rule is true.</td>
<td>N/A</td>
<td>N/A</td>
<td>Action rule only supported for Data element from the newest event in the current program type and tracked entity attribute variables.</td>
</tr>
<tr>
<td>Send Message</td>
<td>Send Message triggers a notification based on provided message template. This action will be taken whenever there is a change in data value. However this behaviour can be controlled by providing event-enrollment status in program rule expression</td>
<td>N/A</td>
<td>N/A</td>
<td>This feature is executed on the server side.</td>
</tr>
<tr>
<td>Schedule Message</td>
<td>Schedule Message will schedule notification at date provided by Expression in the data field.</td>
<td>N/A</td>
<td>N/A</td>
<td>This feature is executed on the server side.</td>
</tr>
</tbody>
</table>
### 10.3 Program rule Actions supported (Other variables)

<table>
<thead>
<tr>
<th>Action</th>
<th>Description of Action</th>
<th>Data Element from the Newest Event in the Current Program (with registration)</th>
<th>Data Element from the Newest Event in the Current Program (without registration)</th>
<th>Data Element from Previous Event (with registration)</th>
<th>Data Element from Previous Event (without registration)</th>
<th>Data Element from the Newest Event for a Program Stage (with registration)</th>
<th>Data Element from the Newest Event for a Program Stage (without registration)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hide Field</td>
<td>Hides an individual data element if the rule is true.</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Hide Section</td>
<td>Hides a whole section and its data elements if the rule is true.</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Hide Option</td>
<td>Hide a single option for an option set in a given data element/tracked entity attribute. When combined with show option group the hide option takes precedence.</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Hide Option Group</td>
<td>Hide all options in a given option group and data element/tracked entity attribute. When combined with show option group the hide option takes precedence.</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Assign Value</td>
<td>Assigns a value to a specified data element or attribute if the rule is true.</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
<td>☑</td>
</tr>
<tr>
<td>Action</td>
<td>Description of Action</td>
<td>Data Element from the Newest Event in the Current Program (with registration)</td>
<td>Data Element from the Newest Event in the Current Program (without registration)</td>
<td>Data Element from Previous Event (with registration)</td>
<td>Data Element from Previous Event (without registration)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Warning</td>
<td>Shows pop-up warning to the user if rule is true; does not prevent the user from continuing.</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Warning on Complete</td>
<td>Shows a pop-up warning to the user if, at the point “complete” is clicked, a rule is true; this does not prevent the user from continuing.</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Show Error</td>
<td>Shows a pop-up error message to the user as soon as a rule is true, and prevents user from continuing until rule is no longer true.</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of Action</td>
<td>Data Element from the Newest Event in the Current Program (with registration)</td>
<td>Data Element from the Newest Event in the Current Program (without registration)</td>
<td>Data Element from Previous Event (with registration)</td>
<td>Data Element from Previous Event (without registration)</td>
<td>Data Element from the Newest Event for a Program Stage (with registration)</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>---------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Error on Complete</td>
<td>Shows a pop-up warning to the user if, at the point “complete” is clicked, a rule is true; this does not prevent the user from continuing.</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Make Field Mandatory</td>
<td>Sets a data element as “mandatory” if rule is true.</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Display Text (Event Programs)</td>
<td>Used to display information that is not an error or a warning, for example feedback.</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>Display Text (Tracker Programs)</td>
<td>Used to display information that is not an error or a warning, for example feedback.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of Action</td>
<td>Data Element from the Newest Event in the Current Program (with registration)</td>
<td>Data Element from the Newest Event in the Current Program (without registration)</td>
<td>Data Element from Previous Event (with registration)</td>
<td>Data Element from Previous Event (without registration)</td>
<td>Data Element from the Newest Event for a Program Stage (with registration)</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Display Key Value/Pair (Event Programs)</td>
<td>Used to display information drawn from a data element.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Action</td>
<td>Description of Action</td>
<td>Data Element from the Newest Event in the Current Program (with registration)</td>
<td>Data Element from the Newest Event in the Current Program (without registration)</td>
<td>Data Element from Previous Event (with registration)</td>
<td>Data Element from Previous Event (without registration)</td>
<td>Data Element from the Newest Event for a Program Stage (with registration)</td>
<td>Data Element from the Newest Event for a Program Stage (without registration)</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
<td>------------------------------------------------------</td>
<td>-------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Display Key Value/Pair (Tracker Programs)</td>
<td>Used to display information drawn from a data element.</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
<td>✅</td>
</tr>
<tr>
<td>Action</td>
<td>Description of Action</td>
<td>Data Element from the Newest Event in the Current Program (with registration)</td>
<td>Data Element from the Newest Event in the Current Program (without registration)</td>
