Open Foris Collect

Handbook

21 October 2019

http://www.openforis.org
CONTENTS

1 Introduction ........................................................................................................... 3
2 Key features ........................................................................................................... 3
3 Installation............................................................................................................. 4
   3.1 Prerequisites ................................................................................................. 4
3.2 Standalone Installation (single user environment) ........................................... 4
3.3 Download and run the installer ....................................................................... 4
   3.3.1 Windows systems ..................................................................................... 4
   3.3.2 Linux systems .......................................................................................... 4
   3.3.3 Mac OSx systems ..................................................................................... 4
4 Start-up Collect ..................................................................................................... 5
   4.1 Windows systems ......................................................................................... 5
   4.2 Linux systems ............................................................................................... 5
   4.3 Mac OSx systems .......................................................................................... 5
5 Using Collect Control Panel .............................................................................. 5
6 Shutdown Collect .................................................................................................. 6
7 Updating Collect .................................................................................................... 7
   7.1 Update using Auto Updater .......................................................................... 7
   7.2 Update from version of Collect older than 3.2.2 .......................................... 7
   7.3 Upgrade Collect offline (for expert users only) ............................................. 7
8 Collect Home ......................................................................................................... 9
9 Dashboard .............................................................................................................. 11
10 Data Management ............................................................................................... 13
   10.1 Adding a record .......................................................................................... 14
      10.1.1 Data upload (from Collect Mobile) ....................................................... 14
      10.1.2 Manual data entry .............................................................................. 16
   10.2 Validation report ......................................................................................... 17
   10.3 Adding more records .................................................................................. 18
   10.4 Record completeness .................................................................................. 20
   10.5 Data export .................................................................................................. 20
   10.6 Data Workflow ............................................................................................. 22
      10.6.1 Data entry ............................................................................................ 22
      10.6.2 Data cleansing ...................................................................................... 22
      10.6.3 Data analysis ....................................................................................... 23
11 Survey Designer .................................................................................................. 24
   11.1 New Survey .................................................................................................. 25
   11.2 Code lists ..................................................................................................... 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.3</td>
<td>Species list</td>
</tr>
<tr>
<td>11.4</td>
<td>Sampling point data</td>
</tr>
<tr>
<td>11.5</td>
<td>Schema</td>
</tr>
<tr>
<td>11.5.1</td>
<td>Description of parameters</td>
</tr>
<tr>
<td>11.5.2</td>
<td>Examples of “relevant when” rules</td>
</tr>
<tr>
<td>11.5.3</td>
<td>Examples of default attribute values</td>
</tr>
<tr>
<td>11.5.4</td>
<td>Examples of data validation rules</td>
</tr>
<tr>
<td>11.5.5</td>
<td>Add a node from another survey</td>
</tr>
<tr>
<td>11.6</td>
<td>Advanced functions</td>
</tr>
<tr>
<td>12</td>
<td>Data cleansing</td>
</tr>
<tr>
<td>12.1</td>
<td>Basic concepts</td>
</tr>
<tr>
<td>12.2</td>
<td>Data cleansing workflow</td>
</tr>
<tr>
<td>12.2.1</td>
<td>Data backup</td>
</tr>
<tr>
<td>12.2.2</td>
<td>Define Data Query Types</td>
</tr>
<tr>
<td>12.2.3</td>
<td>Define Data Queries</td>
</tr>
<tr>
<td>12.2.4</td>
<td>Group Queries into Data Query Groups</td>
</tr>
<tr>
<td>12.2.5</td>
<td>Generate a Data Report</td>
</tr>
<tr>
<td>12.2.6</td>
<td>Defining Cleansing Steps for Queries</td>
</tr>
<tr>
<td>12.2.7</td>
<td>Defining a Data Cleansing Chain</td>
</tr>
<tr>
<td>13</td>
<td>Backup</td>
</tr>
<tr>
<td>14</td>
<td>Saiku</td>
</tr>
<tr>
<td>15</td>
<td>Security (User accounts)</td>
</tr>
<tr>
<td>15.1</td>
<td>Users</td>
</tr>
<tr>
<td>15.2</td>
<td>User role permissions table</td>
</tr>
<tr>
<td>15.3</td>
<td>Groups</td>
</tr>
<tr>
<td>15.4</td>
<td>Role in group permissions table</td>
</tr>
<tr>
<td>Annex 1</td>
<td>Examples of code in Open Foris Collect</td>
</tr>
<tr>
<td>Annex 2</td>
<td>PostgreSQL with Collect</td>
</tr>
</tbody>
</table>
1 INTRODUCTION

Open Foris Collect is the main entry point for data collected in field-based inventories. It provides a fast, easy, flexible way to set up a survey with a user-friendly interface. Collect handles multiple data types and complex validation rules, all in a multi-language environment.

Open Foris Collect provides a flexible solution for field data management, allowing full customization of inventory structure, variables and data checks. Collect promotes data quality through an integrated data entry and data cleansing workflow. Collect introduces the concept of the Inventory Data Metamodel (IDML), a formal description (i.e., metadata) of the types of variables, classifications and coding schemes used by the inventory. All inventories documented in this way may be entered and retrieved through a user-friendly interface, without additional programming. Collect is available in either standalone (offline) or web-based (online) versions.

2 KEY FEATURES

The software key features are as follows:

- **User Friendliness**: Nice web interface; Designed based on real users’ needs, No need for technical skills to use it.

- **Rapid Data Entry**: Limited use of mouse needed; Data entry using only keyboard; Auto-complete; Species list search; Immediate feedback on errors/warnings.

- **Highly Configurable**: Design the survey from scratch or starting from a template; Data entry user interface is automatically generated and metadata driven; Validation rules (distance, comparison, pattern...); Multiple layouts (form, table, multiple columns form).

- **Multiple data types**: Basic Types – Text, Number, Boolean, Date, Time. Complex types – Range, Coordinate, File, Taxon. Plus, support for calculated values.

- **Multi-user or standalone**: It can be used in a standalone environment with no need for internet connection; Data can be exported from single/standalone installations and imported into a centralized installation to create a complete data set; In multi-user environment, users can work only on owned records.

- **Controlled QA workflow**: Record goes through different steps: Data entry, Data cleansing, Data analysis. Minimized "data cooking".


- **Multilingual**: Define the survey in multiple languages - Tab labels, Input field labels, Validation messages, Code item labels, Element info tooltips. The user will see the survey in the language of his/her web browser or in the survey default language.

- **Multiple data export/import formats**: XML, CSV, Relational database.
3 INSTALLATION

3.1 PREREQUISITES

1. Web browser: **Google Chrome** is recommended.
2. Verify that the browser has the Adobe Flash plugin installed: access Adobe Flash Player test (http://www.adobe.com/software/flash/about/) and check that Adobe Flash Player is properly installed, otherwise install it following the instructions here: http://get.adobe.com/flashplayer/

3.2 STANDALONE INSTALLATION (SINGLE USER ENVIRONMENT)

If running Collect offline in a single-user environment, follow these instructions.

3.3 DOWNLOAD AND RUN THE INSTALLER

Download the installer from http://www.openforis.org/tools/collect.html (select the proper installer for your operating system)

3.3.1 WINDOWS SYSTEMS

- Run the .exe file and follow the instructions on screen.
- Collect will be installed by default into C:\OpenForis\Collect.
- The user data will always be stored into your user folder (E.g. C:\Users\YOUR_USERNAME\OpenForis\Collect)
- If the installation completes successfully, you will have a group of shortcuts in the start menu, **Open Foris Collect**
- If a message from Windows Firewall appears saying that Java is trying to access the network, **allow it** (if you have administrative rights, otherwise close the message popup). This operation needs to be done only once.

3.3.2 LINUX SYSTEMS

- Make the downloaded .run file executable.
- Run the .run file and follow the instructions on screen.
- Collect will be installed by default into ~/OpenForis/Collect
- If the installation completes successfully, you will have 2 icons in the Desktop, **Launch Open Foris Collect** and **Update Open Foris Collect**

3.3.3 MAC OSx SYSTEMS

- Open the .dmg file and follow the instructions on the screen.
- Collect will be installed by default into ~/OpenForis/Collect
- If the installation completes successfully, you will have 2 shortcuts in the Desktop, **Launch Open Foris Collect** and **Update Open Foris Collect**.
Note: By default, Collect is using a SQLite database in your local machine. If you use PostgreSQL database, see required changes to be done after the first installation or updating in Annex 2.

