open**foris**

Collect

User's manual

Ver. November 2016

www.openforis.org

CONTENTS

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".
- **Rich metadata**: XML format, Complex nested structure of the survey, Validation rules, Multiple Spatial Reference Systems.
- **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

Prerequisites

1. Administrator rights to install new software (on your desktop computer or server, as appropriate)
2. Web browser: **Google Chrome** is recommended.

   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/

Standalone Installation (single user environment)

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

3.1. WINDOWS SYSTEMS

- Download the installer from http://www.openforis.org/ -> Collect
- Run the .exe file and follow the instructions on the screen. Collect will be installed by default into ***C:\opt\openforis***
- If the installation completes successfully, you will have a group of shortcuts in the start menu, Open Foris Collect.

Start Collect

- In the start menu, select "All programs", then go into **Open Foris** folder and select Start-up OF Collect
- A command prompt window should appear, do not close it and wait until you read "Server startup in NNN ms"
- If a message from Windows Firewall appears saying that Java is trying to access the network, **allow it** (need to be done only once)
- After Collect Tomcat started up, select "All programs" from the start menu, then go into Open Foris Collect folder and select Open OF Collect in browser: a new browser window should be opened and Collect login page should appear on the screen. Enter the system 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:8080/collect
- By default, Collect is using 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 1.

Stop Collect

- Close all the opened browser windows that are using Collect
- In the start menu, go into All programs, then select Open Foris Collect folder and select Shutdown OF Collect

3.2. LINUX SYSTEMS

- Download the installer from http://www.openforis.org
- Run the .run file as a super user and follow the instructions on the screen. Collect will be installed by default into */opt/openforis*
- If the installation completes successfully, you will have 2 icons in the Desktop, *Start Open Foris Collect* and *Stop Open Foris Collect*

Start Collect

- Double click on the *Start Open Foris Collect* icon in the desktop
- Open the browser and access this url: http://localhost:8080/collect

Stop Collect

- Close all the opened browser windows that are using Collect
- Double click on the *Stop Open Foris Collect* icon in the desktop

Note: 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:

http://km.fao.org/OFwiki/index.php/Open_Foris_Collect_Installation

3.3. 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. Test versions are periodically created and tested among current users while Production versions (stable) are released for public use and made available. Following are the steps for updating Collect to the newest released version.

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 OF Collect" startup menu item and follow the instructions on screen
4. startup Collect

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

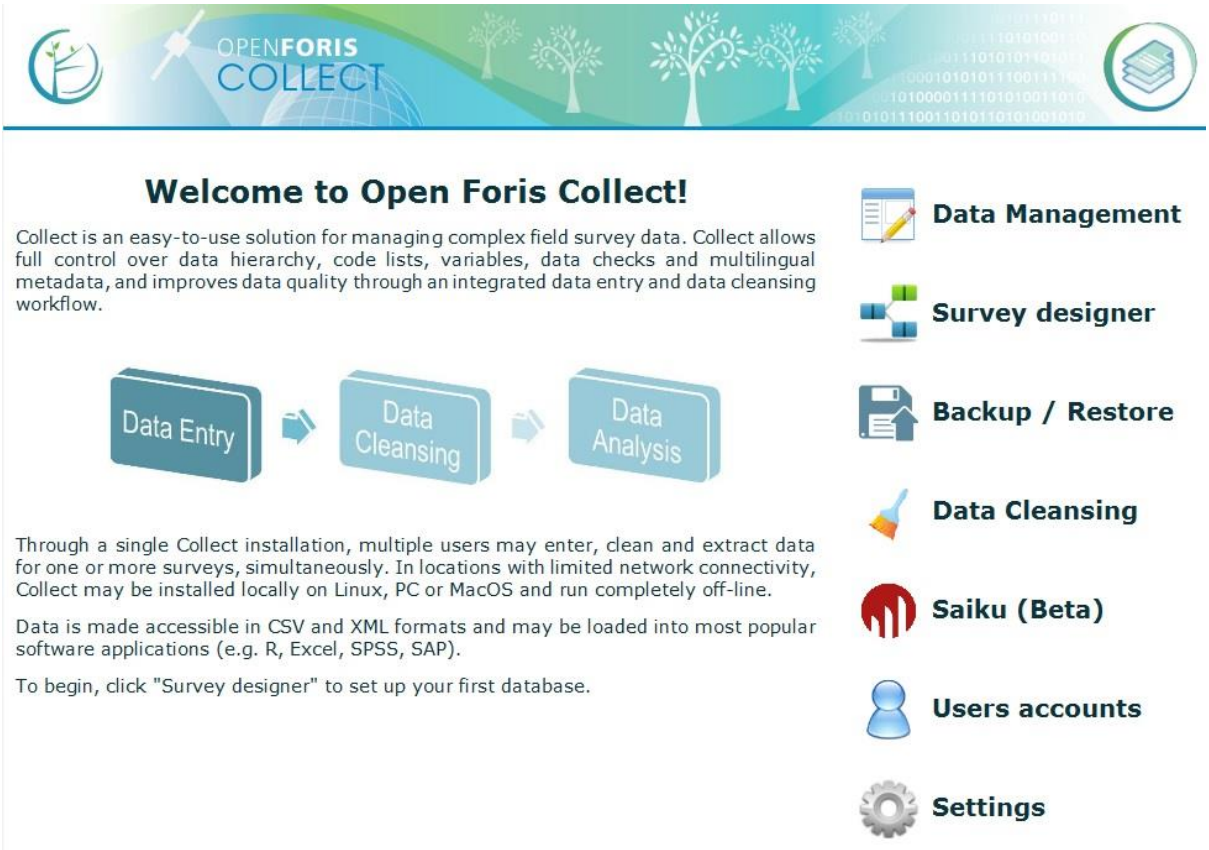
3. make a copy of the folder *data* (path: c:\opt\openforis\collect\tomcat\data) and store it outside of c:\opt\openforis, e.g. in your "user" folder
4. uninstall Collect
5. install Collect using the new installer
6. copy the folder *data* previously backed up into the folder c:\opt\openforis\collect\tomcat\
7. start Collect and verify that all old data are present

Upgrade Collect using the war file (for expert users only)

1. Stop the Collect Tomcat instance, if running
2. Create a backup copy of your existing Collect installation (c:\opt\openforis folder)
3. Create a backup of the PostgreSQL database (if you are using that as database)
4. Download the latest war file of Collect:
   http://www.openforis.org/nexus/service/local/artifact/maven/redirect?r=releases&g=org.openforis.collect&a=collect-webapp-tomcat&v=LATEST&e=war
5. Delete the folder collect-server/tomcat/webapps/collect
6. Delete the file collect-server/tomcat/webapps/collect.war, if exists
7. Copy the downloaded collect-web-VERSION.war file into collect/tomcat/webapps folder and rename it into collect.war. If you are using a unix or linux system, change collect.war owner and group to the same as tomcat user (e.g. Change the collect.war owner and group to "tomcat": sudo chown tomcat:tomcat collect.war)
8. Start the Collect Tomcat instance

Warning: if you are using SQLite database in your local machine, do never first uninstall Collect. 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 an SQL server for Collect, see Appendixes 1 & 2.

4. COLLECT HOME



Collect welcome page offers the following options: *Data management, Survey designer, Backup/Restore, Data cleansing, Saiku analytics, Users accounts and Settings.*

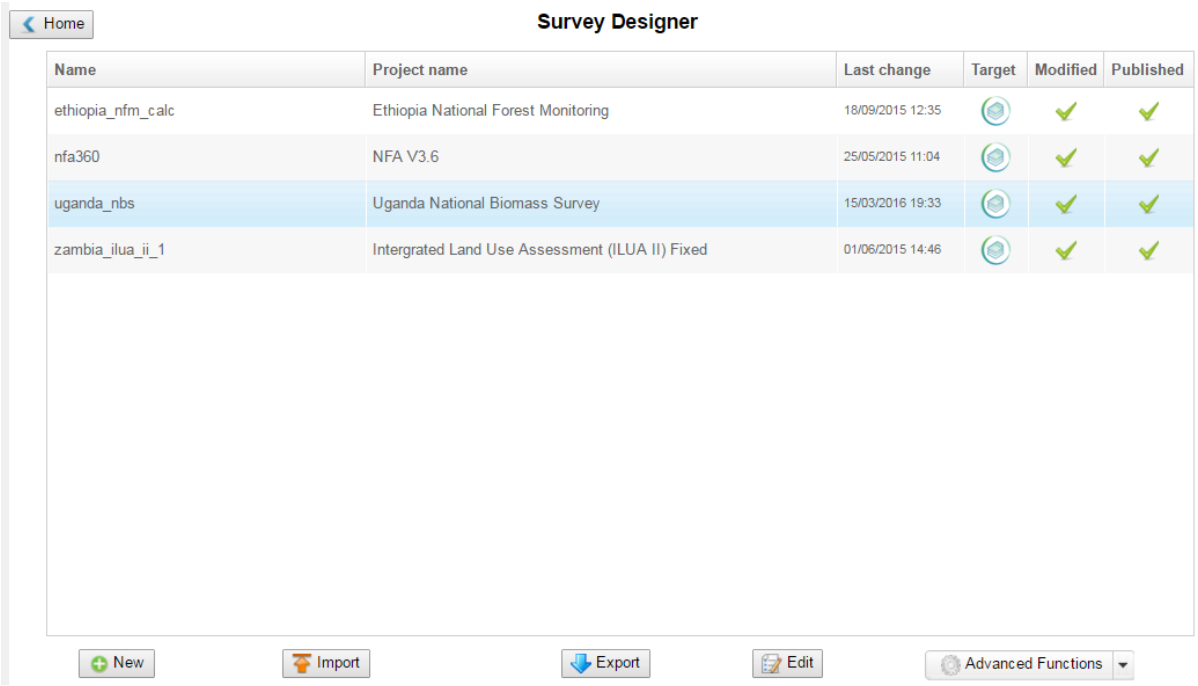
- **Survey designer**: the starting point for setting up and manage your survey. Select Survey designer to Create, Import, Export, Edit, Publish/Unpublish, Validate, Clone or Delete surveys.
- **Data management**: once a survey has been created select Data management to start entering data and manage data workflow. Stored surveys can be accessed and records can be managed and edited.
- **Backup/Restore**: used for data backup and restoring.
- **Data cleansing**: Collect Data Cleansing Toolkit can help you to find errors in the data and to fix them.
- **Users accounts**: create a personal user profile (password protected) specifying role in the workflow: *Data Entry, Data Cleansing, Data Analysis, Administrator*.
- **Settings**: customize upload path, record index path and graphical elements of Collect.

And in the **footer**: Collect current version and user login/logout

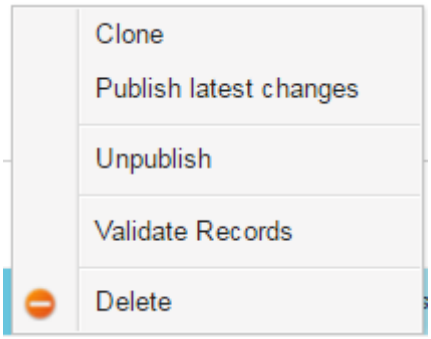
5. SURVEY DESIGNER

5.1. NEW SURVEY

In the Survey Designer Management Panel you will see the Surveys currently uploaded (if any) and their status: whether they have been *Modified* or *Published* as well as *Date of last change*. The operations available from this panel include: *+New, Import, Export, Edit and Advanced functions*. Details on these operations will be covered after the instructions on how to set up a new survey.

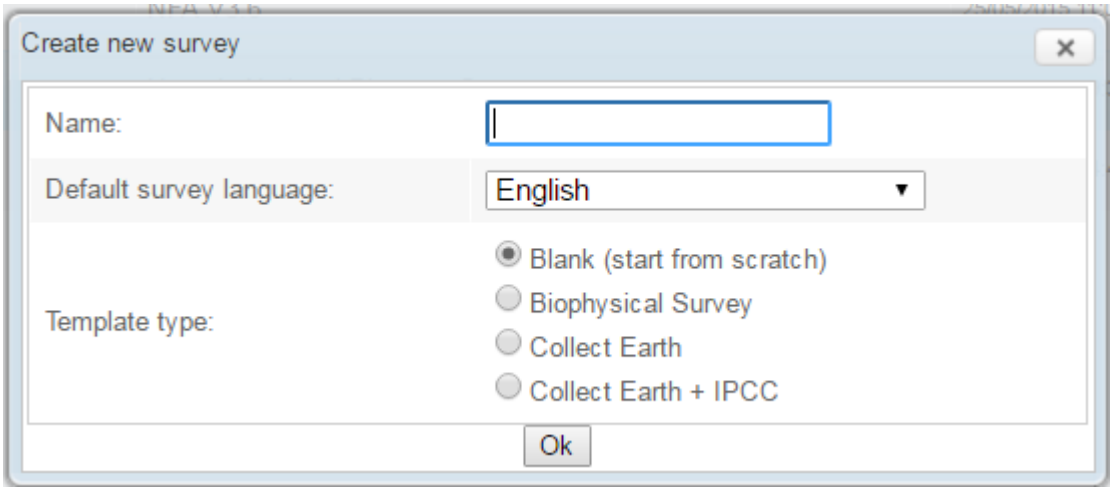


In Advanced functions there are the following tools:



To start setting up your first Survey click on **+New**.