<td>Data Element from the Previous Event (with registration)</td>
<td>Data Element from the Previous Event (without registration)</td>
<td>Data Element from the Newest Event for a Program Stage (with registration)</td>
<td>Data Element from the Newest Event for a Program Stage (without registration)</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
<td>-----------------------------------------------------------------</td>
</tr>
<tr>
<td>Hide Program Stage</td>
<td>Hides a whole program stage from the user if the rule is true.</td>
<td>![✓]</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Send Message</td>
<td>Send Message triggers a notification based on provided message template. This action will be taken whenever there is a change in data value. However this behaviour can be controlled by providing event-enrollment status in program rule expression.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Schedule Message</td>
<td>Schedule Message will schedule notification at date provided by Expression in the data field.</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>
## 10.4 Functions to use in program rule expressions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description of function</th>
<th>Status</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>d2:ceil</td>
<td>Rounds the input argument up to the nearest whole number.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:floor</td>
<td>Rounds the input argument down to the nearest whole number.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:round</td>
<td>Rounds the input argument to the nearest whole number.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:modulus</td>
<td>Produces the modulus when dividing the first with the second argument.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:zing</td>
<td>Evaluates the argument of type number to zero if the value is negative, otherwise to the value itself.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:oizp</td>
<td>Evaluates the argument of type number to one if the value is zero or positive, otherwise to zero.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:concatenate</td>
<td>Produces a string concatenated string from the input parameters. Supports any number of parameters.</td>
<td>✔️</td>
<td>Use <code>d2:concatenate</code> function instead of using “+” as the expression evaluator in the app will be adding numbers if it can.</td>
</tr>
<tr>
<td>d2:daysBetween</td>
<td>Produces the number of days between the first and second argument. If the second argument date is before the first argument, the return value will be the negative number of days between the two dates. The static date format is 'yyyy-MM-dd'.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>d2:weeksBetween</td>
<td>Produces the number of full weeks between the first and second argument. If the second argument date is before the first argument, the return value will be the negative number of weeks between the two dates. The static date format is 'yyyy-MM-dd'.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Function</td>
<td>Description of function</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>d2:monthsBetween</td>
<td>Produces the number of full months between the first and second argument. If the second argument date is before the first argument the return value will be the negative number of months between the two dates. The static date format is 'yyyy-MM-dd'.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:yearsBetween</td>
<td>Produces the number of years between the first and second argument. If the second argument date is before the first argument, the return value will be the negative number of years between the two dates. The static date format is 'yyyy-MM-dd'.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:addDays</td>
<td>Produces a date based on the first argument date, adding the second argument number of days.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:count</td>
<td>Counts the number of values that is entered for the source field in the argument.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:countIfValue</td>
<td>Counts the number of matching values that is entered for the source field in the first argument. Only occurrences that matches the second argument is counted.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:countIfZeroPos</td>
<td>Counts the number of values that is zero or positive entered for the source field in the argument. The source field parameter is the name of one of the defined source fields in the program.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:hasValue</td>
<td>Returns the number of numeric zero and positive values among the given object arguments. Can be provided with any number of arguments.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:validatePattern</td>
<td>Evaluates to true if the input text is an exact match with the supplied regular expression pattern. The regular expression needs to be escaped.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:left</td>
<td>Evaluates to the left part of a text, num-chars from the first character.</td>
<td>✅</td>
<td></td>
</tr>
<tr>
<td>d2:right</td>
<td>Evaluates to the right part of a text, num-chars from the last character.</td>
<td>✅</td>
<td></td>
</tr>
</tbody>
</table>
### Function Description of function | Status | Notes on implementation
---|---|---
**d2:substring** | Evaluates to the part of a string specified by the start and end character number. | ✓ |  |
**d2:split** | Split the text by delimiter, and keep the nth element (0 is the first). | ✓ |  |
**d2:length** | Find the length of a string. | ✓ |  |
**d2:zpvc** | Returns the number of numeric zero and positive values among the given object arguments. Can be provided any number of arguments. | ✓ |  |
**d2:inOrgUnitGroup** | Evaluates whether the current organization unit is in the argument group. The argument can be defined with either ID or organization unit group code. |  |  |
**d2:hasUserRole** | Returns true if the current user has this role otherwise false. |  |  |
**d2:zScoreWFA** | Function calculates z-score based on data provided by WHO weight-for-age indicator. Its value varies between -3.5 to 3.5 depending upon the value of weight. |  |  |