Further information and technical details for the installation of Open Foris Collect and the difference between the installations for a single (local) user versus installing a server for a multiuser environment can be found here:


4 START-UP COLLECT

4.1 WINDOWS SYSTEMS

- In the start menu, select "All programs", then go into Open Foris folder and select Launch Open Foris Collect
- The Collect Control Panel will appear. See Using Collect Control Panel

4.2 LINUX SYSTEMS

- Double click on the Launch Open Foris Collect icon in the desktop
- The Collect Control Panel will appear. See Using Collect Control Panel

4.3 MAC OSx SYSTEMS

- Double click on the Launch Open Foris Collect icon in the desktop
- The Collect Control Panel will appear. See Using Collect Control Panel

5 USING COLLECT CONTROL PANEL

- After clicking on Launch Open Foris Collect icon, the control panel will appear:

Please wait until the “Starting up...” message disappears and the “Running!” message appears:

![Starting up...]

Collect running in default web browser at: http://localhost:8380/collect

![Running!]

Log
Collect will open automatically a new window in the default web browser, linking the address: [http://localhost:8380/collect](http://localhost:8380/collect). This address can be different in your system: the port 8380 can be already in use in your computer, so another free port will be chosen by the installer.

If the Control Panel says that Collect is running but the window is not automatically opened, please open the web browser and access that address manually.

Collect login page should appear on the web browser; enter the system the first time using "admin" as username and "admin" as password. You can change the password later from the user interface. [If nothing happens, open a new browser window and access this url: http://localhost:8380/collect]

If an error occurs, please click on the Log button; the log details box will be opened and it will be possible to have more information about the error, or at least copy and paste the content of this text area and send it to the Open Foris Team to have support.

- Log info is divided into 3 sections:
  - Server: errors related to the application server (Jetty) that runs Collect. Usually here you can find error information when Collect does not start up.
  - Collect: here you can find detailed information about errors related to the application (Collect) itself, for example when you have an error during the normal execution of the application and a popup shows up.
  - Saiku: errors related to Saiku (the reporting system embedded into Collect and accessible from the Home Page).

6 **SHUTDOWN COLLECT**

- Close all the opened browser windows that are using Collect
7 UPDATING COLLECT

Open Foris Collect developers are constantly working on improving the system also taking into consideration, and trying to address, the feedback from the users in the countries.

When you work with Collect in the web browser, in the bottom left corner, there is a small icon next to application version: this can be used to check whether the latest version is in use.

Following are the steps for updating Collect to the newest released version.

7.1 UPDATE USING AUTO UPDATER

Starting from version 3.2.2, Collect has an Auto Updater that you can run from the start menu

1. close every browser window that is using Collect
2. shutdown Collect
3. click on Update Open Foris Collect startup menu item and follow the instructions on screen
4. start Collect

7.2 UPDATE FROM VERSION OF COLLECT OLDER THAN 3.2.2

If you want to upgrade Collect from an older version than 3.2.2 to the new one, follow these steps:

1. close every browser window that is using Collect
2. shutdown Collect
3. make a copy of the folder data (path: c:\opt\openforis\collect\tomcat\data) and store it outside of c:\opt\openforis, e.g. in your "user" folder, under a path like OpenForisCollectBackup
4. uninstall Collect
5. install Collect using the new installer
6. copy the folder data previously backed up into the folder c:\Users\USERNAME\OpenForis\Collect\
7. start Collect and verify that all old data are present

7.3 UPGRADE COLLECT OFFLINE (FOR EXPERT USERS ONLY)

1. The easiest way is to update Collect offline is to download the latest updater:
2. Stop the Collect Control Panel in offline computer, if running. Wait for 2 minutes.

3. Create a backup copy of your existing Collect installation (c:\OpenForis\Collect folder)

4. Create a backup of the PostgreSQL Collect database (if you are using that as database)

5. Then copy the downloaded file into c:\OpenForis\Collect\autoupdate and run it (it must be inside this folder, otherwise it won't find correctly the Collect installation). This process doesn't require administrative rights.

6. Start the Collect and verify that everything works fine

**Warning:** if you installed Collect before April 2017 (i.e. you are using older version than 3.12.30) and you are using SQLite database in your local machine, copy your data before uninstalling Collect, otherwise you will lose all your Collect data! If you are using PostgreSQL database for storing Collect data, uninstaller can be used. For more information on installing and running a PostgreSQL server for Collect, see Appendix 2.
8 COLLECT HOME

Collect welcome page offers the following options: Dashboard, Data management, Survey designer, Data cleansing, Map, Saiku, Backup/Restore, and Security.

- **Dashboard**: for each published survey, it allows to get an overview on the data collection process, by showing statistics on the records already created, modified, entered, and/or cleansed.

- **Data management**: once a survey has been created, select Data management to start entering data and manage data workflow. Stored surveys can be accessed and records can be managed and edited.

- **Survey designer**: the starting point for setting up and manage your survey. Select Survey designer to Create, Import, Export, Edit, Publish/Unpublish, Validate, Clone or Delete surveys.

- **Data cleansing**: Collect Data Cleansing Toolkit can help you to find errors in the data and to fix them.

- **Map**: it shows the data collected on a map, by using GPS coordinates.

- **Saiku**: it is web-based open source software that facilitates data visualization and data querying.

- **Backup/Restore**: used for data backup and restoring.

- **Security**: it allows creating personal user profiles (password protected) by specifying the role in the workflow: View, Entry limited, Entry, Cleansing, Analysis,
Administrator. It also allows creating user groups, who will work each on different surveys or on different parts of one same survey.

And in the **bottom left corner**: Collect current version and user login/logout
9 DASHBOARD

In the dashboard, you will be asked to select a published survey, which you can pick from the drop-down list at the top right of the page.

Once one survey has been selected, you will see simple statistics on the data collection workflow related to that survey (data created, modified, entered, and/or cleansed – if any), sorted by day, month, and year.
10 DATA MANAGEMENT

This section of Collect is the starting point for Data Entry and Data Management. In this section records can be entered, edited and managed. Once records are entered into Collect, this section presents a log of errors and warnings encountered during data entry as well as information on when records were created or modified and their status in the data cleansing process. In addition, it is also possible to visualize which user entered a specific record.

Clicking on Data Management opens the list of records present (if any).

A typical Data Management view is shown in the example image below. Note that different user types will have different views and editing permissions as outlined in the “Users accounts” section.

Records can be sorted by columns. In addition, it is possible to show only the data collected by a specific user, by clicking on the dropdown field within the last column – you can choose to show all data or those belonging to yourself or to another user. The buttons at the top allow running a Validation report, to export or import data, and to manage the data Workflow, by promoting (steps forward) or demoting (steps backward) the records.

At the central (or sub-central) office, a data manager should be identified as the main responsible person for handling the data coming from the field. This means setting up an archiving and backup system, keeping a constant contact with field crews, uploading the data into OF Collect Desktop, leading the data cleansing process and, if needed, requesting
field crews for explanation on unusual values and/or requesting the re-measurement of dubious/wrong attributes.

Data Archiving

From a collector’s point of view, there are two options for sending records from the field to the data manager. First, once a record is completed, it can be send directly through a specific function of Collect Mobile, so that it migrates automatically toward the central server. Second, the completed record can be downloaded with Collect Mobile and sent it by email or saved on a local drive or a cloud system (for more details, please see the Collect Mobile Manual). In this second case, soon after data collection in the field has started, the data manager will receive data files from the field crew. In order to avoid confusion, a structured archiving system should be set up. Regardless whether the data files will be stored on a local drive or through a cloud system, a structure with multiple sub-folders is preferred. This should be agreed in advance and one possibility is to assign a folder to each crew (if a cloud system is used, each crew could be given only the link to its dedicated sub-folder).

10.1 Adding a record

10.1.1 Data upload (from Collect Mobile)

If data is recorded in the field with OF Collect Mobile, data files are either sent directly through Collect Mobile, or exported and sent through another device (see above). In this second case, the data can be uploaded. At the top of the Data Management screen, click on the Import yellow button and choose a format (CSV or Collect format).

The user will be prompted to click on an empty box to select a file to import, or to drop it inside. The data collected will be a .collect-data file.
Click on the button **Generate a complete summary**, that will let you know if the data you are importing is compatible with that specific survey.

You will see the list of completed records with few metadata (Id, date of creation, step, etc.). From this list, you can select one, many, or all the records.

Once you selected the records you want to import, click on the button **Import**. 

About column **Importability**: if the records you are importing are newer than the existing records, importability will be green, otherwise red. If the records have exactly the same date
and time labels, the equal symbol (=) is shown, and it’s not necessary to import those records.

The screen will then display the current status, listing the records that have been added successfully to the database, as shown below. In case of Errors or Warnings, they will be indicated and can be dealt with during data cleansing.

10.1.2 Manual data entry

To add data manually, click on the New blue button at the top left of the screen. The user can start to enter data moving from field to field using 'Tab'. The behavior of each record field depends on its parameters (see examples in the chapter ‘Schema’).

As shown in the image below, Errors will be highlighted in red, warnings in yellow. If no value was present in the paper form, the data entry user can leave the field blank and specify the reason (by right-click on the field and choosing one of the options).
Once all the fields have been filled-in for a record, the user can **Submit** it (top-right corner). The record will be added to the log and available for the next steps in the data work flow.