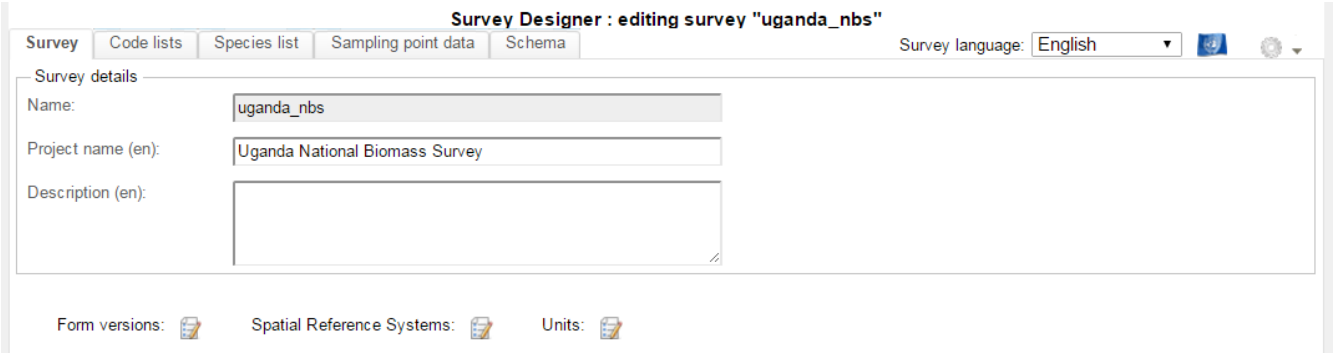
A pop up window will open requesting the user to enter a *name* for the survey, select the *default survey language* and choose from a *template type*. Selecting a *Blank* template means starting from scratch, with no preset information loaded. Alternatively, it is possible to choose from templates in 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 language will show a codified language abbreviation in parenthesis, for example (en) for English.

The user can now proceed to fill in the fields in each of the five tabs of the Survey designer: *Survey*, *Code lists*, *Species list*, *Sampling Design* and *Schema*.

5.2. SURVEY



In this tab 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.

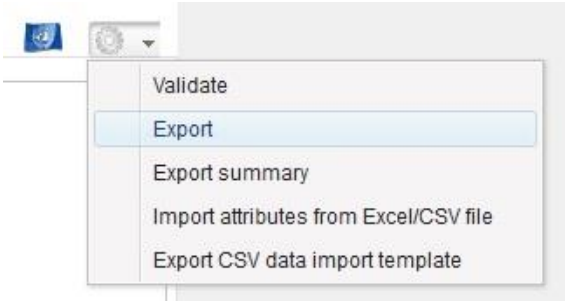
NOTE: Try to avoid using different form version in you survey. This feature can cause troubles in data analysis when using Open Foris Calc phase if different versions have different variables and/or data types.

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 (.ced) and SQL Relational database (.sql) formats.

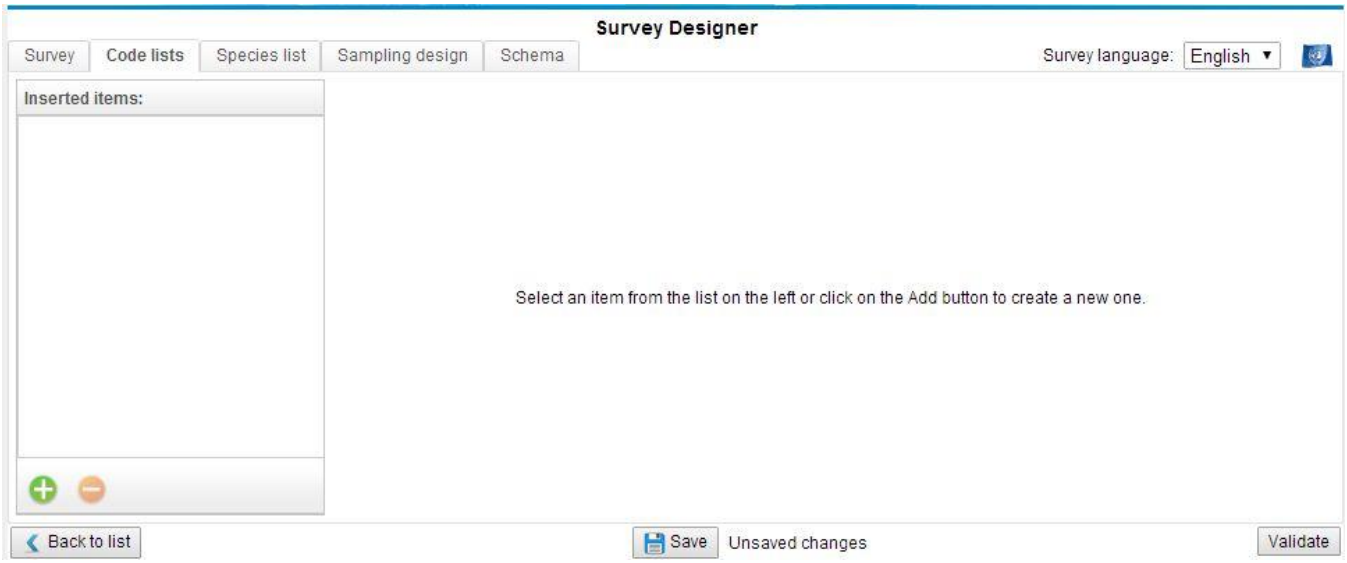
Export summary: Creates a CSV file that summarizes the data structure of the survey. This is useful for defining conditions for attributes' relevance, for example.

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.

5.3. 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.



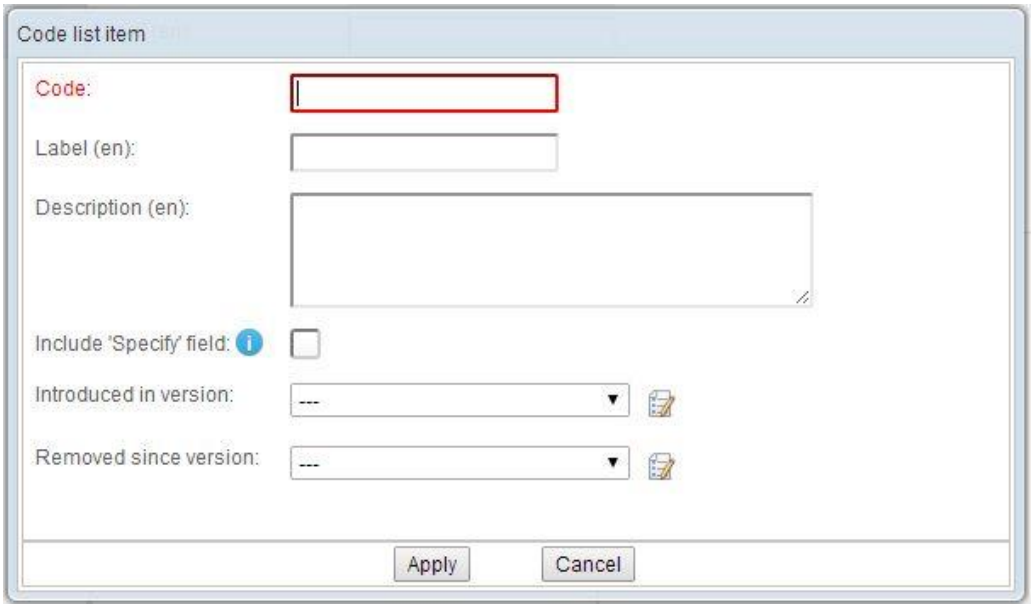
Adding a Code list

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

	A	B
1	item_code	item_label_en
2	logging	Logging
3	fire	Fire
4	grazing	Grazing
5	gardening	Gardening
6	other_human_impact	Other
7	none	None
8		

Example of Hierarchical Code list

	A	B	C	D
1	use_code	use_label_en	type_code	type_label_en
2	forest	Forest		
3	forest	Forest	forest_subdivision	Forest subdivision (example)
4	forest	Forest	unknown_forest	Unknown Forest
5	cropland	Cropland		
6	cropland	Cropland	crop_subdivision	Crop subdivision (example)
7	cropland	Cropland	unknown_crop	Unknown crop
8	grassland	Grassland		
9	grassland	Grassland	grassland_subdivision	Grassland subdivision (example)
10	grassland	Grassland	unknown_grassland	Unknown grassland
11	settlement	Settlement		
12	settlement	Settlement	settlement_example	Settlement (example)
13	settlement	Settlement	other_settlement	Other settlement
14	otherLand	Other Land		
15	otherLand	Other Land	other_example	Other (example)
16	wetLand	Wet Land		
17	wetLand	Wet Land	wetland_example	Wetland example
18	wetLand	Wet Land	other_wetland	Other wetland
19	noData	No data reason		
20	noData	No data reason	sea	Sea
21	noData	No data reason	clouds	Clouds
22	noData	No data reason	other_reason	Other reason
23				

5.4. 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. Additional species should instead 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:

- **no**: numeric unique identifier associated to the species
- **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 center of the screen.

Optionally the user can specify other columns for 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 "lat" as header (Latin language). Multiple values are allowed for vernacular names and they need to be separated with a slash (/) character.

	A	B	C	D	E	F
1	no	code	family	scientific_name	swh	eng
2	5	OLE/CAP/macrocarpa	Oleaceae	Olea capensis ssp. macrocarpa		
3	6	OLE/EUR	Oleaceae	Olea europaea		
4	7	OLE/EUR/cuspidata	Oleaceae	Olea europaea subsp. cuspidata		
5	12	ALB/SCH/amaniensis	Fabaceae	Albizia schimperiana var. amaniensis		
6	9	ALB	Fabaceae	Albizia spp.		
7	8	AFZ/QUA	Fabaceae	Afzelia quanzensis Welw.	Mbambakofi	Mahogany
8	10	ALB/ADI	Fabaceae	Albizia adianthifolia		
9	11	ALB/GLA	Fabaceae	Albizia glaberrima (Shumach.&Thonn)Benth	Mgerenge / Mchani	
10	13	BOU/PET	Boraginaceae	Bourreria petiolaris(Lam.)Thulin / Syn.Ehretia petiolaris Lam.	Mpanda jongoo	
11	14	BOM/RHO	Bombacaceae	Bombax rhodognaphalon K.Schum. / Syn:Rhodognaphalon schu	Msufi mwitu	
12						

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.

5.5. SAMPLING DESIGN

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 (see below). If needed, the file can also be Exported.

	A	B	C	D	E	F
1	level1_code	level2_code	level3_code	x	y	srs_id
2	7_81			792200	9484420	EPSG:21035
3	7_81	2		792200	9484420	EPSG:21035
4	7_81	3		792200	9484670	EPSG:21035
5	7_81	4		792200	9484920	EPSG:21035
6	7_81	5		792200	9485170	EPSG:21035
7	7_81	6		792450	9485420	EPSG:21035
8	7_81	7		792700	9485420	EPSG:21035
9	7_81	8		792950	9485420	EPSG:21035
10	7_81	9		793200	9485420	EPSG:21035
11	10_117			806680	9305020	EPSG:21035
12	10_117	6		805680	9305020	EPSG:21035
13	10_117	7		805930	9305020	EPSG:21035
14	10_117	8		806180	9305020	EPSG:21035
15	10_117	9		806430	9305020	EPSG:21035
16	10_117	10		806680	9305020	EPSG:21035

Additional columns (to a maximum of 10) 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.