* Available in DHIS 2 v2.30 ** Available in DHIS 2 v2.31 onwards *** Available in DHIS 2 v2.32

### 10.5 Standard variables to use in program rule expressions

Available in DHIS2 v2.30

| Variable | Description of function | Status | Notes on implementation |
---|---|---|---
**V{current_date}** | Contains the current date whenever the rule is executed. | ✓ |  |
**V{event_date}** | Contains the event date of the current event execution. Will not have a value at the moment the rule is executed as part of the registration form. | ✓ |  |
**V{event_status}** | Contains status of the current event or enrollment. | ✓ |  |
**V{due_date}** | This variable will contain the current date when the rule is executed. Note: This means that the rule might produce different results at different times, even if nothing else has changed. | ✓ |  |
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description of function</th>
<th>Status</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>V{event_count}</td>
<td>Contains the total number of events in the enrollment.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{enrollment_date} *</td>
<td>Contains the enrollment date of the current enrollment. Will not have a value for single event programs.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{incident_date} *</td>
<td>Contains the incident date of the current enrollment. Will not have a value for single event programs.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{enrollment_id} *</td>
<td>Universal identifier string (UID) of the current enrollment. Will not have a value for single event programs.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{event_id}</td>
<td>Universal identifier string (UID) of the current event context. Will not have a value at the moment the rule is executed as part of the registration form.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{orgunit_code}</td>
<td>Contains the code of the orgunit that is linked to the current enrollment. For single event programs the code from the current event Org Unit will be used instead.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{environment}</td>
<td>Contains a code representing the current runtime environment for the rules. The possible values is “WebClient”, “AndroidClient” and “Server”. Can be used when a program rule is only supposed to run in one or more of the client types.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>V{program_stage_id}</td>
<td>Contains the ID of the current program stage that triggered the rules. This can be used to run rules in specific program stages, or avoid execution in certain stages. When executing the rules in the context of a TEI registration form the variable will be empty.</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Description of function</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>V(program_stage_name)</td>
<td>Contains the name of the current program stage that triggered the rules. This can be used to run rules in specific program stages, or avoid execution in certain stages. When executing the rules in the context of a TEI registration form the variable will be empty.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Only applies to tracker*
11 Program Indicators

The following is a comprehensive list of all Program indicator variables available in DHIS2, and notes on whether or not these have been implemented in the Android Capture app.

Any issues around using a particular feature with Android are highlighted with an exclamation mark.

<table>
<thead>
<tr>
<th>legend</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️</td>
<td>Component implemented</td>
</tr>
<tr>
<td></td>
<td>Component not implemented (rule fails)</td>
</tr>
<tr>
<td>N/A</td>
<td>Not applicable</td>
</tr>
<tr>
<td>⚠️</td>
<td>Work in progress. Feature not completely implemented yet or with unexpected behaviour already reported.</td>
</tr>
</tbody>
</table>