10.2 **VALIDATION REPORT**

At this point it is possible to run a Validation Report to investigate the nature of the errors shown in the list of records. Click on the **Validation Report** green button. The result will be a `.csv` file with details on the error(s) present in the records.
The process for correcting mistakes should be agreed with the Data manager who may have to contact the field crew leader responsible for the specific record (SU) and ask for clarifications. In some cases, the field crew may have to go back to the field to re-collect the wrong information. The correct information can then be sent with the successive data export/submission from the field or the data manager may manually correct the value from the data entry interface.

### 10.3 Adding more records

As field data collection continues, the data manager will receive additional data files. The new data files will contain either new records or newer versions of existing records. In the latter case, for example, if data related to sample unit #1 was already sent but field work was not completed, the new export file will contain a newer version of the same record which will substitute the older version.

Repeating the same steps as above for importing a new data file, leads to the following summary.
In the top table we see that for records #01, 02 and 03 a new and more complete version was sent with the second data export. As before, modification date and error numbers is presented. The column Completion difference (green increment bar) indicates the number of filled attributes of the new record as compared to the existing one. The importability column (green dot) indicates whether the latest record is more complete (in the case of a new record with less information, the dot would be red, in the case of unaltered information, the symbol “=” would be displayed). Placing the mouse over the two columns shows further details as shown below.

The last column on the right: “Replace?” is used to select those records that should be uploaded to replace older (and less complete) versions of the same records. Click to select record to be uploaded choosing from: select all, only newer (green dot), or select records individually. The bottom table, as in the first data upload simply lists records not currently present in the database, thus ready for upload by default.
After a selection has been made, ticking appropriately, click Start to Import. When the message “Data import completed” appears, close the summary window (‘X’ on the top right corner). The resulting log of records in the database is shown below. All records in the database are listed, including errors, warnings and dates of creation and modification.

10.4 RECORD COMPLETENESS

The indication of Completion difference and Importability, as described above, are useful tools to assess how much data has been collected for a specific record. However, in the context of a National Forest Inventory, this may not be needed. It is unlikely that field crews stop working in the middle of a cluster or a plot and export data or, at least, they should be instructed not to do so, rather, they should wait until the work on a cluster/plot is completed before exporting data.

An additional feature that could help in having an immediate view of, for example, how many plots have been entered for a specific cluster, is to click the box for “Show count in record summary list” for the plot entity in Survey Design. See below.

By doing so, in the log of records in Data Management, a new column showing count for the selected entity will be added. See below.

10.5 DATA EXPORT
Another way to have a closer look at the data is to export it. In the Data Management window, click on the (blue) *Export data* button. Selecting XML (.collect-data) allows sharing the data set so that another (with the same collect survey) can import data.

This function can also be used to perform a backup, with or without uploaded files (such as images – see below).

Otherwise, selecting CSV and clicking on the “All entities” export mode will create CSV file(s) for each of the entity in the survey.

Select a data Step. Click Export. When the message “Exporting data complete” appears, click *Download CSV file*. The result is a Zip file containing a CSV file for each entity (e.g. cluster.csv, plot.csv, tree.csv).
Collect implements a workflow which divides data processing into three phases: data entry, data cleansing, and data analysis.

**10.6 DATA WORKFLOW**

10.6.1 DATA ENTRY

- Data are entered into the system exactly as they appear on paper field forms.
- They are neither interpreted nor corrected during this phase.
- Missing or incorrect values are preferred “over-cooked data” ([http://en.wikipedia.org/wiki/Cooking_(science)](http://en.wikipedia.org/wiki/Cooking_(science))), as incorrect guesses are an unwanted source of hidden errors. As no interpretation is done at this phase, data entry can be done by staff with no forestry-specific knowledge.
- During this phase, default values are not shown in order to further discourage data cooking.
- For fields of *relevant* attributes with no values, the user must always specify the *reason* the field was left blank (blank on form, n/a or dash on form, illegible value).
- Invalid values (those marked as “error”) must be checked against the field forms. If the value was copied correctly, the value should be manually accepted. If it was copied incorrectly, the value should be entered as it appears on the field forms.
- On all fields, remarks may be entered to provide information which may assist data cleansing and analysis.
- Once all errors are either resolved or confirmed, the data entry clerk submits the *record* for data cleansing. Before entering the cleansing phase, a copy is made and marked read-only. No further changes may be made to this record by data entry staff (unless the record is returned by data cleansing staff).

10.6.2 DATA CLEANSING

- Experts with knowledge of field conditions and data work with a copy of the data entered in phase 1 to correct or remove missing or invalid values. Estimated and imputed values may be entered in this phase and unavailable values removed.
- If the data cleansing expert deems that the data entry task was not completed or that the record contains too many errors, the *record* may be returned to phase 1 (entry) for additional changes.
- When available, default values automatically appear for optional, relevant fields left blank in the data entry phase. For all other fields, available defaults may be applied manually. Defaults may always be overridden or removed by the user.
- Additional remarks may be added to each field to assist in interpretation of the data during analysis.
- Once all data errors are resolved, the record may then be submitted for data analysis. This locks the *record* so that no further changes may be made.
10.6.3 Data analysis

- Records marked for data analysis may be exported to a relational database for further elaboration and analysis. Tools such as *Open Foris Calc*, R, Excel and others may then be used to derive results.

- If during analysis data are deemed incomplete or incorrect, the record may be rejected and sent back for further cleansing. Once updated and resubmitted, the record may once again be exported for calculation and analysis.
**11 Survey Designer**

Under the Survey Designer management menu item, there are three other items: 1) List of surveys; 2) New survey; and 3) Import survey. By clicking on the first one, you will see the surveys currently uploaded (if any) and their status: whether they have been published or not, the date of the last change made, and which is the user group of each survey.

By clicking on one survey, three buttons will appear at the top of the screen: *Edit*, *Export*, and *Advanced Functions*. Details on these operations will be covered after the instructions on how to set up a new survey.

To start setting up your first Survey, click on the *New* blue button or on the second menu item *New survey*. You will be requested to enter a *name* for the survey, choose from a *template*, select the *default survey language*, and choose one *user group*. Selecting a *Blank* template means starting from scratch, with no preset information loaded. Alternatively, it is
possible to choose from templates in which some fields in the Code Lists are prefilled in accordance to pre-set survey specifications.

**Note**: additional languages can also be added at a later stage by clicking on the UN flag (top right corner) and select and add additional languages for the survey. The user can then switch from one language to another by selecting a language in the drop-down menu on the top right corner of the screen. Note that the fields that allow multiple languages will show a codified language abbreviation in parenthesis, for example (en) for English.

By clicking on the third menu item **Import survey**, you can also use an existing survey by importing it in the following formats: Collect Survey (.collect) Collect Backup (.collect-backup), Collect Earth (.cep) or Collect Survey XML (.xml).

11.1 **NEW SURVEY**

In the first tab (Survey), the user is asked to provide general and basic information about the Survey.

The fields to be filled in are as follows:

- **Name**: defined when the survey was created
- **Project name**: can be filled in to specify an additional name for the project. Must be given in English.

- **Description**: can be filled to include any additional note to describe the Survey. Must be given in English.

**NOTE**: Mandatory fields are bordered in red, while all others, when clicked, are bordered in blue.

**Additional fields**: *Form versions, Spatial Reference System, Units and Files.*

Popup windows open by clicking the icon 📋 to the right of field name. Inside of each pop up window, new items can be: created by clicking on the Green "plus" button; moved up or down by clicking the up and down arrow; deleted by clicking the Red "minus" button.

Once finished, close the window by clicking **Apply**.

**Form versions**: This field can be used to keep track of successive versions of the Survey forms (for entering data). The user can add a new form version by clicking the Green "plus" button and specifying: Name (e.g. 1.0, 1.1 etc.); Label (a codified label used to express useful information (e.g. date, form version, location [e.g. BP 1.3.2010])); Description (any additional information); Date (usually the date of when that form version was first used).

**NOTE**: Try to avoid using different form version in you survey. *This feature can cause troubles in data analysis when using Open Foris Calc phase if different versions have different variables and/or data types.*

**Spatial Reference Systems (SRS)**: This field is used to specify the geospatial settings used in the survey by selecting a predefined SRS from the list (bottom left of the window). The user can enter information related to the Spatial Reference System(s) used in the survey by specifying: Id, Label, Description and Well Known Text.

**Units**: This field is used to define the units of measure that will be used throughout the Survey. A set of predefined units are automatically loaded. They include units to measure for Angles, Areas, Currency, Length, Mass, Ratio and Time. Should the user wish to include additional units he/she may do so by clicking the green "+" button and filling in the relevant fields: Name, Label, Abbreviation, selecting a Dimension, and Conversion factor. Note that the conversion factor relates units within their own dimension (length, angle, etc.). One centimetre has a conversion factor of 0.01 to a meter, if the conversion factor of meter is set to 1.