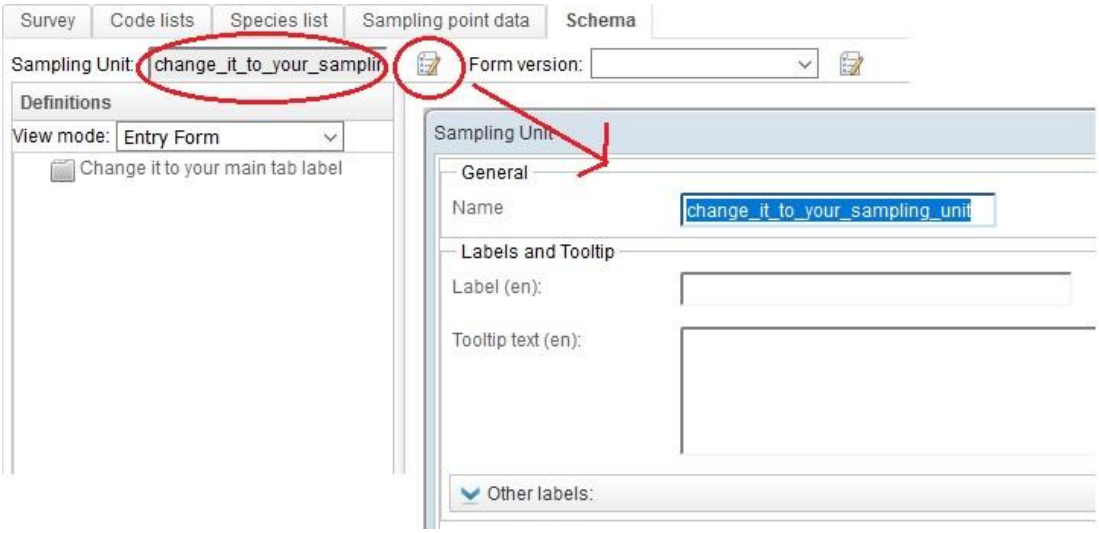
5.6. SCHEMA

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).

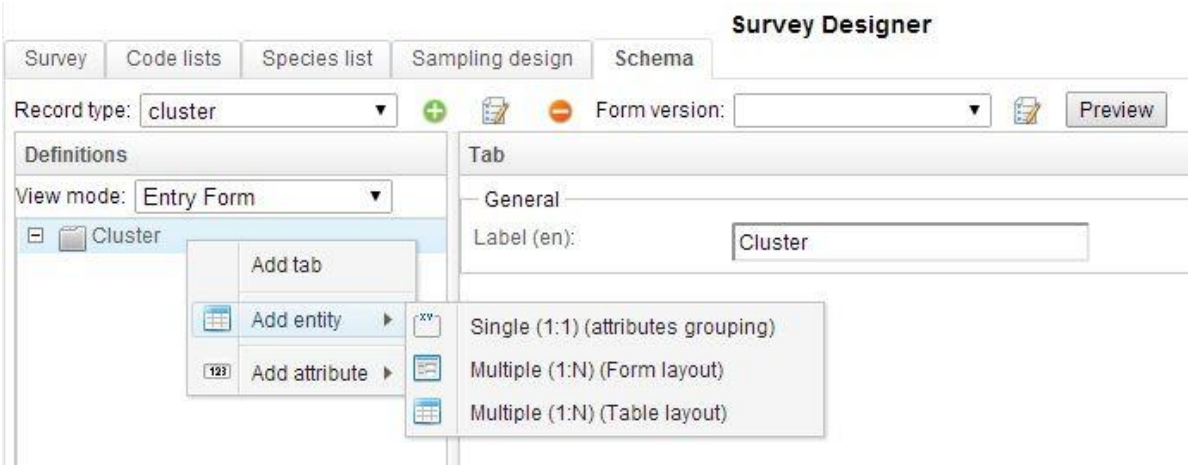
- **Boolean**: check/un-check
- **Code**: refers to a previously added *Code lists*
- **Coordinate**: allows to enter geographical coordinates (See *Sampling design* section)
- **Date**: entered in dd/mm/yyyy format
- **File**: allows to upload a file (e.g. photos, notes, etc.)
- **Number**: numeric 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.

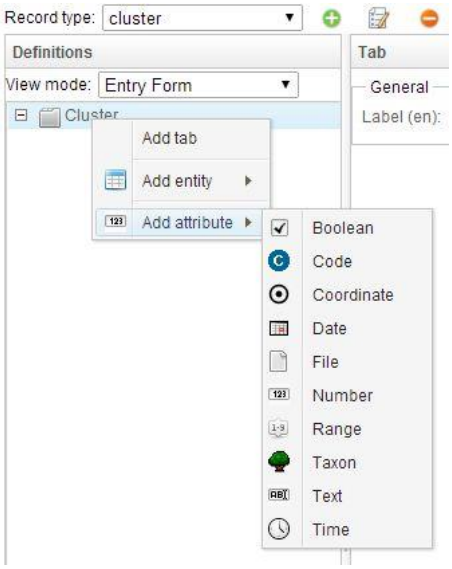
- *Cluster No*.: is intended as a unique numerical identifier for each cluster, therefore it should be added as Number attribute.
- *Accessibility*: if it is used to indicate whether the Cluster was accessible or not, then it could be entered as a Boolean attribute (yes/no); in the case the user wishes 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 information 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*: is simply allow for a space where notes can be added, clearly a Text attribute.

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

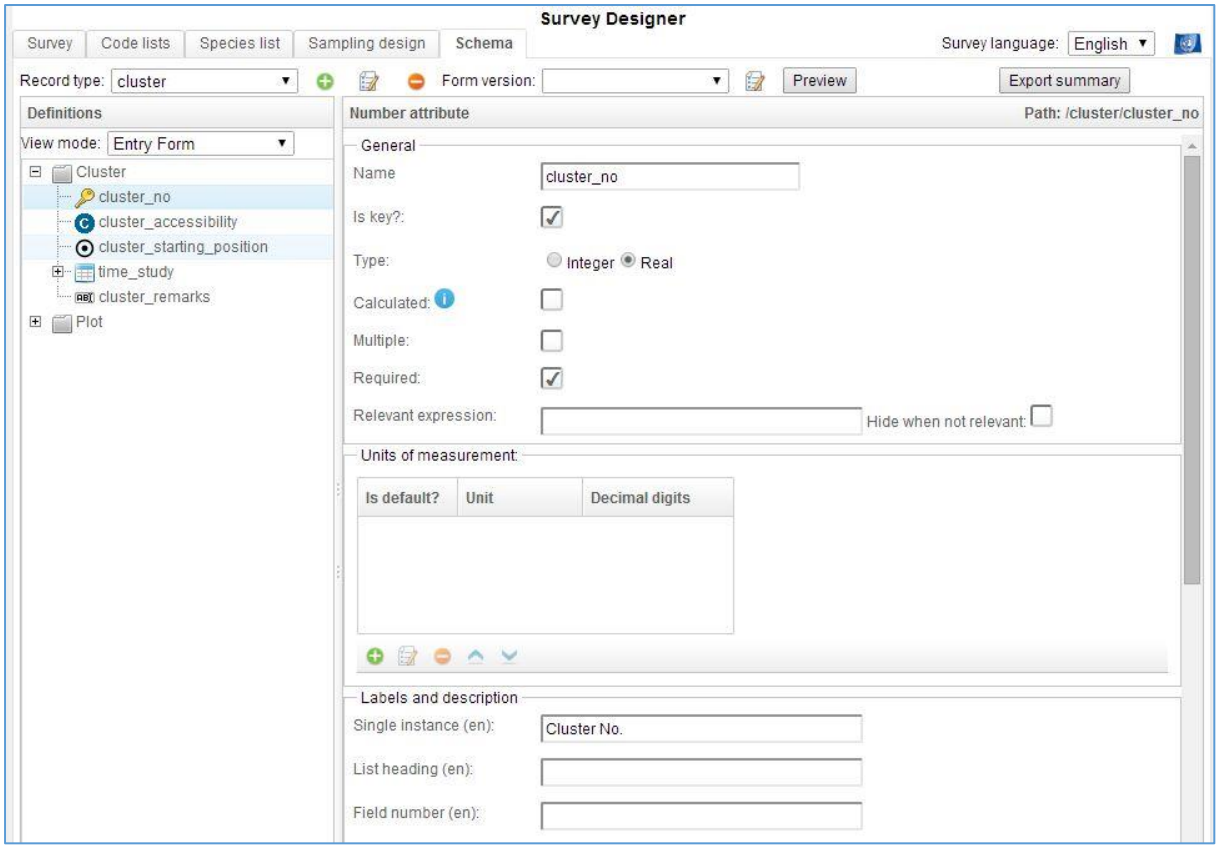
Adding entities:



Adding attributes:



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.



5.6.1 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.

General

Name	Attribute's or Entity's name in lower case. Should be quite short and simple as functions within the survey use this name.
Type	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)
Multiple	Allows multiple entries for this attribute
- Min count	Fixes minimum number of entries allowed
- Max count	Fixes maximum number of entries allowed
Required	If checked, the record can be saved only if a valid value is entered for this attribute
- Required when	Expression that specifies when the attribute is required (when "Required" is not checked)
Relevant when	Function that determines under what conditions the field is applicable

- Hide when not relevant	If checked, the field will be hidden in the data entry form when the 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
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, sub-species, 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	Specifies 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 version	Allows to keep track of since which survey version a specific Entity/Attribute has been added to the schema.
Removed since version	Allows to keep track of since which survey version a specific Entity/Attribute has been removed from the schema.
Layout	
Column	Determines the position of the node (entity/attribute) in the field form layout.

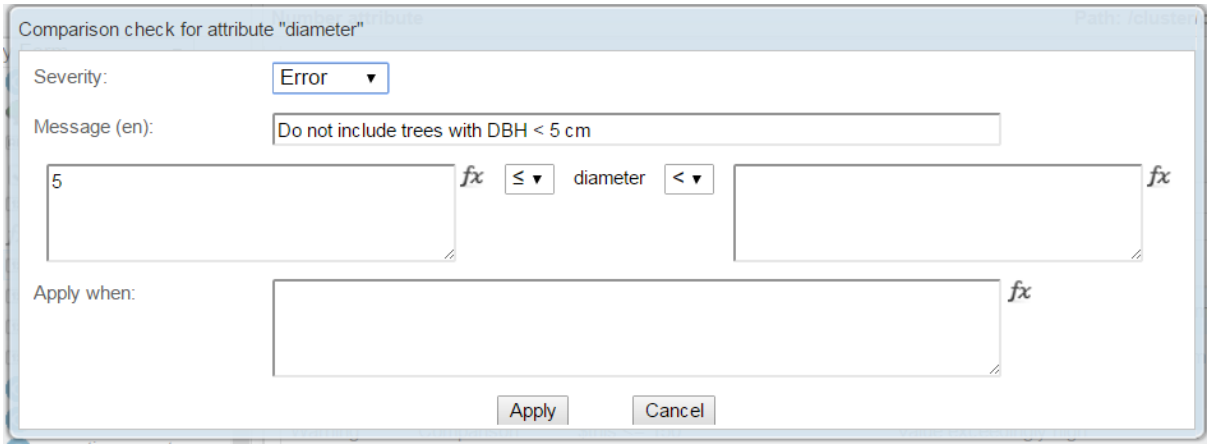
Column span	Determines the number of columns that the node (entity/attribute) should span across *** (click 'i' button to view an example)
Width	Sets the width of the column ((blank=default)
Label width	Sets the width of the label ((blank=default)
Default Value	Used to determine how to handle empty values when record is submitted from data entry phase to data cleansing phase
Constant value	Value that will replace empty field
Expression	Expression that will calculate the value to replace empty field
Apply when	Sets the condition for when the replacement should occur
Checks	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
Type	**Comparison** - checks the value entered against a value or range of values (see examples after this table). **Custom** - freely customizable check with an expression **Distance** - (only for coordinate attributes) checks that the coordinate entered is within previously specified limits (see an example after this table). **Pattern** - (only for text attributes) checks that the text is entered according to a predefined format **Uniqueness** - checks that the entered values is unique (not repeated) within a specified group of values
Severity	**Error** - blocks the data entry user from submitting the record to data cleansing **Warning** - assigns a warning flag to the record but does not prevent submission to data cleansing
Message	Message to be displayed to flag error/warning (see an example below)
Apply when	Allows to specify when the check is applicable (see an example below)
Expression	Expression that defines the logic of the check

5.6.2 EXAMPLES OF DATA VALIDATION RULES

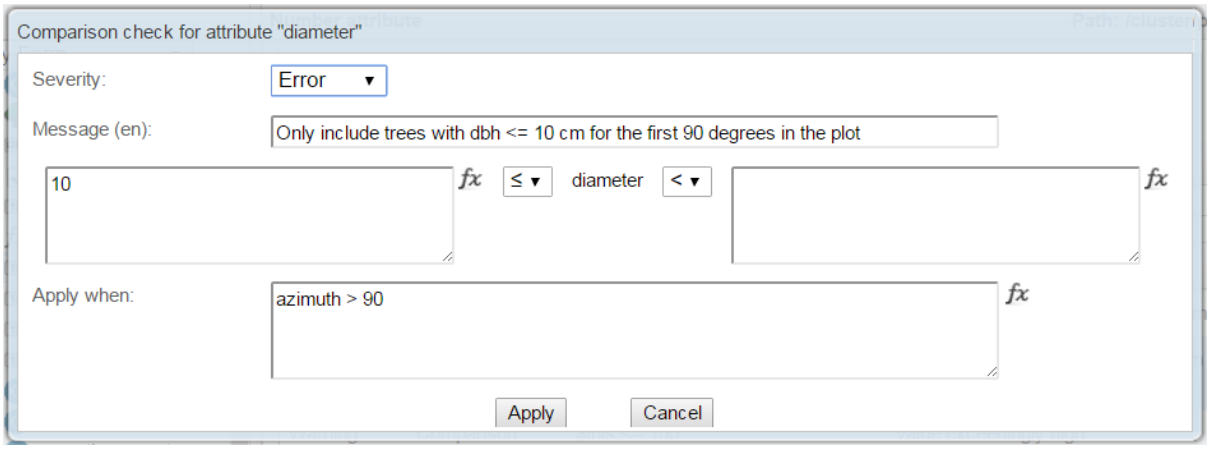
Collect uses the XPath programming language for programming validations. See Appendix 3 and online documentation for more examples of its use.