11.1 Variables to use in a program indicator expression or filter

<table>
<thead>
<tr>
<th>Variable type</th>
<th>Description of variable type</th>
<th>Status</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event Date</td>
<td>The date of when the event took place.</td>
<td>⚠️</td>
<td></td>
</tr>
<tr>
<td>event_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Creation Date*</td>
<td>The date of when an event or enrollment was created in the system.</td>
<td>⚠️</td>
<td></td>
</tr>
<tr>
<td>creation_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Due Date</td>
<td>The date of when an event is due.</td>
<td>⚠️</td>
<td></td>
</tr>
<tr>
<td>due_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sync Date*</td>
<td>The date of when the event or enrollment was last synchronized with the Android app.</td>
<td>⚠️</td>
<td></td>
</tr>
<tr>
<td>sync_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Incident Date</td>
<td>The date of the incidence of the event.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>incident_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enrollment Date (not visible on UI)</td>
<td>The date of when the tracked entity instance was enrolled in the program.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>enrollment_date</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enrollment Status</td>
<td>Can be used to include or exclude enrollments in certain statuses.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enrollment_status</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable type</td>
<td>Description of variable type</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>--------</td>
<td>----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Current Date current_date</td>
<td>The current date.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Completed Date</td>
<td>The date the event is completed.</td>
<td>🟢</td>
<td></td>
</tr>
<tr>
<td>Value Count value_count</td>
<td>The number of non-null values in the expression part of the event.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Zero or positive value count</td>
<td>The number of numeric positive values in the expression part of the event.</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Event Count event_count</td>
<td>The count of events (useful in combination with filters).</td>
<td>✔️</td>
<td></td>
</tr>
<tr>
<td>Enrollment Count enrollment_count</td>
<td>The count of enrollments (useful in combination with filters).</td>
<td>N/A</td>
<td>Indicators in the Android App are calculated in the domain of one TEI enrollment. Value always 1.</td>
</tr>
<tr>
<td>TEI Count tei_count</td>
<td>The count of tracked entity instances (useful in combination with filters).</td>
<td>N/A</td>
<td>Indicators in the Android App are calculated in the domain of one TEI enrollment. Value always 1.</td>
</tr>
<tr>
<td>Program Stage Name program_stage_name</td>
<td>Can be used in filters for including only certain program stages in a filter for tracker programs.</td>
<td>🟢</td>
<td></td>
</tr>
<tr>
<td>Program Stage ID program_stage_id</td>
<td>Can be used in filters for including only certain program stages in a filter for tracker programs.</td>
<td>🟢</td>
<td></td>
</tr>
<tr>
<td>Reporting Period Start reporting_period_start</td>
<td>Can be used in filters or expressions for comparing any date to the first date in each reporting period.</td>
<td>N/A</td>
<td>Indicators in the Android App are calculated in the domain of one TEI enrollment.</td>
</tr>
<tr>
<td>Variable type</td>
<td>Description of variable type</td>
<td>Status</td>
<td>Notes on implementation</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Reporting Period End</td>
<td>Can be used in filters or expressions for comparing any date to the last inclusive date in each reporting period.</td>
<td>N/A</td>
<td>Indicators in the Android App are calculated in the domain of one TEI enrollment.</td>
</tr>
</tbody>
</table>

**Documentation Reference** DHIS 2 v2.32
12 Value types supported

The following is a comprehensive list of all value types available in DHIS 2, and notes on whether or not these have been implemented in the Android Capture app.

Any issues around using a particular feature with Android are highlighted with an exclamation mark !.

<table>
<thead>
<tr>
<th>Value type</th>
<th>Description of value type</th>
<th>Program with registration</th>
<th>Program without registration</th>
<th>Data Set</th>
<th>Notes on implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>Time only</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Date &amp; Time</td>
<td>Date plus time</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td>Date only</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Age</td>
<td>Enables entry of either an age in years/months/days or a date-of-birth (both are stored as date-of-birth)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Phone number</td>
<td>A valid phone number</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Email</td>
<td>An email address in a valid format</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Yes/no</td>
<td>Boolean yes/no (or no response)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Yes only</td>
<td>Yes or no response</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Value type</td>
<td>Description of value type</td>
<td>Program with registration</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Data Set</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>-------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Number</td>
<td>Any valid number, including decimals</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Integer</td>
<td>Any integer (whole numbers, no decimals)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Positive Integer</td>
<td>Only positive integers (no zero or negative values)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Positive or Zero Integer</td>
<td>Only zero or positive integers (no negative values)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Negative Integer</td>
<td>Only negative integers (no zero or positive values)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Percentage</td>
<td>Any decimal value between 0 and 100</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Unit interval</td>
<td>Any decimal value between 0 and 1</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Text</td>
<td>Text (length of text up to 50,000 characters)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Long text</td>
<td>Text (no constraints on length)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Letter</td>
<td>A single letter</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Value type</td>
<td>Description of value type</td>
<td>Program with registration</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Data Set</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>File</td>
<td>Enables upload of files in various formats (requires appropriate storage to be configured)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Organisation unit</td>
<td>Enables selection of a DHIS2 organisation unit as the chosen value</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Tracker Associate</td>
<td>Enables selection of an existing Tracker ‘tracked entity instance’ (e.g. a person) as the value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Username</td>
<td>Enables selection of a valid DHIS2 username as the value</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordinate</td>
<td>Enables manual entry of geographical coordinates (doesn’t enable automatic capture of coordinates)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>URL</td>
<td>Enables manual entry of a URL.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Value type</td>
<td>Description of value type</td>
<td>Program with registration</td>
<td>Program with registration</td>
<td>Program without registration</td>
<td>Data Set</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>---------------------------</td>
<td>-----------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Image</td>
<td>Enables upload of images.</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

12 Value types supported

11.1 Variables to use in a program indicator expression or filter