**Files**: You can upload files to go with your survey. This feature is there mostly for Collect Earth, but it can be useful for carrying inventory guidelines or other documents into the field.
Advanced functions

**Validate:** Click to check if your survey has any errors or warnings. These might be minor things like unused code lists or things that prevent the survey from working, like lacking key attributes. Any errors will prevent the publishing of the survey.

**Export:** Produces a file of the survey that can be imported and used in other devices. You have a choice between Collect (.collect), Collect mobile (.collect-mobile), Collect Earth (.cep) and SQL Relational database (.sql) formats.

**Export schema summary:** Creates a CSV file that summarizes the data structure of the survey. This is useful for defining conditions for attributes’ relevance, for example.

**Import attributes from Excel/CSV:** Useful for creating large surveys from scratch: you can create a survey data structure in CSV and import it. This saves time and effort if you know what you are doing, i.e. you know how the data structure works. You can use an exported survey’s CSV as a basis for your own, for example.

**Export CSV data import template:** Prints a CSV file in which columns are named and ordered so that they are compatible with the survey. This makes it easier to import large amounts paper form data, as it can be typed into the CSV without the need to click on entities and attributes. Care should be taken however, that data is input correctly.

### 11.2 Code lists

This tab is used to define lists of codified information that will be used throughout the survey to provide options for specific fields during the data entry phase. If the user selected one of the pre-filled templates, several Code lists are loaded automatically. They include codified lists to define, for example, accessibility, land use classes, ownership etc. These fields are set up in accordance to pre-defined classification schemes however the user is free to add, remove or modify them. Code lists can be used to classify a wide range of attributes such as administrative levels, personnel, topographic elements and many more.
Adding a Code list

Code lists can be added in two ways: "manually" or by importing a pre-prepared file in csv format.

To add a code list manually click on the Green "+" button (bottom left corner of the window) and proceed to fill in: Name [mandatory field] (e.g. region), Label: (e.g. AdmLev-1-Reg), and Description (any additional specification). Then select the List type: whether the code list is flat or hierarchical. A flat list has a binary structure while a hierarchical list allows for sub (nested) levels. [See list types examples below].

Once a code list has been named, the actual items (codes and labels) should be entered. Code and Label of each item of the code list can be added by clicking the smaller green "+" button and filling in Code [mandatory field] (e.g. "1") and Label (e.g. Northern). This should be repeated for as many items there are in the Code list. Each item can then be edited (click on edit icon) or deleted (red "-" )button.
The user can also wish to include a Specify field by clicking the appropriate box. In this way, it will be possible to specify unlisted values in a text input field that will appear next to the code list item in the data entry form.

In order to keep a separate record of the code list, it is suggested to Export the code list by clicking the Export button (at the bottom of the window). This will automatically generate and download a csv file.

In the case of hierarchical Code lists, multiple levels can be added by clicking on the "Add level" button.

While the manual process for adding code lists is perfectly legitimate and can be used for simple lists, for more complex list (e.g. list that have many items or with a hierarchical structure) it is strongly recommended to prepare them in advance and upload them into Collect by clicking the Import button (at the bottom of the window). When clicking Import a pop up window will open and the user can import a code list (flat or hierarchical) from a csv file. From the same window the user can also download an example file, while clicking on the blue "i" button will open a window with specifications on how to set up the csv file.

To import the csv file, click on Select file, locate the file within your directory and then click on Start import. Upon successful upload an Import completed message will appear, click ok and proceed with a new upload or, if done, click close at the bottom of the window.
To import multiple csv files at the same time, create a compressed folder containing the files and select “Batch import”. The code lists will be visible once the import is completed like when importing single files.

Example of Flat Code list

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>item_code</td>
<td>item_label_en</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logging</td>
<td>Logging</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fire</td>
<td>Fire</td>
<td></td>
<td></td>
</tr>
<tr>
<td>grazing</td>
<td>Grazing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>gardening</td>
<td>Gardening</td>
<td></td>
<td></td>
</tr>
<tr>
<td>other_human_impact</td>
<td>Other</td>
<td></td>
<td></td>
</tr>
<tr>
<td>none</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example of Hierarchical Code list - Vegetation types

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>use_code</td>
<td>use_label_en</td>
<td>type_code</td>
<td>type_label_en</td>
</tr>
<tr>
<td>forest</td>
<td>Forest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>forest</td>
<td>Forest</td>
<td>forest_subdivision</td>
<td>Forest subdivision (example)</td>
</tr>
<tr>
<td>unknown_forest</td>
<td>Unknown_Forest</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cropland</td>
<td>Cropland</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cropland</td>
<td>Cropland</td>
<td>crop_subdivision</td>
<td>Crop subdivision (example)</td>
</tr>
<tr>
<td>other_land</td>
<td>Other Land</td>
<td></td>
<td></td>
</tr>
<tr>
<td>grassland</td>
<td>Grassland</td>
<td>grassland_subdivision</td>
<td>Grassland subdivision (example)</td>
</tr>
<tr>
<td>other_land</td>
<td>Other Land</td>
<td>other_example</td>
<td>Other (example)</td>
</tr>
<tr>
<td>wetland</td>
<td>Wet Land</td>
<td></td>
<td></td>
</tr>
<tr>
<td>wetland</td>
<td>Wet Land</td>
<td>wetland_example</td>
<td>Wetland example</td>
</tr>
<tr>
<td>other_land</td>
<td>Other Land</td>
<td>other_wetland</td>
<td>Other wetland</td>
</tr>
<tr>
<td>noData</td>
<td>No data reason</td>
<td></td>
<td></td>
</tr>
<tr>
<td>noData</td>
<td>No data reason</td>
<td>sea</td>
<td>Sea</td>
</tr>
<tr>
<td>noData</td>
<td>No data reason</td>
<td>clouds</td>
<td>Clouds</td>
</tr>
<tr>
<td>noData</td>
<td>No data reason</td>
<td>other_reason</td>
<td>Other reason</td>
</tr>
</tbody>
</table>
Example of Hierarchical Code list - Administrative units

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>province_code</td>
<td>province_label_en</td>
<td>province_description_en</td>
<td>district_code</td>
<td>district_label_en</td>
<td>district_description_en</td>
<td>lgc_code</td>
<td>lg_label_en</td>
</tr>
<tr>
<td>2</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>Ballimoo Urban LLG</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Western</td>
<td>1</td>
<td>Middle Fly District</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Western</td>
<td>1</td>
<td>2 North Fly District</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Western</td>
<td>1</td>
<td>3 South Fly District</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Western</td>
<td>1</td>
<td>3 South Fly District</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Western</td>
<td>1</td>
<td>3 South Fly District</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Western</td>
<td>1</td>
<td>3 South Fly District</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Western</td>
<td>1</td>
<td>3 South Fly District</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>Gulf</td>
<td>2</td>
<td>1 Kerema</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>Gulf</td>
<td>2</td>
<td>2 Kikori</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>Gulf</td>
<td>2</td>
<td>2 Kikori</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>Gulf</td>
<td>2</td>
<td>2 Kikori</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

11.3 SPECIES LIST

This tab allows the user to add one or more species lists by uploading a file in csv format. The list should contain all the taxonomical species that the user will select from during data entry. Species lists should be as comprehensive as possible and should be created using all resources available in the country: taxonomical books, experts’ knowledge etc. For security reasons it will not be possible to add species during data entry. Instead, additional species should be added to the csv file (by a system administrator) and the updated species list uploaded to Collect.

**Adding a Species list:**

The species list (CSV file) file should be prepared in advance following the formatting requirements indicated below.

The required columns are:

- **code**: alphanumeric unique identifier code associated to the species
- **family**: family name of the species
- **scientific_name**: latin name of the species (genus + [optionally] species name + [optionally] subspecies name e.g. *Pinus* spp. or *Pinus radiata*).