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

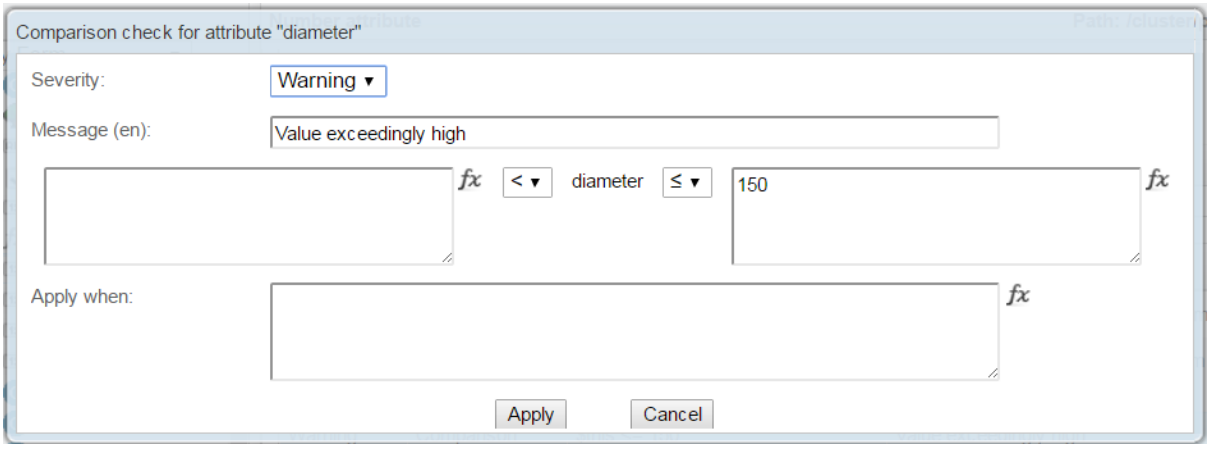
And the first rule.



The second rule.

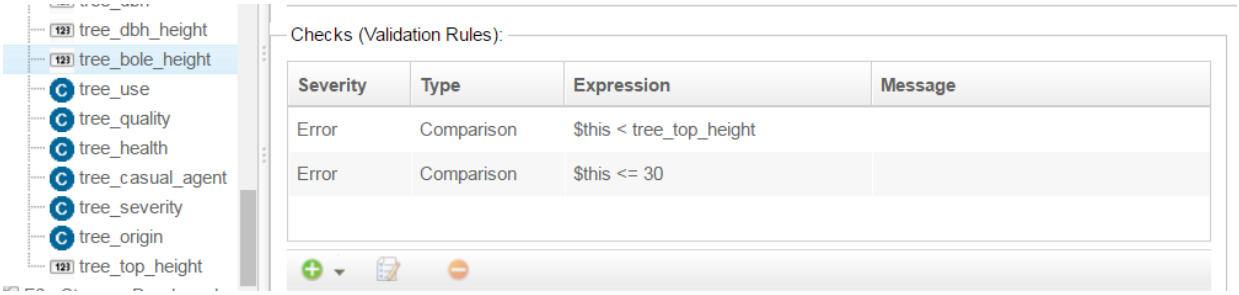


And the third rule.

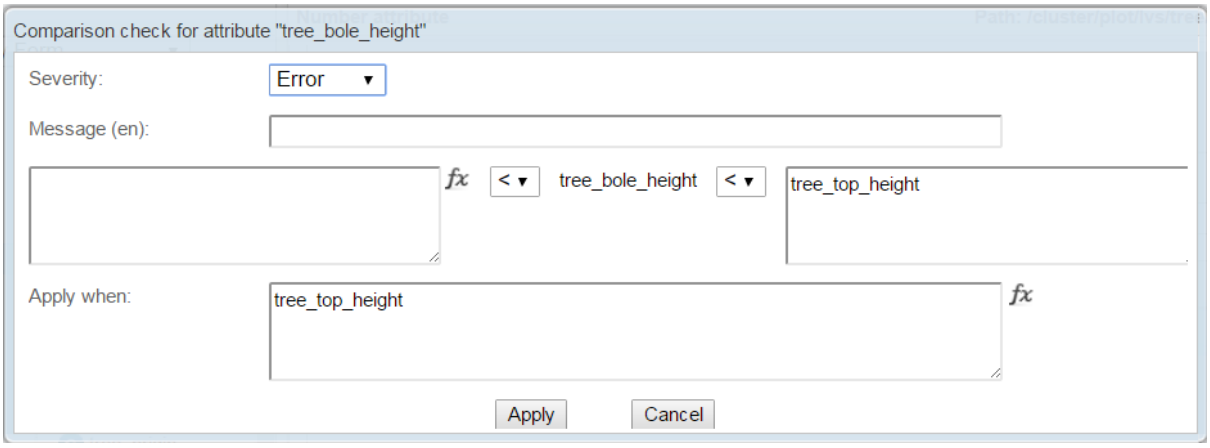


2) Bole height validation rules are as follows:

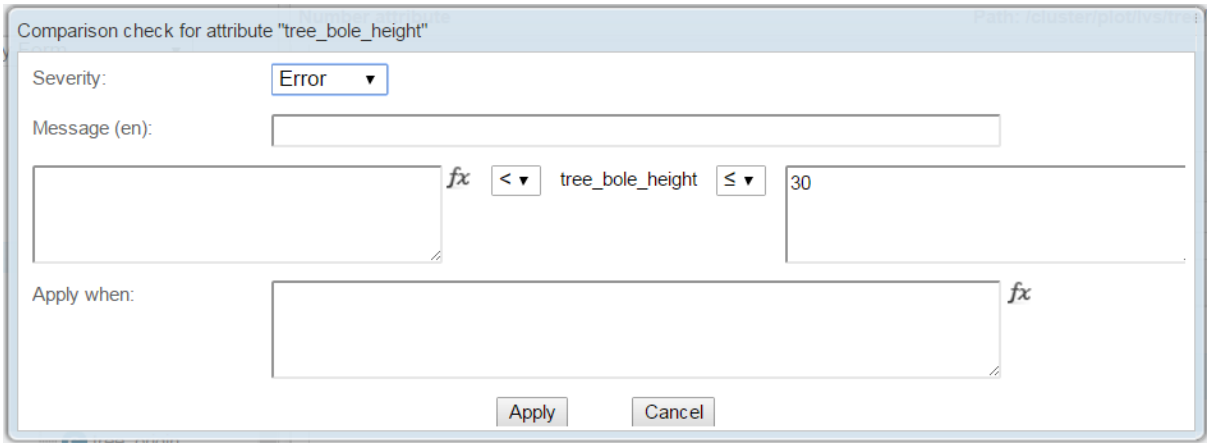
- Should be always less than tree top height ('tree_top_height') if top height is given, and
- Should be always less than 30 m.



And in the edit mode this look as follows:

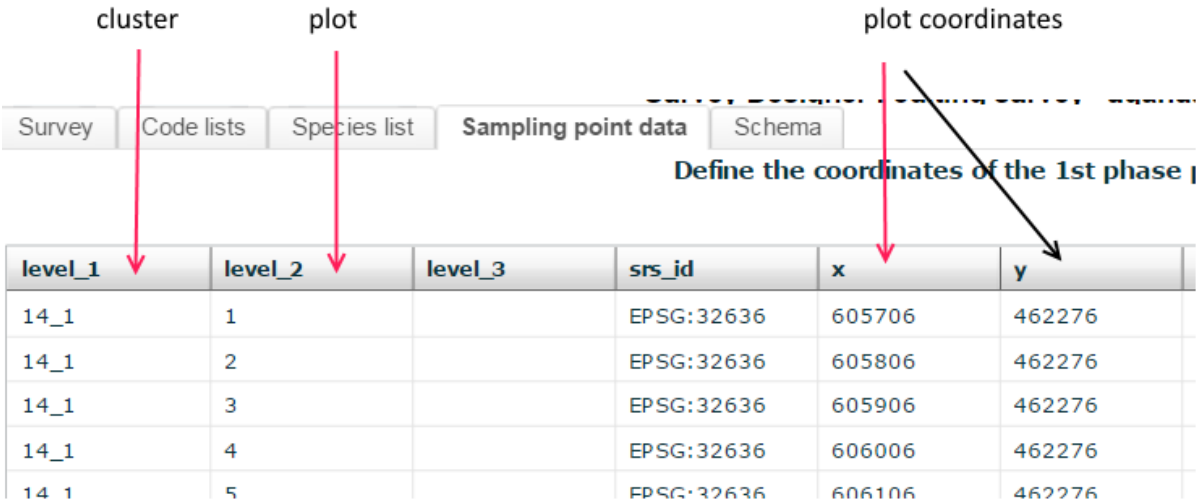


And the second rule.



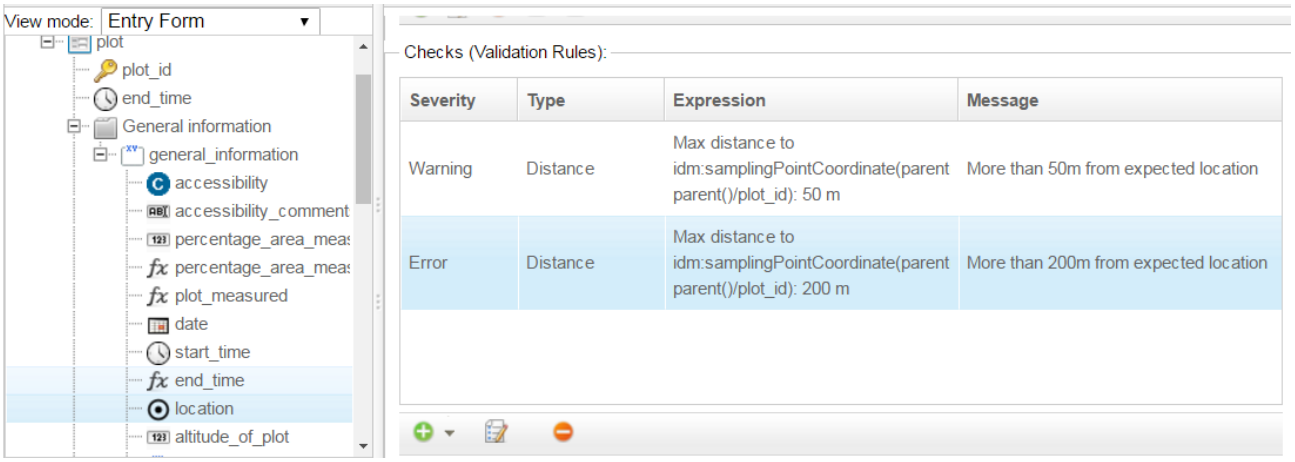
3) Plot location check for inputted coordinates.

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

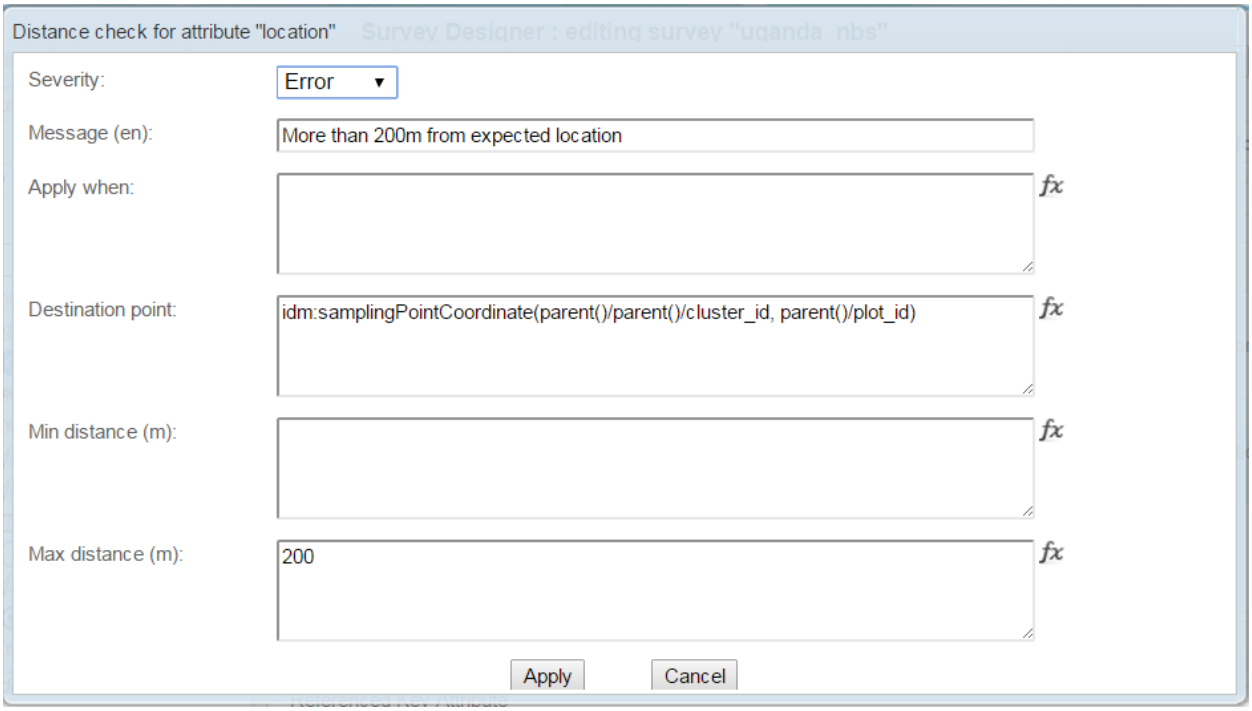


level_1	level_2	level_3	srs_id	x	y
14_1	1		EPSG:32636	605706	462276
14_1	2		EPSG:32636	605806	462276
14_1	3		EPSG:32636	605906	462276
14_1	4		EPSG:32636	606006	462276
14_1	5		EPSG:32636	606106	462276

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.



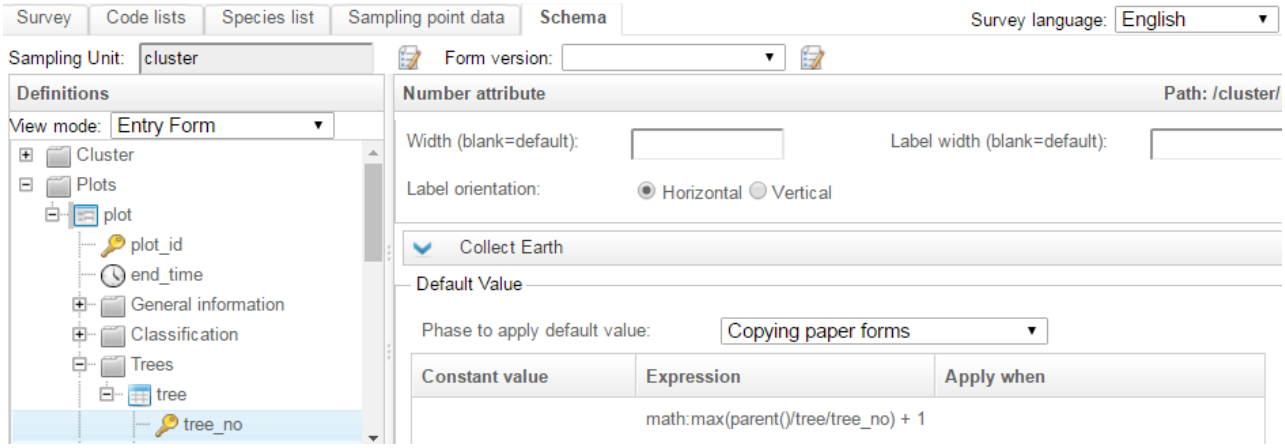
And the *Error* validation rule looks in the edit mode as presented below.



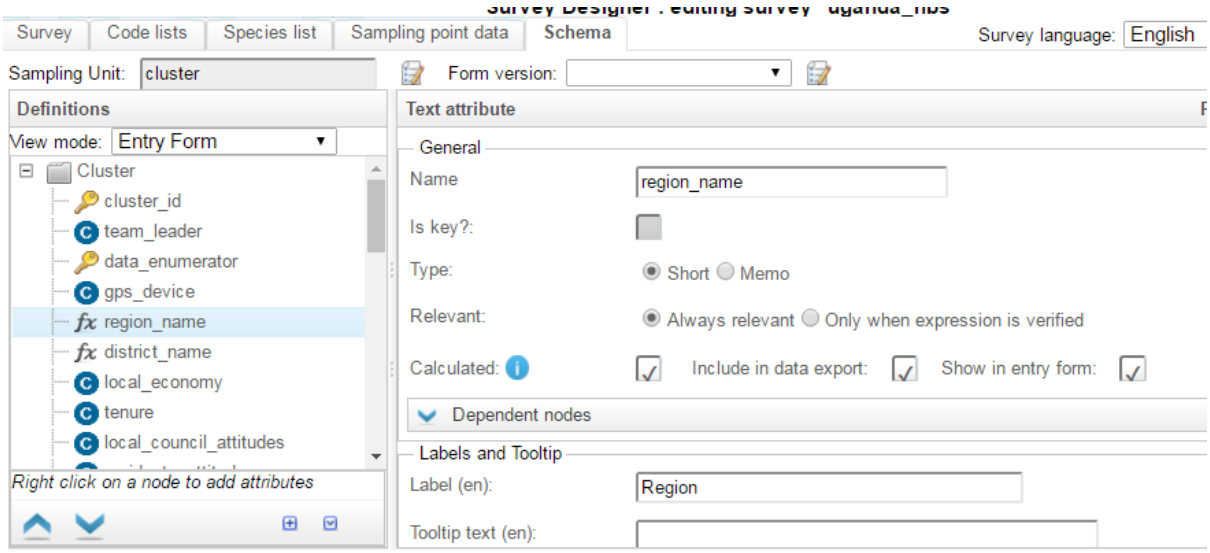
Where *parent()/parent()* refers to 'cluster' and *parent()* refers to 'plot' (entity) in the inventory data model (*idm*).

4) Trees are in the survey as *cluster/plot/tree* and we need to get tree numbers automatically.

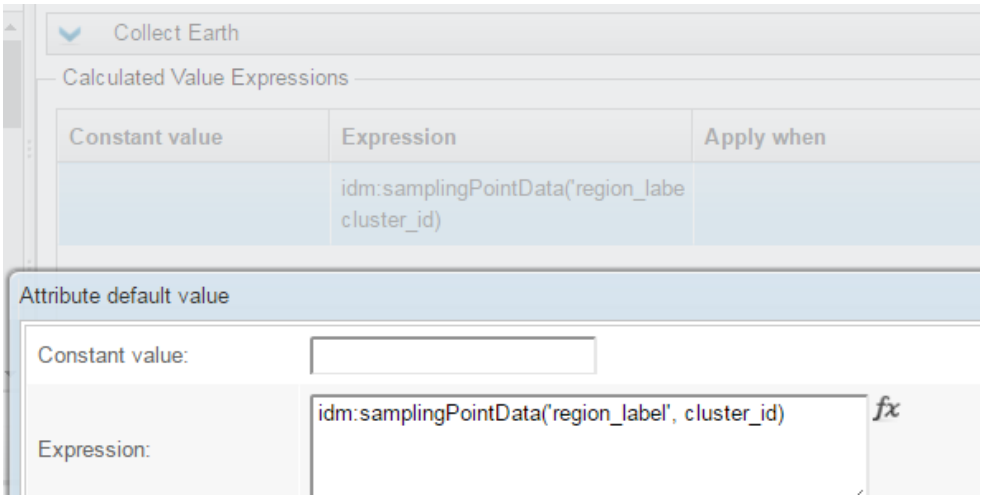
   Then we can apply '***math***' function as follows.



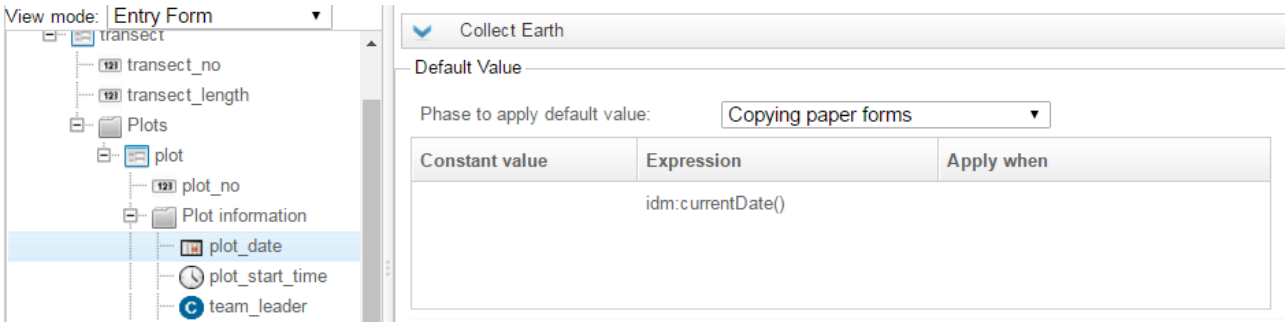
5) Region name is given in one column of 'Sampling Point data' and we want get this name automatically after the cluster id is entered. This case we need to create a text attribute which is tagged as 'Calculated'.



   And the calculated value expression under section '*Collect Earth*' is as follow.

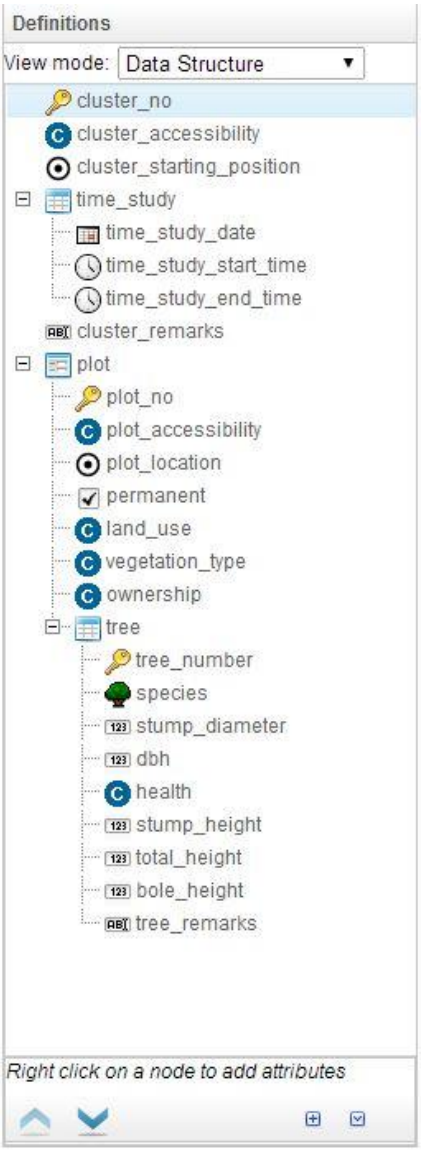


6) We want to get plot date automatically. Apply the expression ***idm:currentDate()***



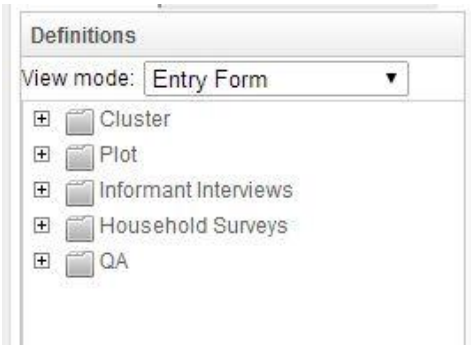
7) And in the previous image there is the attribute '*plot_start_time*' which we can get automatically. The expression is ***idm:currentTime()***

Once all entities and attributes have been entered and defined, the hierarchical structure of the survey will look similar to the image below.