**Note**: An Example file can be downloaded by clicking the button at the bottom centre of the screen.

Optionally the user can specify other columns, as ‘no’ for a numeric unique identifier associated to the species, the vernacular names of the species using the language code (in 3
characters ISO-639-2 format, e.g. 'swh' for Swahili, 'eng' for English) as header of these columns. The user can specify synonyms using “synonyms” or "lat" as header (Latin language). Multiple values are allowed for vernacular names and they need to be separated with a slash (/) character.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>no code</td>
<td>family</td>
<td>scientific_name</td>
<td>swh</td>
<td>eng</td>
</tr>
<tr>
<td>2</td>
<td>OLE/CAP/macarupa</td>
<td>Oleaceae</td>
<td>Olea capensis ssp. macarupa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>OLE/EUR</td>
<td>Oleaceae</td>
<td>Olea europaea</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>OLE/EUR/cuspidata</td>
<td>Oleaceae</td>
<td>Olea europaea subsp. cuspidata</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ALB/SCH/amaniensis</td>
<td>Fabaceae</td>
<td>Albizia schimperiana var. amaniensis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ALB</td>
<td>Fabaceae</td>
<td>Albizia spp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>AF2/QUA</td>
<td>Fabaceae</td>
<td>Afzelia quanzensis Welw.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Mbambakoofi</td>
<td>Fabaceae</td>
<td>Alizia adiantherifolia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Mgerengo / Mchani</td>
<td>Fabaceae</td>
<td>Alizia glaberrima (Shumach.&amp;Thonn)Benth</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>BOLI/PET</td>
<td>Borreriaeae</td>
<td>Bororria petualaris(Lam.)Thulin / Syn.Ethrelia petualaris Lam.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Mpanda jongoro</td>
<td>Borreriaeae</td>
<td>Bororria petualaris(Lam.)Thulin / Syn.Ethrelia petualaris Lam.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Bombed/RHO</td>
<td>Bombacaceae</td>
<td>Bombax rhodognaphalton K.Schum. / Syn.Rhodognaphalton schu Mswifl mwitu</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When the csv file is ready for upload, click **New** and specify a Name for the list. Then click **Import**, select the file to be uploaded and wait for the confirmation window. The screen will now show the records present in the species list. Species lists can also be Renamed, Deleted or Exported. If needed, additional species lists can be created and uploaded in the same manner.

### 11.4 SAMPLING POINT DATA

This tab is used to define the list of coordinates of each sample point location. This can be done by importing a CSV file that should be prepared following the formatting indicated below:

The csv file should contain:

- **levelX_code columns**: 1 column for each level X (maximum 3 levels, e.g. if you have 2 levels, cluster and plot, you will have a column "level1_code" with cluster id value and a column "level2_code" for plot id value);
- **x**: easting
- **y**: northing
- **srs_id**: id of the coordinate reference system, the same used as in the Coordinate Reference Systems settings of the survey

The csv file can be uploaded by clicking on the **Import** button. An Example file can be downloaded by clicking the button at the bottom center of the screen. If needed, the file can also be Exported.
Additional columns (to a maximum of 30) can be added to the csv file in order to record additional information related to the sampling points (e.g. slope, administrative units, etc.). These values can be used as source data for expressions or calculated values in the survey definition.

**Example.** Sampling Point data contains information about province, and in schema there is a (code) attribute with calculated expression to retrieve the province code (see below).
11.5 Schema

This tab constitutes the core of the Survey design. It is at this stage that the user defines every item (entity) and related attributes that should be measured. Before starting to work on the Schema it is necessary to have a very clear idea of the logical structure of the survey and a detailed list of the variables to be measure during field work (and in general during the survey) and also decide on the optimal way of measuring each variable.

The first step is to define one (or more) Sampling unit (the highest level of the hierarchical structure of the sampling design, typically the cluster, which contains plots. If the user started to work on the Survey Designer by selecting a blank template, it is now necessary to assign a name to the Record type (click on icon to the right of “Change it to your main tab label”) and give it an appropriate name (typically Cluster), then click Apply. Then don’t forget click Save (at the bottom of the page).

When you name objects, like a survey or attributes, you have the change to define the object’s labelling for other environments. Select “Other labels” and you can define how the object is labeled in other environments, such as in Saiku or Collect Mobile. This could be useful if you have a small screen on the device you plan to use Collect Mobile on, and you might want the labels to be shorter.

At this point Tabs, Entities and Attributes can be added.

Tabs do not have a role in the hierarchical and logical structure, rather they are used to arrange the way the data entry form will appear. At any time click on Preview to display the layout of the data entry form. If more tabs are present, they will be visible at the top of the form.

Entities: When adding an Entity, the user can choose among three types: Single (grouping), Multiple (form layout) or Multiple (table layout).

- **Single (grouping):** A Single entity is used to record an item (variable) that will appear only once (with a relationship 1:1 with its parent Entity). For example: start date (the
Date when the measurement work started on any given plot, will necessarily have a 1:1 relationship with its parent entity (for example, the plot). The term “grouping” means that a single entity can also be used to group a number of attributes all depending on the same entity (in our example, starting time, ending time etc.).

- **Multiple (Form or Table layout):** Multiple Entity are used when the relationship is 1: N meaning that there will be more instances of that entity all related to the parent entity. For example, plot (there will be more plots all referring to the parent entity Cluster, or, another example, tree: there will be more trees referring to the parent entity Plot. The difference between Form and Table layout refers to the graphical representation that the user wishes to give to that entity in the data entry form.

**Attributes** are always added in relation to an Entity and define the way in which the data is recorded. Attribute types are the following (each one with a unique recognizable label).

- **Boolean:** check/un-check
- **Code:** refers to a previously added Code lists
- **Coordinate:** allows to enter geographical coordinates (See Sampling design section)
- **Date:** entered in dd/mm/yyyy format
- **File:** allows uploading a file (e.g. photo, audio file, video, document, etc.)
- **Number:** numeric value (integer or real)
- **Range:** allows to define a numerical range
- **Taxon:** allows to add a record from a previously added Species list
- **Text:** allows to add free text
- **Time:** entered in hh:mm format

For each of the items that the user wishes to measure in the field, it is necessary to determine whether it is best expressed as an entity or an attribute and of which kind. As an example, a few variables measured at the Cluster level are listed below, indicating the type of considerations needed to determine whether it should be expressed as an entity or attribute.

- **Cluster No.** is intended as a unique numerical identifier for each cluster, therefore it should be added as Number attribute.
- **Accessibility:** if it is used to indicate whether the Cluster was accessible or not, then it could be entered as a Boolean attribute (yes/no); in the case the user wishes to be able to include other “categories” of accessibility, thus it will be entered as a Code attribute.
- **Starting Position:** is clearly a Coordinate attribute as it indicates the geographical coordinates of where field work for a specific cluster started from.
- **Time Study**: the intention here is to record multiple entries simultaneously: date and starting and ending time of the field work. In this case the most appropriate way is to enter it as a Multiple Entity in Table layout.

- **Remarks**: for a space where notes can be added, clearly a Text attribute.

To add **Entities and Attributes** right-clicking on the previously created *record type folder* and make the appropriate choice.

**Adding entities:**

![Survey Designer](image)

**Adding attributes:**

![Attribute Selection](image)

While adding entities and attributes, the user is required to fill in a number of related characteristics, in the main window as for example shown in the image below.
### 11.5.1 Description of Parameters

Some of the parameters are common for different types of Entities or Attributes, while others are unique to a specific type. Below is a list of groups of parameters and related explanation.

<table>
<thead>
<tr>
<th><strong>General</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Attribute's or Entity's name in lower case. Should be quite short and simple as functions within the survey use this name.</td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td>For Boolean (Y/N/Blank or Y/Blank); for Number (numerical type: integer / real); for Text (text box style: short= one row / memo= text box)</td>
</tr>
<tr>
<td><strong>Multiple</strong></td>
<td>Allows multiple entries for this attribute</td>
</tr>
<tr>
<td><strong>Min count</strong></td>
<td>Fixes minimum number of entries allowed</td>
</tr>
<tr>
<td><strong>Max count</strong></td>
<td>Fixes maximum number of entries allowed</td>
</tr>
<tr>
<td><strong>Required</strong></td>
<td>If checked, the record can be saved only if a valid value is entered for this attribute</td>
</tr>
<tr>
<td><strong>Required when</strong></td>
<td>Expression that specifies when the attribute is required (when &quot;Required&quot; is not checked)</td>
</tr>
<tr>
<td><strong>Hide when not</strong></td>
<td>If checked, the field will be hidden in the data entry form when the</td>
</tr>
</tbody>
</table>
relevant condition is not verified

Is key
If checked, this attribute serves as a unique identifier

Code list
Select a code list already created or create one

- Strict
If checked, once code list is defined, no additional code options can be added

Calculated
If checked, the input field won’t be editable and the value will be generated at runtime according to the specified Default Value

Qualifier
If checked, the attribute can be used to filter records (e.g. if there is a qualifier defined in the User Group)

Show in record summary list
If the attribute is a qualifier, by checking this box it will be shown in the record summary list

Max size MB
Defines max allowed size for file upload

- Allowed extension
Defines allowed extension types for file upload

Species list
Select a species list

- Highest rank
Defines highest rank in species taxonomy (family, genus, species, subspecies, variety)

Auto-complete group
(Only for text attributes) if specified, text auto complete will retrieve entries from the list of previously digitized values

Show row number in tables
If checked, it allows to show row number in tables

Show count in record summary list
If checked, it allows to Show count in record summary list

Units of measurements (Only for numerical attributes)

- Is default
Specify which unit of measure is default

Unit
Select from the list of unit of measurement (defined in the survey tab)