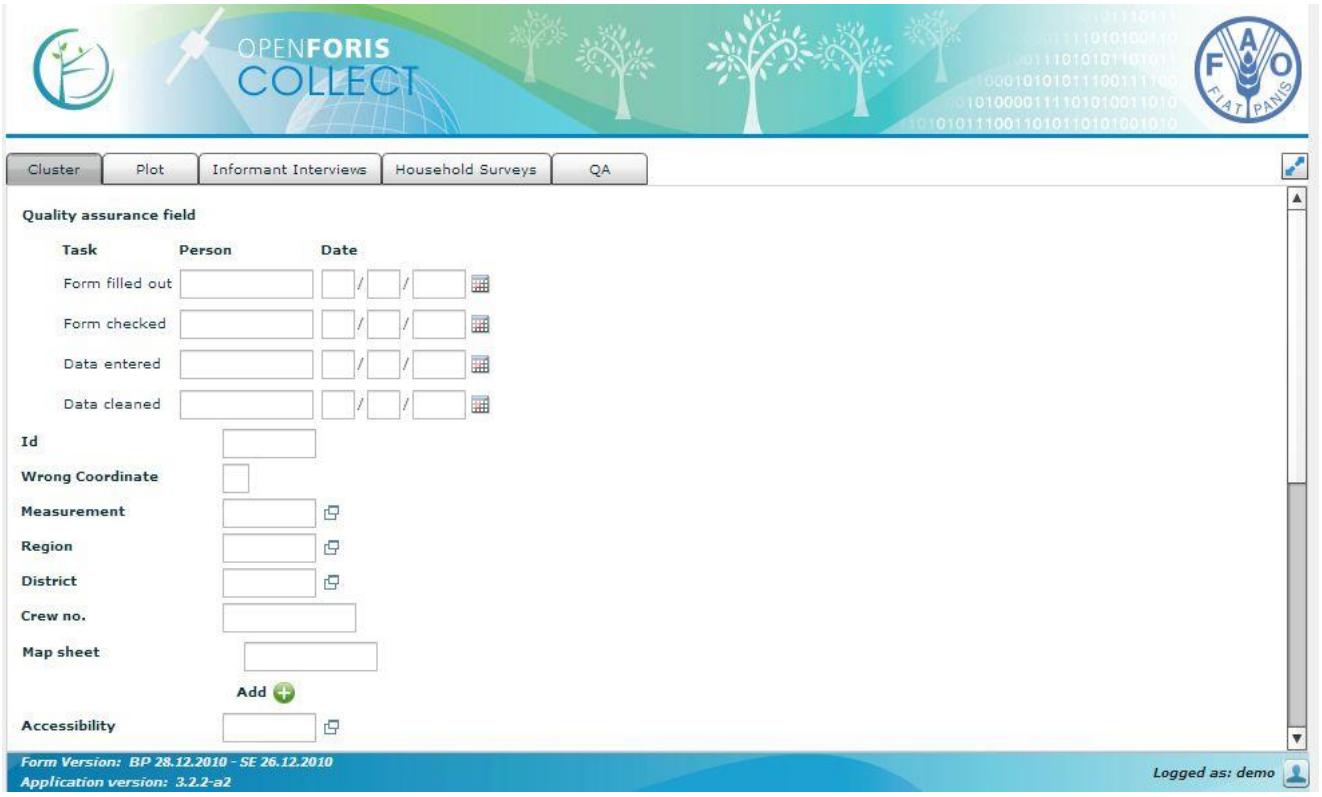


Data Structure vs. Entry Form

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 [+] and [-] 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.



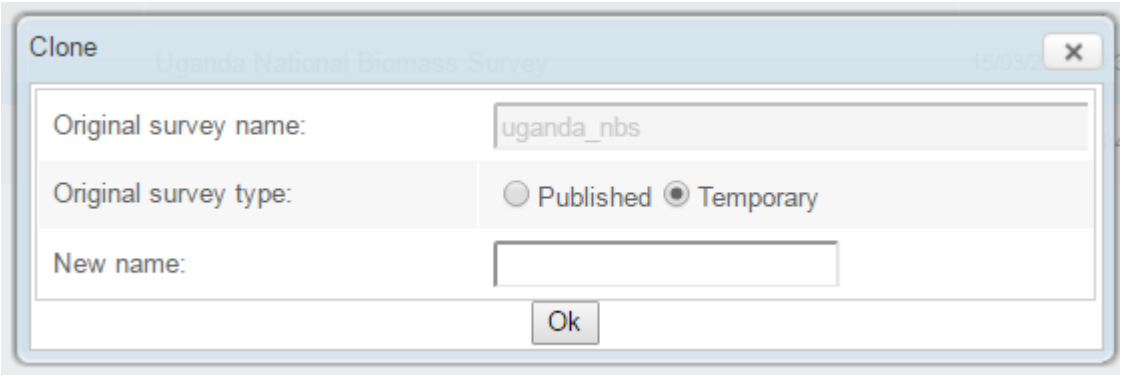
At any time it is possible to visualize the appearance of the data entry form by clicking on the **Preview** button.



5.7. ADVANCED FUNCTIONS

Select a survey from the list and advanced functions comes visible. These are as follows

1) **Clone:** Make a copy of an existing survey.



2) **Publish latest changes:** Only for published surveys. Includes any changes you have put into the survey since it was last published.

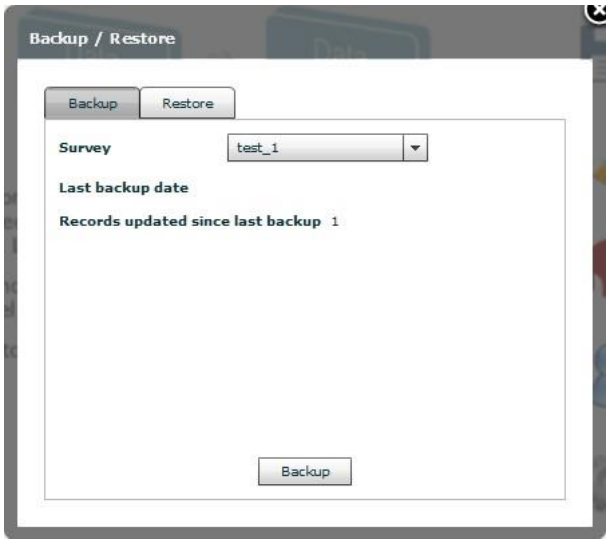
3) **Unpublish**: Only for published surveys. Allows you to edit the survey, but destroys all existing records for it. Useful if you accidentally publish a survey before it is ready.

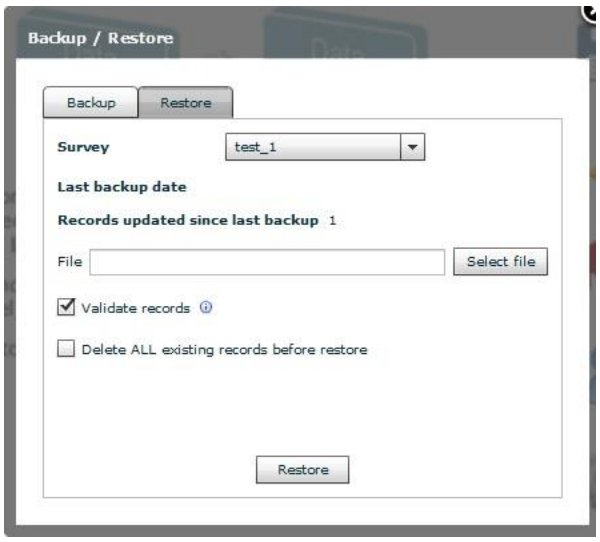
4) **Validate records**: This is automatically done when you publish a survey. Checks that everything is okay data structure-wise.

5) **Delete the selected survey**:

6. 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 “Backup”. A “.collect-backup” file is created. Should something bad happen to your survey, you can restore it in the Restore tab. Selecting “Validate records” updates all the error and warning counts in the record, 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”. It is a good idea to store all backups in a cloud storage or onto multiple separate hard drives.





7. DATA MANAGEMENT

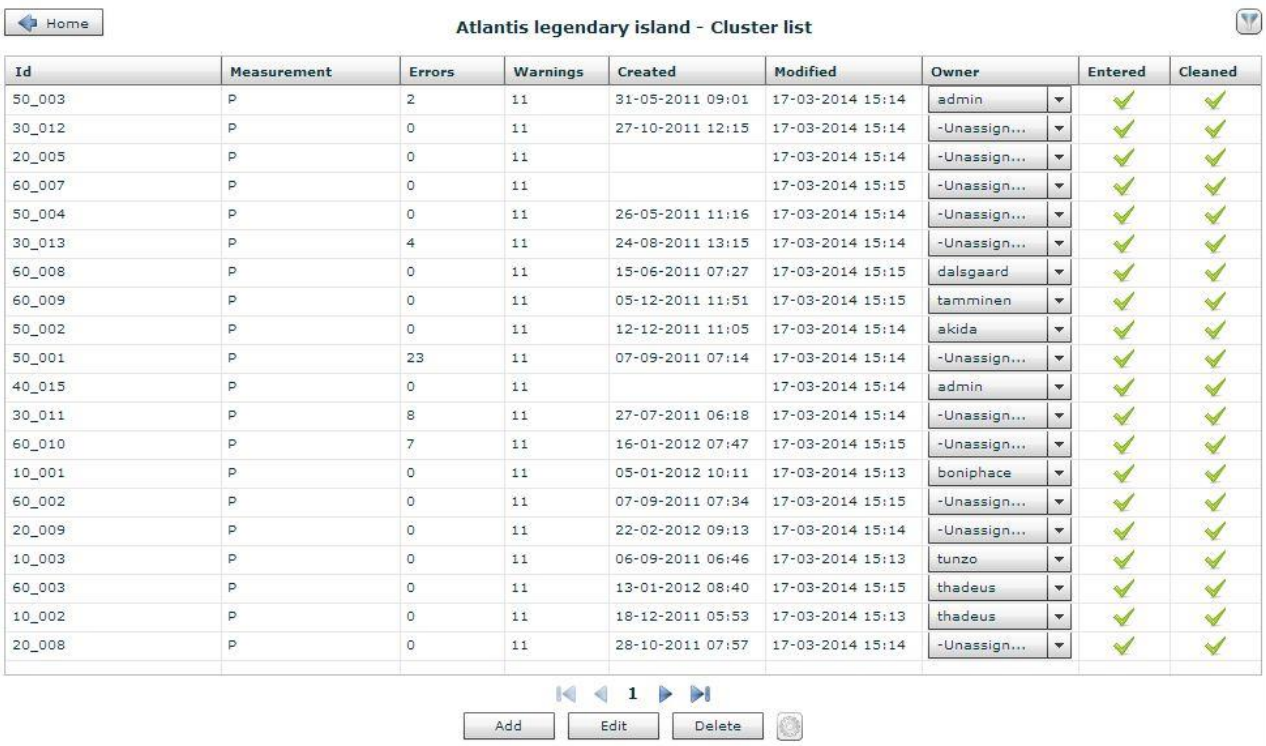
This section of Collect is the starting point for **Data Entry** and **Data Management.** In this section records can be entered, edited and managed. Once records are entered into Collect, this section presents a log of errors and warnings encountered during data entry as well as information on when 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.

Specifications on the **Open Foris Collect Workflow** (Data entry, Data cleansing and Data analysis) can be found at

http://www.openforis.org/OFwiki/index.php/Open_Foris_Collect_Workflow

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.



Id	Measurement	Errors	Warnings	Created	Modified	Owner	Entered	Cleaned
50_003	P	2	11	31-05-2011 09:01	17-03-2014 15:14	admin	✓	✓
30_012	P	0	11	27-10-2011 12:15	17-03-2014 15:14	-Unassign...	✓	✓
20_005	P	0	11		17-03-2014 15:14	-Unassign...	✓	✓
60_007	P	0	11		17-03-2014 15:15	-Unassign...	✓	✓
50_004	P	0	11	26-05-2011 11:16	17-03-2014 15:14	-Unassign...	✓	✓
30_013	P	4	11	24-08-2011 13:15	17-03-2014 15:14	-Unassign...	✓	✓
60_008	P	0	11	15-06-2011 07:27	17-03-2014 15:15	dalsgaard	✓	✓
60_009	P	0	11	05-12-2011 11:51	17-03-2014 15:15	tamminen	✓	✓
50_002	P	0	11	12-12-2011 11:05	17-03-2014 15:14	akida	✓	✓
50_001	P	23	11	07-09-2011 07:14	17-03-2014 15:14	-Unassign...	✓	✓
40_015	P	0	11		17-03-2014 15:14	admin	✓	✓
30_011	P	8	11	27-07-2011 06:18	17-03-2014 15:14	-Unassign...	✓	✓
60_010	P	7	11	16-01-2012 07:47	17-03-2014 15:15	-Unassign...	✓	✓
10_001	P	0	11	05-01-2012 10:11	17-03-2014 15:13	boniphace	✓	✓
60_002	P	0	11	07-09-2011 07:34	17-03-2014 15:15	-Unassign...	✓	✓
20_009	P	0	11	22-02-2012 09:13	17-03-2014 15:14	-Unassign...	✓	✓
10_003	P	0	11	06-09-2011 06:46	17-03-2014 15:13	tunzo	✓	✓
60_003	P	0	11	13-01-2012 08:40	17-03-2014 15:15	thadeus	✓	✓
10_002	P	0	11	18-12-2011 05:53	17-03-2014 15:13	thadeus	✓	✓
20_008	P	0	11	28-10-2011 07:57	17-03-2014 15:14	-Unassign...	✓	✓

Records can be *filtered* (click on filter icon in the top right corner) and *sorted* by columns. The advanced functions button (at the bottom) allows to *export* or *import* data as well as run a *Validation report*.