Decimal digits
Controls number of decimal digits that will be displayed

Labels and description

Single instance
Label if single instance

List heading
Label if multiple instances

Field number
Additional labeling field

Description
Free text description

Versioning

Introduced in
Allows keeping track of since which survey version a specific version Entity/Attribute has been added to the schema.
<table>
<thead>
<tr>
<th>Removed since version</th>
<th>Allows keeping track of since which survey version a specific Entity/Attribute has been removed from the schema.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Layout</strong></td>
<td></td>
</tr>
<tr>
<td>Column</td>
<td>Determines the position of the node (entity/attribute) in the field form layout.</td>
</tr>
<tr>
<td>Column span</td>
<td>Determines the number of columns that the node (entity/attribute) should span across *** (click 'i' button to view an example)</td>
</tr>
<tr>
<td>Width</td>
<td>Sets the width of the column ((blank=default))</td>
</tr>
<tr>
<td>Label width</td>
<td>Sets the width of the label ((blank=default))</td>
</tr>
<tr>
<td>Default Value</td>
<td>Used to determine how to handle empty values when record is submitted from data entry phase to data cleansing phase</td>
</tr>
<tr>
<td>Constant value</td>
<td>Value that will replace empty field</td>
</tr>
<tr>
<td>Expression</td>
<td>Expression that will calculate the value to replace empty field</td>
</tr>
<tr>
<td>Apply when</td>
<td>Sets the condition for when the replacement should occur</td>
</tr>
<tr>
<td>Checks</td>
<td>Used to automatically flag errors during field data collection (using Collect Mobile) or during data entry. Using checks greatly minimizes data entry mistakes and facilitates data cleansing</td>
</tr>
<tr>
<td>Type</td>
<td></td>
</tr>
<tr>
<td><strong>Comparison</strong></td>
<td>checks the value entered against a value or range of values (see examples after this table).</td>
</tr>
<tr>
<td><strong>Custom</strong></td>
<td>freely customizable check with an expression</td>
</tr>
<tr>
<td><strong>Distance</strong></td>
<td>(only for coordinate attributes) checks that the coordinate entered is within previously specified limits (see an example after this table).</td>
</tr>
<tr>
<td><strong>Pattern</strong></td>
<td>(only for text attributes) checks that the text is entered according to a predefined format</td>
</tr>
<tr>
<td><strong>Uniqueness</strong></td>
<td>checks that the entered values is unique (not repeated) within a specified group of values</td>
</tr>
<tr>
<td>Severity</td>
<td></td>
</tr>
<tr>
<td><strong>Error</strong></td>
<td>blocks the data entry user from submitting the record to data cleansing</td>
</tr>
<tr>
<td><strong>Warning</strong></td>
<td>assigns a warning flag to the record but does not prevent submission to data cleansing</td>
</tr>
<tr>
<td>Message</td>
<td>Message to be displayed to flag error/warning (see an example below)</td>
</tr>
<tr>
<td>Apply when</td>
<td>Allows to specify when the check is applicable (see an example below)</td>
</tr>
<tr>
<td>Expression</td>
<td>Expression that defines the logic of the check</td>
</tr>
</tbody>
</table>

Once all entities and attributes have been entered and defined, the hierarchical structure of the survey will look similar to the image below.
View Mode: Data Structure vs. Entry Form

While constructing the hierarchical structure of Entities and Attributes, and their organization in Tabs (folders), it is possible to switch the View mode: **Data Structure** or **Entry Form**. The first option will open all the branches and sub-branches to reveal every element and its properties (as in the image above), while the second option will collapse the elements into their respective Folders (image below). The same process can be achieved by clicking the small buttons [+/-] to Expand or collapse all nodes. This option becomes useful as the structure becomes more and more complex. The two arrows pointing up and down are used to move nodes accordingly.
NOTICE: If you plan to use the survey in Collect Mobile, use Data Structure view to plan and see how the items will be shown in Mobile, because Collect Mobile ignores the tabs and shows the attributes grouped by entities. This is important part of design work when organizing the order of entities and attributes for data collection using tablet.

At any time, it is possible to visualize the appearance of the data entry form by clicking on the Preview button.
11.5.2 Examples of “RELEVANT WHEN” Rules

“Relevant when” rule can be used activate/deactivate or hide/unhide entities and attributes. Here are some examples, see also Appendix 1.

1) Entity ‘tree’ is hidden if plot is not accessible.

2) Attribute “stem form” is available if a tree is living (\texttt{tree\_status}='L') and DBH more or equal to 40 cm (\texttt{tree\_dbh}>=40).

3) Trees and stumps are collected in the same form. Stump diameter is only recorded for stumps (i.e. attribute ‘stump’ is TRUE).
4) Trees and stumps are collected in the same form. “Causative agent” is only recorded for trees which are not healthy (numeric value of code ‘tree_condition’ is greater than 1).
11.5.3 Examples of default attribute values

Default attribute values can speed up collection of data. These default values can be constants or expressions. Specific ‘Apply when’ conditions can be set as needed.

1) Trees are in the survey as \textit{cluster/plot/tree} and we need to get tree numbers automatically. Then we can apply \texttt{math} function as follows.

2) Region name is given in one column of ‘Sampling Point data’ and we want get this name automatically after the cluster id is entered. This case we need to create a text attribute which is tagged as ‘Calculated’.

And the calculated value expression under section ‘Collect Earth’ is as follow.
3) We want to get plot date automatically. Apply the expression `idm:currentDate()`.

4) And in the previous image there is the attribute ‘plot_start_time’ which we can get automatically. The expression is `idm:currentTime()`.

5) In case of a Boolean attribute, we set a default value TRUE or FALSE as below.
11.5.4 **EXAMPLES OF DATA VALIDATION RULES**

Collect uses the XPath programming language for programming validations. See Appendix 1 and online documentation for more examples of its use.

1) Tree diameter (DBH) check. Diameter is a numeric attribute (real).

And the first rule.

The second rule.
And the third rule.

2) Bole height validation rules are as follows:
   - Should be always less than tree top height (‘tree_top_height’) if top height is given, and
   - Should be always less than 30 m.

And in the edit mode this look as follows:
3) Plot location check for inputted coordinates.

In ‘Sampling point data’ there is given a list of clusters, plots and plot coordinates, as follows.

And we need validation rules for attribute ‘location’, i.e. for plot coordinates. Please notice that the attribute ‘location’ belongs to entity ‘general_information’ which is under ‘plot’ in this survey.
And the *Error* validation rule looks in the edit mode as presented below.

Where `parent()/parent()` refers to ‘cluster’ and `parent()` refers to ‘plot’ (entity) in the inventory data model (*idm*).
11.5.5 Add a node from another survey

In Collect schema you use a part of schema made in another survey. You can add that node for a tab or for an entity. In the following example we copy an entity and all its child attributes from another survey under a selected tab. First we right click the tab, then select “Add node from another survey”. Collect will show a list of all existing surveys, then we select entity ‘tree’ to be copied. If there are coded variable(s) in the entity of interest, Collect will also copy the code list(s).

The "another survey" can be published or temporary: if there is only the published copy of the survey, that one will be used, otherwise the survey name will have a suffix "(temporary)" and that version of the survey will be used.
11.6 **ADVANCED FUNCTIONS**

Select a survey from the list and advanced functions comes visible.

- **Publish**: Publish survey for Data Management. Includes any changes you have put into the survey since it was last published.

- **Unpublish**: Only for published surveys. Allows you to edit the survey, but **destroys all existing records for it**. Useful if you accidentally publish a survey before it is ready.

- **Clone**: Make a copy of an existing survey.

- **Delete the selected survey**: if the survey has been published, you will be warned that deleting the survey will delete even all the associated data / records. If the survey has been published and there is also a temporary copy of it, selecting this item will delete only the temporary copy: the published survey and the associated data will remain.
12 DATA CLEANSING

The Data Cleansing section on the Home page of *OF Collect* contains features that are most applicable when using *OF Collect Earth*. It allows running queries on the database and performing bulk updates. In the context of a data flow from *OF Collect Mobile* to *OF Collect Desktop* alternative ways should be followed.

The validation checks built in during the survey design will ensure a high quality data set. However, a process of data cleansing is always needed, for example to spot outliers, to check min and max values and standard deviation. At this stage, a quantitative data cleansing should be performed outside of Collect.

As shown before, running a Validation report allows working with a list of “issues” (error and or warnings) that should be dealt by the data manager by contacting the field crew leaders and eventually updating the records with correct values. This could be seen as a qualitative cleansing. Data Cleansing should only be used with Collect if massive changes need to be implemented, such as changing values from a faulty code list or an incorrect Spatial Reference System has been used for the data.

12.1 BASIC CONCEPTS

In essence, Queries find records according to a set of criteria. These records can then be “repaired” by Data Cleansing Chains, which either calculate or substitute the old data with correct values.