At the 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

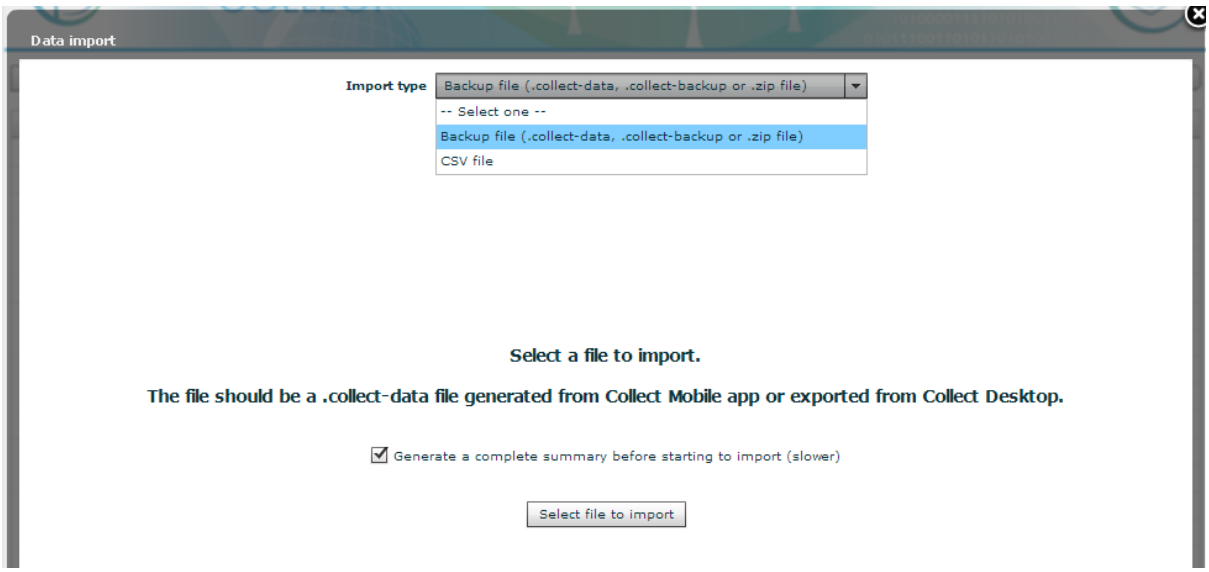
Soon after data collection in the field has started, the data manager will receive data files from the field crew. In order to avoid confusion, a structured archiving system should be set up. Regardless whether the data files will be stored on a local drive or through a cloud system, a structure with 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).

7.1 ADDING A RECORD

7.1.1 DATA UPLOAD (FROM COLLECT MOBILE)

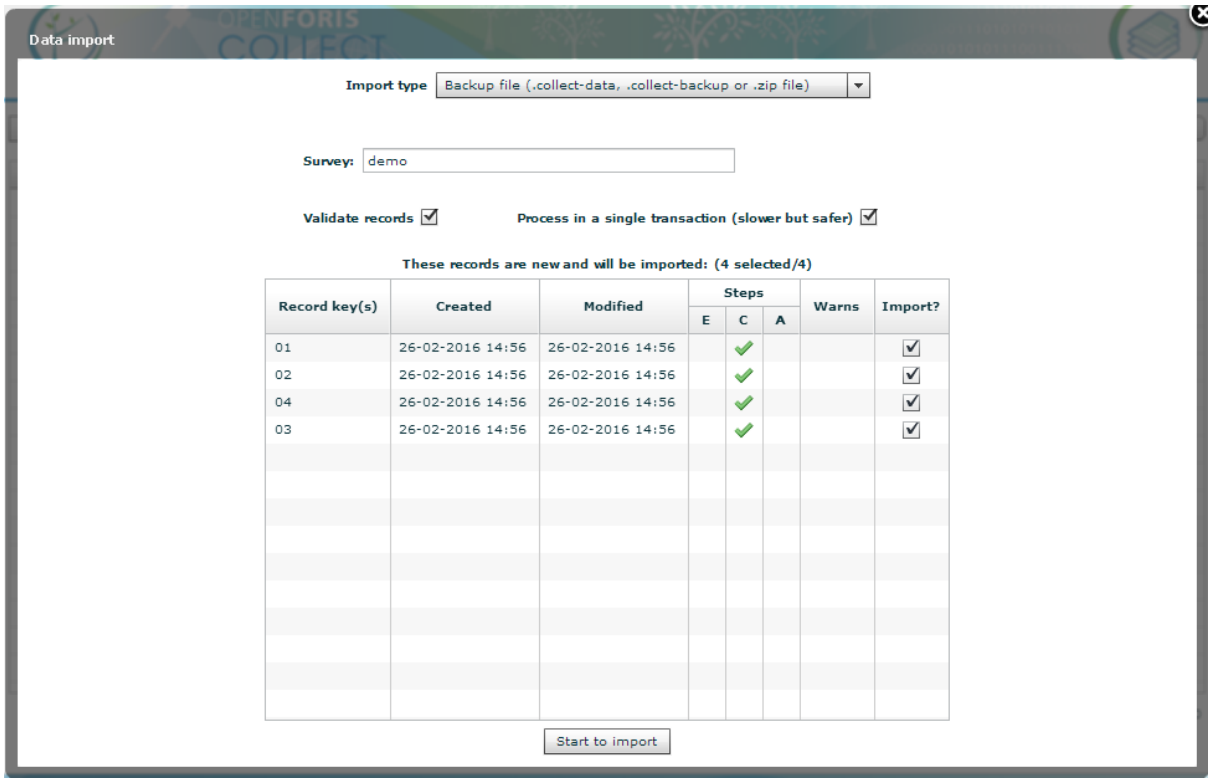
If data is recorded in the field with OF Collect Mobile, data files (exported from Collect Mobile) can be uploaded. At the bottom of the Data Management screen, click on the "wheel" and select Import data.

The user will be prompted to *Select a file to import*. Select the option: Backup file (.collect-data). The option to click the box to *Generate a complete summary* before starting to import is recommended. Locate the file in your archiving system and click *Import*.



Before proceeding with file import, Collect presents a summary of the records that will be imported. The screen, as in the image below, shows: the import file type, the survey name and the choice whether to run a Validate records during data import (suggested). The choice to Process in a single transaction is mainly applicable when working with OF Collect Earth and

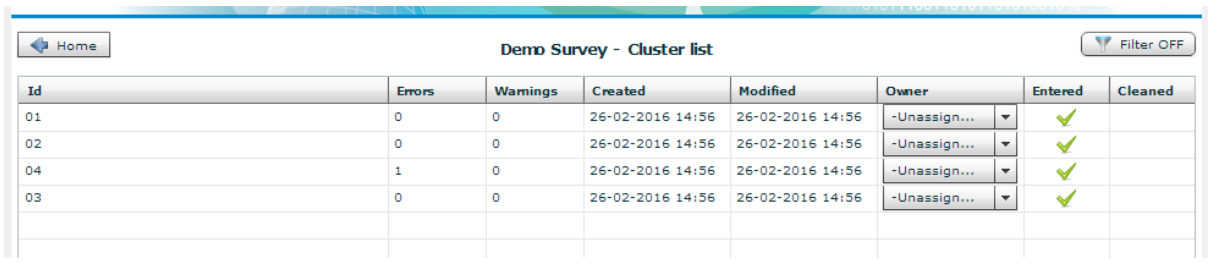
importing a large number of records (eg. more than 10,000). This feature prevents Collect from stopping data import when errors are encountered.



The summary table also shows the list of records that are ready for upload with an indication of the Record key (derived from the sampling unit list), the date in which the record was Created and Modified, the data flow Step and a tick box to select the records to be imported.

After reviewing the summary table and selecting the records to be imported, click *Start to import*. When the message *Data import completed* appears, close the summary window ('X' on the top right corner).

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.

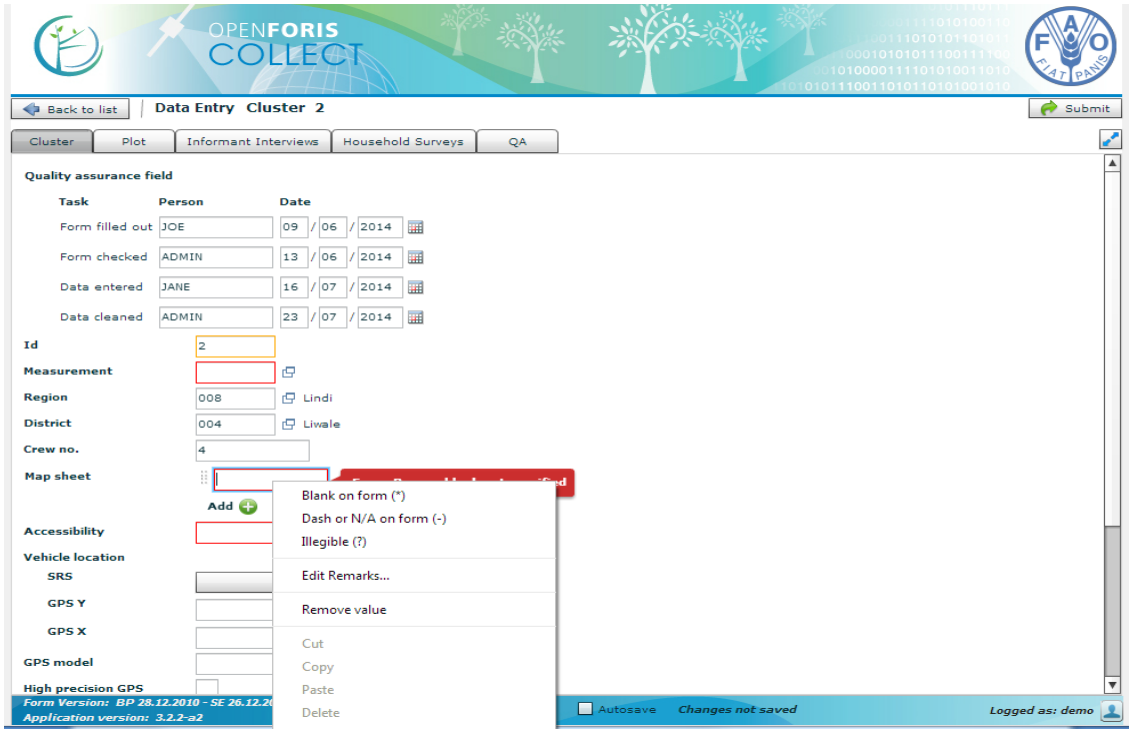


7.1.2 MANUAL DATA INPUT

To add data manually, click on Add to open the data entry form as designed in the Survey Designer. The user can start to enter data moving from field to field using 'Tab'. The behavior of each record field depends on its parameters as outlined in the Open Foris Collect Data Validation States at

http://www.openforis.org/OFwiki/index.php/Open_Foris_Collect_Data_Validation_States

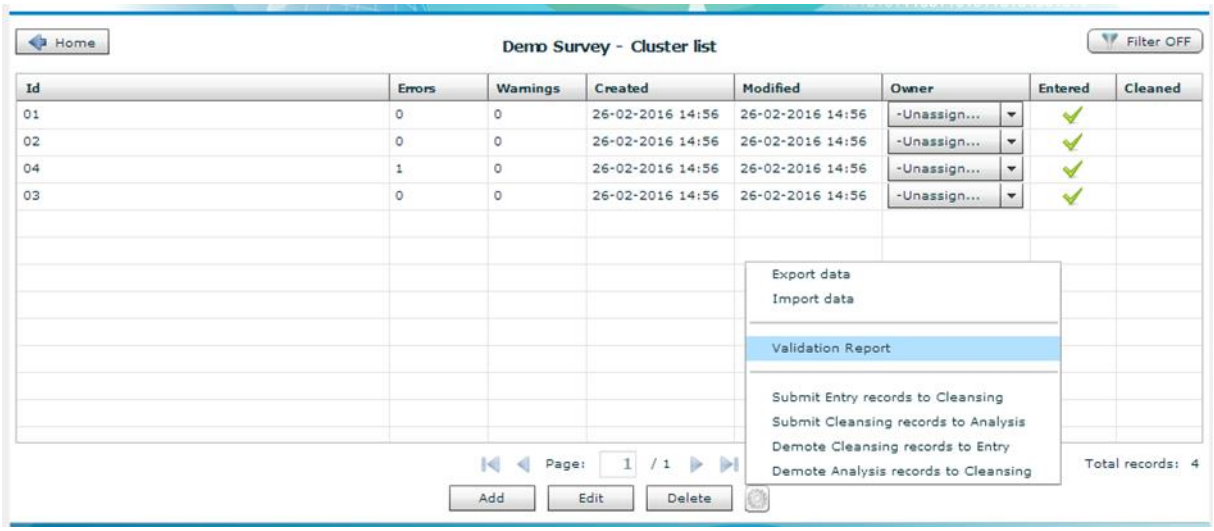
As shown in the image below, Errors will be highlighted in red, warnings in yellow. If no value was present in the paper form, the data entry user can leave the field blank and specify the reason (by right-click on the field and choosing one of the options).