<table>
<thead>
<tr>
<th>Concept</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Query Type</td>
<td>Identifies what kind of queries will be available, useful when grouping queries into categories (typos, incomplete records, changed code).</td>
</tr>
<tr>
<td>Data Query</td>
<td>Defines a query on the data, criteria to find values among the collected records.</td>
</tr>
<tr>
<td>Data Query Group</td>
<td>A group of queries that can be used to run them in bulk to save time.</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data Report</td>
<td>Generates a list of record values respecting the condition of a Query Group, i.e. a list of entries that have certain kinds of errors.</td>
</tr>
<tr>
<td>Data Cleansing Step</td>
<td>Changes the values of attributes to another value according to conditions of a query (constant or expression)</td>
</tr>
<tr>
<td>Data Cleansing Chain</td>
<td>Groups a set of Cleansing Steps for executing and gives a summary of records involved in the cleansing process.</td>
</tr>
</tbody>
</table>

12.2 DATA CLEANSING WORKFLOW

1. Backup your data,
2. Define one or more Data Query types,
3. Define Data Queries,
4. Group Queries into Data Query Groups,
5. Generate a Data Report for a query group,
6. For each Data Query, define one or more Data Cleansing steps,
7. Define a Data Cleansing Chain that groups the desired steps,
8. Run the Data Cleansing Chain,
9. Generate a new Data Report to verify that the records have been fixed,
10. Start over from step one if necessary.

12.2.1 DATA BACKUP

See Chapter 12 for instructions on backing up data. To start Data Cleansing, go to Data management and submit the records to cleansing from the advanced functions menu.

12.2.2 DEFINE DATA QUERY TYPES

Select the green “New” button to add a Data Query type. Remember that these should be general types of mistakes, not specific things to correct. Specify the name and code for this Query Type in the pop up window. Note that the code can be either letters or numbers, but it has to be unique for that specific survey. To update a Query Type, select the blue icon in the right end of the row. To delete a Query Type, select the (red) cross symbol in the right end of the row. You have to reload the browser page for the deleted types to disappear.
12.2.3 Define Data Queries

Data Queries are defined as XPath expressions, written into the “Conditions” box. Start by selecting the node you want to target from the Entity and Attribute lists. Then select what type of query this should be (from the ones created in Query Types) and select if this error is a warning or an error. For help in programming the condition, see Appendix 1.

To see if a Query works, go to the “Test and Run” tab. Select the appropriate record step (the phase of data management at which your data is now, usually Data Cleansing now) and click “Run”. A summary of successfully located records appears on the window.
12.2.4 **GROUP QUERIES INTO DATA QUERY GROUPS**

For efficiency, Data Queries will be run as groups. To create a group, click on the green “New” button and a pop up will appear. You form a group by selecting the relevant Queries from the left box and clicking on the arrows in the middle. To move all Queries at once, click on the uppermost or down most arrow. You can select specific Queries by holding down the CTRL-key and clicking them.
You can view the included Queries by clicking on the plus sign next to the Query Group’s name.

12.2.5 GENERATE A DATA REPORT

Before this step make sure the survey you are cleaning is up to date (no unpublished changes) and submitted to data cleansing. When creating a report, choose the correct record step.

To view the contents of the Data Report, click on the blue edit sign.
You can view the results in browser, or export them as a CSV. The report is stored in the system and is used for checking that the data cleansing operation works as expected. After running your Data Cleansing Chains, produce a new Data Report and check if the records have been fixed.

12.2.6 Defining Cleansing Steps for Queries

Cleansing Steps are created just like Queries, only that now you assign Queries to cleansing steps. A cleansing step searches for records based on its Query, but it also changes or deletes the record according to its Cleansing type:

**Update attribute**: Changes the value of any attribute that matches the condition set by the Query being used. The replacement can be a single value or
Delete entity: Deletes any entities that match the condition set by the Query being used. This is useful for removing large batches of irrelevant data.

Delete record: Deletes any records that match the condition set by the Query being used.

12.2.7 Defining a Data Cleansing Chain

Notice that the order of steps is important. In the picture above, the cleansing process moves down through the data structure (from tract to plot to subplot). This tidies up the process; excess tracts are removed first, so the plots and subplots on them don’t have to be cleansed.

To change the order of the steps, select a step and move it with the “Move step up” and “Move step down” buttons. To remove a step, click on the (red) cross symbol on the right. To run the Cleansing Chain, select a record step and then “Run”.

After you have run all the Data Cleansings you need, go back to the Data Reports tab and create new reports for your data.
As you can see, the Data Cleansing chain has removed all the faulty records. **Note that the report only shows the kinds of records that are relevant to the conditions of the Queries in them.** In the picture above, for example, you can only see the count of records that are related to tree allometry. There might still be other kinds of faults in the data, so a Data Report should be produced for every Cleansing Chain that was executed.

**MAKE SURE YOU HAVE AT LEAST ONE BACKUP OF YOUR DATA BEFORE YOU CLEANSE IT.** If something goes wrong, or cleansings don’t work the way they should, there is no other way to get back your data except through backups.
13 BACKUP

You can create a downloadable backup of your survey in the backup section. Select the (published) survey you want to create a backup for and click “Generate new backup”. A ".collect-backup" file is created.

Should something bad happen to your survey, you can restore it in the Restore tab. Selecting in Additional Options “Validate records” the process updates all the error and warning counts, meaning that the counts are updated based on the backup’s conditions. If you wish to restore as quickly as possible, select “Delete ALL existing records before restore”. It is a good idea to store all backups into cloud storage or onto separate hard drives.
14 SAIKU

Saiku is web-based open source software that facilitates data visualization and data querying. To launch Saiku in Collect, select it from the Home view. A pop-up will appear. Select the survey you want to analyze in the dropdown menu. The survey has to be published.

When you have chosen a survey, click on *Generate* to format the Saiku analysis. Do this if you are handling a new survey or have made changes or added data since the last analysis. You can check the up-to-dateness of the data by looking at the *Updated records* number.

For detailed instructions on how to use Saiku, refer to the Collect Earth documentation.

15 SECURITY (USER ACCOUNTS)

Security is the area of Collect where the inventory managers can manage the personnel involved in Survey workflow, assign roles and keep track of progress of data workflow.

15.1 USERS

From the Security tab of the home panel, you will have two tabs. By clicking on the Users tab, you will see the list of current users (if any) and will be able to create new user accounts through the green New button.

You will be then able to fill the new user form with details as appropriate. Then click Save. User’s details include Name and password and a check box for enabling access. Most importantly, each user will have a specified Role which will determine the level of access to different components of Collect and the ability to work on specific phases of data workflow.

Users’ roles:

- **View**: Lowest level of permission. Only data visualization and data export are allowed.
- **Entry limited**: Entry limited users can only modify existing data, but not add new records.
- **Entry**: Entry users have access only to data entry phase, they can enter new records, edit own records, submit them for cleansing and export records.

- **Cleansing**: Same as Data entry + permission to edit records in cleansing phase and submit them to data analysis.

- **Analysis**: All previous rights + the ability to unlock records and re-submit them to data cleansing phase.

- **Design**: this is a generic user, who can design and modify surveys but whose rights depend on the role assigned within each user group.

- **Administrator**: Full rights. All previous steps + Survey designer, Users management and ability to import data (records) in Data Management. If needed, more than one admin is allowed.

Clicking on one user from the list will show the details of that user. You can also click on one or more to delete it/them, through the red bin button.
### 15.2 User Role Permissions Table

<table>
<thead>
<tr>
<th></th>
<th>View</th>
<th>Entry Limited</th>
<th>Entry</th>
<th>Cleansing</th>
<th>Analysis</th>
<th>Design</th>
<th>Admin</th>
</tr>
</thead>
<tbody>
<tr>
<td>View records</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Create new records</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Edit records in Entry phase</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(only owned records)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Edit owned and not owned</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>records from Entry to</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analysis phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Submit records from Entry</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>to Cleansing phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(only owned records)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reject records from</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cleansing to Entry phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delete records (only in Entry</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phase)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Saiku analysis</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data Cleansing</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Survey Designer</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Backup/Restore</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Users/Roles management</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
15.3 Groups

From the Groups tab on the left side, you will see the list on user groups (if any).

To create a new group, you have to click on the green New button at the top. A page with the details of the group will appear.

You can assign a name, a label and a short description to the group, make it public or private, select a parent group to which you want to link it, and enable it to start a survey.
If a record attribute/code list filter is specified, the users with Entry role within the group will be able to insert and see only records having the specified attribute equal to the specified value. For instance, if the attribute name is “country” (a “country” attribute must already exist in one of the surveys), in the attribute/code value box you can specify the name of one country – for instance “Italy”, so that those records where “Italy” has been selected will be automatically assigned to that user group.

From the list of existing users at the bottom left, you can assign some to the group, by selecting them, choosing a role that they will have in that group, and clicking on the arrow button.

To avoid conflicts between more roles of one same user, within a group you should assign to him a role that has equal or less rights that he usually has. For instance, if a user has an “entry” role at a general level, within a specific group you will not be able to assign to him the role of “analysis”, since the general level role defines the limits of his rights.

### 15.4 ROLE IN GROUP PERMISSIONS TABLE

<table>
<thead>
<tr>
<th></th>
<th>Viewer</th>
<th>Operator</th>
<th>Data Analyst</th>
<th>Supervisor</th>
<th>Owner</th>
<th>Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>View records (read only mode)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Create new records</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Edit records</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Delete records (in entry phase)</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Import records</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Change record owner</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Promote/Demote records in bulk</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
ANNEX 1. EXAMPLES OF CODE IN OPEN FORIS COLLECT

The commands are based on two inheritance structures: one for a cluster survey and another one for single plots design.

**General hierarchical structures (either cluster or plot-based)**

**Comparison checks (i.e., start_date, start_time, end_time, tree_no, azimuth, tree distance, dbh, total_height and bole_height):**

Example attribute (number type): Path: /plot/details/elevation
Severity Error
Message Elevation must be between 0 and 4000 m
0 <= elevation <= 4000

Example attribute (date type): Path: /cluster/start_date
Severity Error
Message You have written the starting date in the future!
start_date <= idm:currentDate()

Example attribute (time type): Path: /cluster/start_time
Severity Warning
Message Working hours for the survey should be from 06:00 to 19:00!
0600 < start_time <= 1900

Example attribute (time type): Path: /cluster/end_time
Severity Error
Message End time should be always after start time
start_time <= end_time
(Here we assume that start_time and end_time are under the same entity, i.e., cluster, or plot)

Example attribute (number type): Path: /cluster/plot/tree/tree_no
Severity Error
Message tree number must be >0
0 < tree_no

Example attribute (number type): Path: /cluster/plot/tree/azimuth
Severity Error
Message angle must be between 0 (inclusive) and 360 (exclusive)
0 <= azimuth < 360

Example attribute (number type, refers to tree distance): Path: /cluster/plot/tree/distance
Severity Error
Message tree outside of plot. Plot radius is 12 m.
distance <= 12

Example attribute (number type): Path: /cluster/plot/tree/dbh
Severity Warning
Message Unusually large dbh
dbh < 200

Severity Error
Message dbh must be positive
0 <= dbh

Severity Error
Message tree must be at least 20 cm if distance>2 m
20 <= dbh Apply when distance>2
Severity  Error  tree must be at least 5 cm in any case  
5 <= dbh

Example attribute (number type): Path: /cluster/plot/tree/total_height
Severity  Warning  
Message Unusually tall tree  
total_height < 80

Severity  Error  Impossibly tall tree  
total_height < 150

Severity  Error  height must be positive  
0 <= total_height

Example attribute (number type): Path: /cluster/plot/tree/bole_height
Severity  Error  bole height must be positive  
0 <= bole_height

Severity  Error  bole height must be less or equal than total height  
bole_height <= total_height

Default values (i.e., start_date, start_time and tree_no):

Example attribute (date type): Path: /cluster/start_date
Calculated value expressions:  Expression:  idm:currentDate()

Example attribute (time type): Path: /cluster/start_time
Calculated value expressions:  Expression:  idm:currentTime()

Example attribute (number type): Path: /plot/tree/tree_id
Calculated value expressions:  Expression:  math:max(parent()/tree/tree_id) + 1  
(it will automatically increase the number every time a new tree is recorded in the plot)

Surveys with Plot as sampling unit or root entity and tree as sub-entity

Distance checks (i.e., plot position):

Example attribute: Path: /plot/details/position
Severity  Error  You are more than 50 m from plot location  
Destination point  idm:samplingPointCoordinate(parent()/plot_id)  
Max distance (m) 50

Severity  Warning  You are more than 20 m from plot location  
Destination point  idm:samplingPointCoordinate( parent()/plot_id)  
Max distance (m) 20

Autocompletion (click Calculated in the General tab for the variable) in optional variables from sample design file (i.e., province as a column after design points coordinates in the imported sampling_design file):

Example attribute (text type): Path: /plot/details/province
Calculated value expressions:  Expression:  idm:samplingPointData('province', parent()/plot_id)

Required-status of variable if expression is satisfied (i.e., tree_height only required every third tree):
Example attribute (text type): Path: /plot/tree/tree_height
Required only when expression is verified: \( \text{tree}_\text{id} \mod 3 = 0 \)

**Surveys with Cluster as sampling unit or root entity and plot and tree as sub-entities**

**Distance checks (i.e., plot position):**

Example attribute: Path: /cluster/plot/details/position
Severity: Error
Message: You are more than 100 m from plot location
Destination point: \( \text{idm:samplingPointCoordinate(parent()/parent()/cluster_no, parent()/plot_no)} \)
Max distance (m): 100

Severity: Warning
Message: You are more than 20 m from plot location
Destination point: \( \text{idm:samplingPointCoordinate(parent()/parent()/cluster_no, parent()/plot_no)} \)
Max distance (m): 20

**Autocompletion (click Calculated in the General tab for the variable) in optional variables from sample design file (i.e., province as a column after design point coordinates in the imported sampling_design file):**

Example attribute (text type): Path: /cluster/location/province
Calculated value expressions: Expression: \( \text{idm:samplingPointData('province', parent()/cluster_no)} \)

Example attribute (text type): Path: /cluster/plot/details/province
Calculated value expressions: Expression: \( \text{idm:samplingPointData('province', parent()/plot_no)} \)

**Autocompletion (click Calculated in the General tab for the variable) in tree-related variable (i.e., tree count):**

Example attribute (number type): Path: /cluster/plot/tree_count
Calculated value expressions: Expression: \( \text{math:max(parent()/plot/tree/tree_no)} \)

**Relevance of variable if expression is satisfied (i.e., tree_count, plot position, and tree entity):**

Example attribute (number type): Path: /cluster/plot/tree_count
Relevant only when expression is verified: \( \text{parent()/plot/details/accessibility='0'} \)

Example attribute (coordinates type): Path: /cluster/plot/details/position
Relevant only when expression is verified: \( \text{accessibility='0'} \)
(here we don’t need a hierarchical path since position and accessibility belong to the same hierarchy, i.e: parent()/plot/details/position and parent()/plot/details/accessibility)

Example entity (multiple table layout): Path: /cluster/plot/tree
Relevant only when expression is verified: \( \text{details/accessibility='0'} \)
**Relevance - Environment functions:**

- `env:desktop()` : returns true if Collect runs on a desktop computer, false otherwise (e.g. in Android)
- `env:mobile()` : opposite of `env:desktop()`

An example shown in the next image.
ANNEX 2. POSTGRESQL WITH COLLECT

Install PostgreSQL (v. 9.4 or newer), so that the password is **postgres**. Accept the localhost number **5432**. (If you have installed PostgreSQL in the server, you may need to change this. The server admin should know this). You do **not** need to install StackBuilder.

After installing Collect, go to this folder:

*C:*\Users\USERNAME\OpenForis\Collect

And take a safety copy of the file **collect.properties**

Change the settings to be as follows:

```properties
#settings for PostgreSQL connection
collect.http_port=8380
#db
collect.db.jndiName=jdbc/collectDs
collect.db.driverClassName=org.postgresql.Driver
collect.db.url=jdbc:postgresql://localhost:5432/collect
collect.db.username=collect
collect.db.password=collect123
collect.db.initialSize=5
collect.db.maxActive=20
collect.db.maxIdle=5
```

Then start Collect. If Collect Laucher gives an error and complains about access rights, check using pgAdmin that PostgreSQL database password (**postgres**) and localhost number (**5432**) are correct. If these are correct, refresh database view in pgAdmin and check whether there is database ‘**collect**’ created. If this is not there, you can do as follows:

1) Stop Collect
2) Using pgAdmin, create username ‘collect’, with password: ‘collect123’, and with super-user rights
3) Create database ‘collect’, with owner ‘collect’
4) Under this database, create schema ‘collect’, with owner ‘collect’
5) Start Collect
6) After server starting (see Server log in the launcher), refresh database view in pgAdmin
7) See if there are tables etc. under schema ‘collect’. Anyway, Collect should be running now in your web browser.

In rare cases Collect may not start with PostgreSQL. This problem may arise after an event, that while Collect was starting up something went wrong and you closed Collect. So if this
happens with and Collect does not start in the browser, check the following setting. First stop Collect (i.e. Tomcat service). In ‘collect’ database, schema ‘collect’, view table “databasechangeloglock”. If attribute ‘locked’ is TRUE, change this to FALSE and restart Collect.