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.

7.2 VALIDATION REPORT

At his 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 “wheel” and select *Validation Report*. The result will be a .csv file with details on the error(s) present in the records.

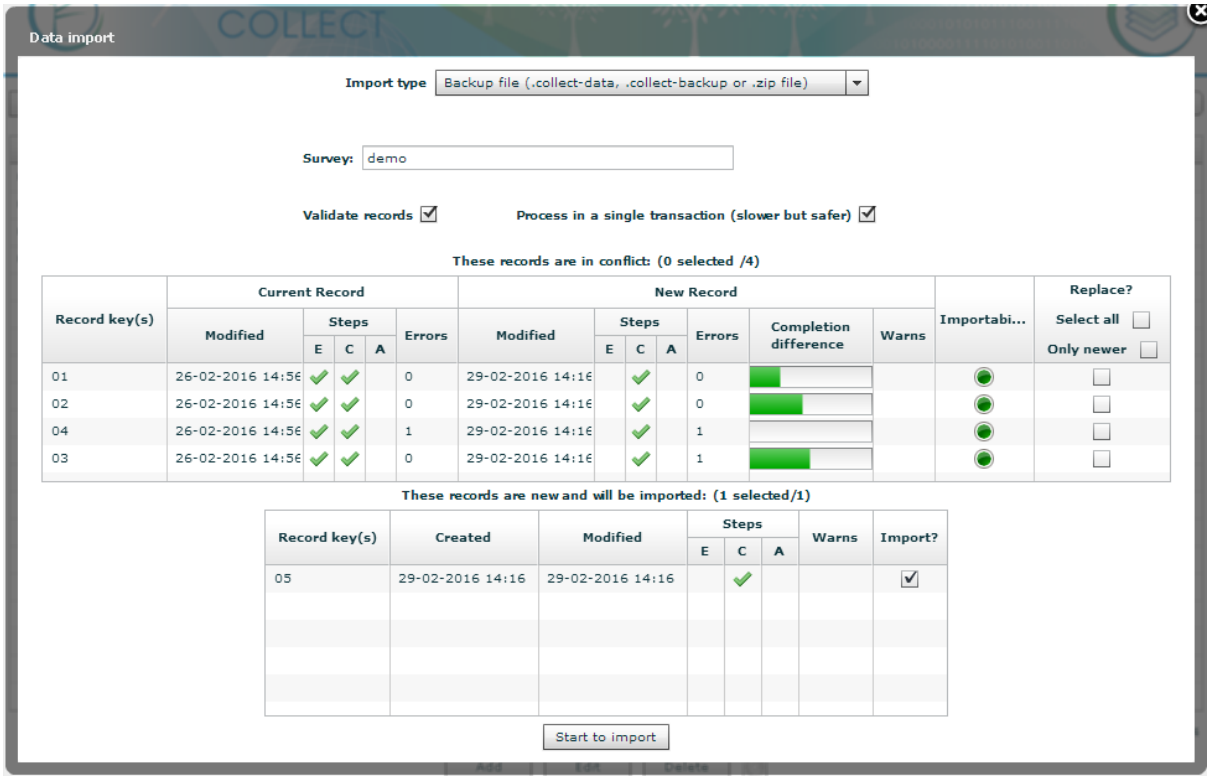


The process for correcting mistakes should be agreed with the Data manager who may have to contact the field crew leader responsible for the specific record (SU) and ask for clarifications. In some cases the field crew may have to go back to the field to re-collect the wrong information. The correct information can then be sent with the successive data export from the field or the data manager may manually correct the value from the data entry interface.

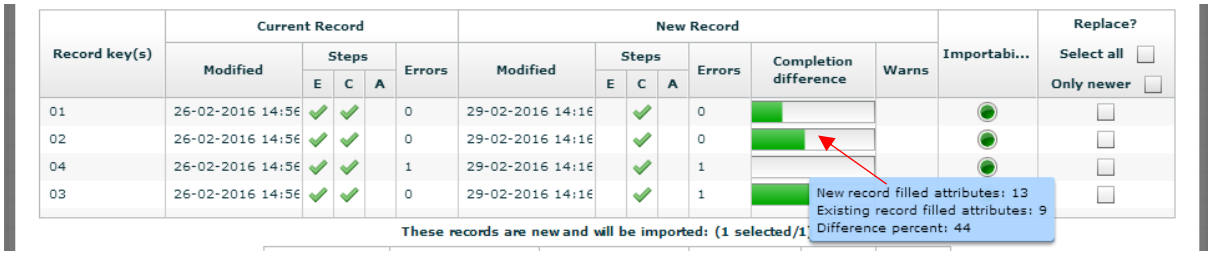
7.3 ADDING MORE RECORDS

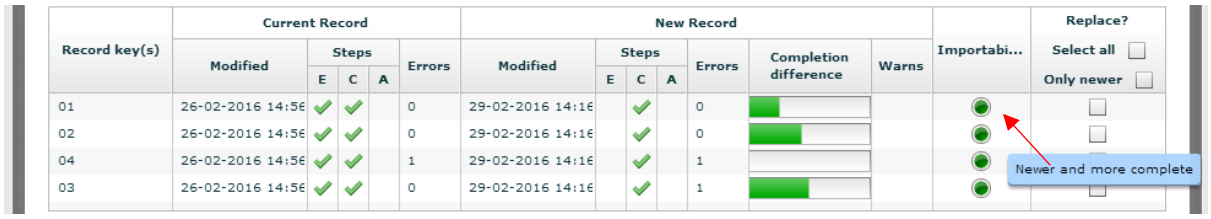
As field data collection continues, the data manager will receive additional data files. The new data files will either contain new records or newer versions of existing records. In the latter case, for example, if data related to sample unit #1 was already sent but field work was not completed, the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

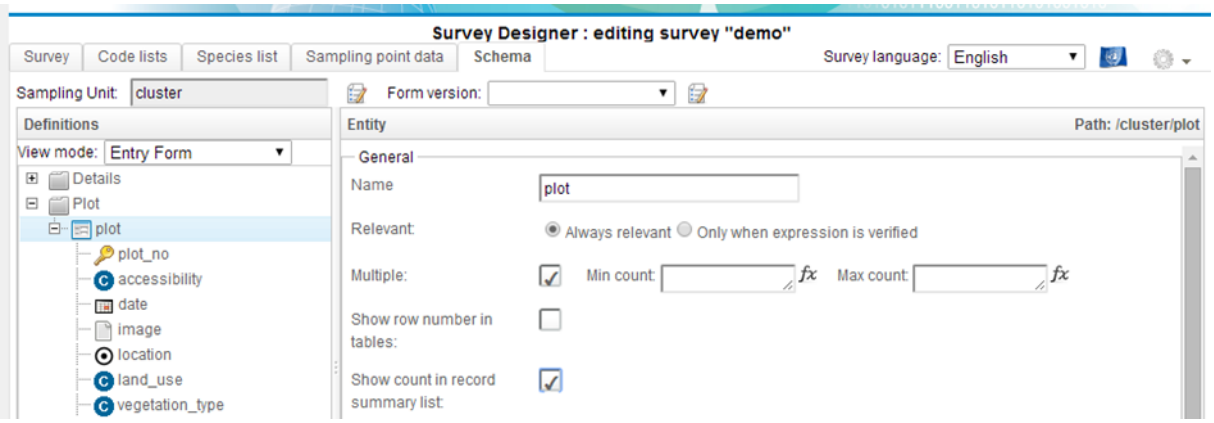
individually. The bottom table, as in the first data upload simply lists records not currently present in the database, thus ready for upload by default.

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.

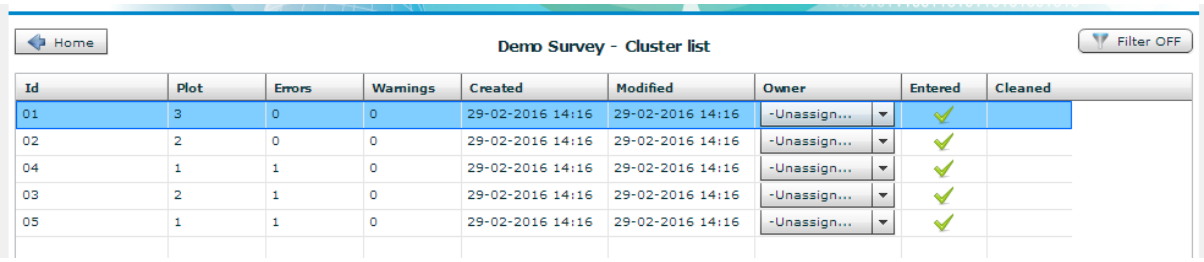
7.4 RECORD COMPLETENESS

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.



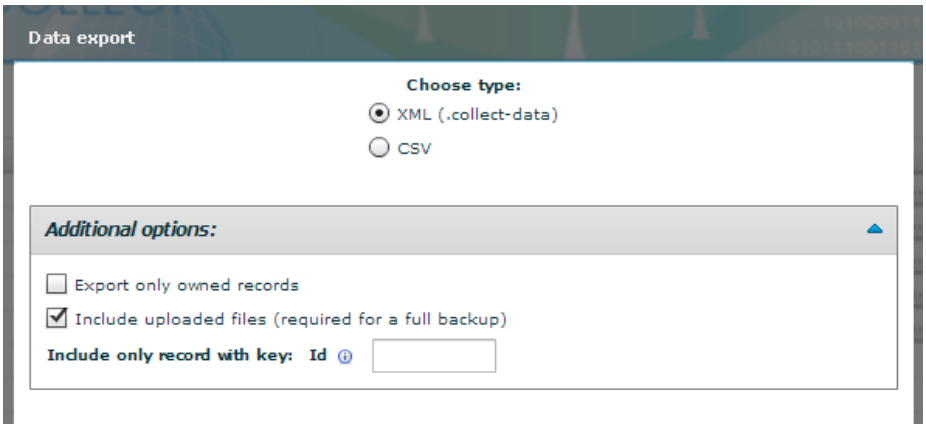
By doing so, in the log of records in Data Management, a new column showing count for the selected entity will be added. See below.



Id	Plot	Errors	Warnings	Created	Modified	Owner	Entered	Cleaned
01	3	0	0	29-02-2016 14:16	29-02-2016 14:16	-Unassign...	✓	
02	2	0	0	29-02-2016 14:16	29-02-2016 14:16	-Unassign...	✓	
04	1	1	0	29-02-2016 14:16	29-02-2016 14:16	-Unassign...	✓	
03	2	1	0	29-02-2016 14:16	29-02-2016 14:16	-Unassign...	✓	
05	1	1	0	29-02-2016 14:16	29-02-2016 14:16	-Unassign...	✓	

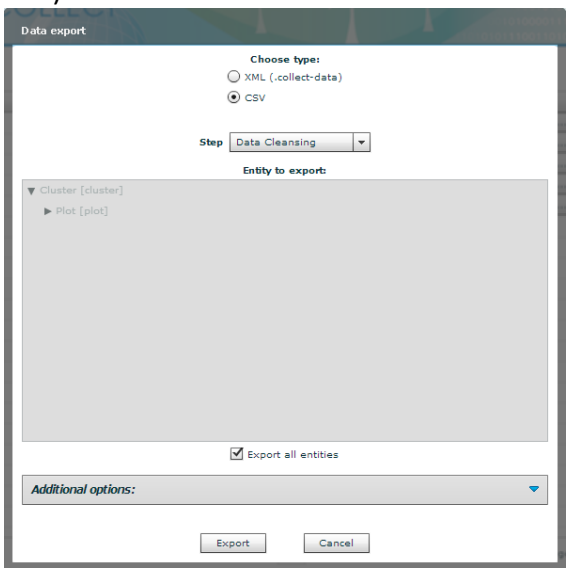
7.5 DATA EXPORT

Another way to have a closer look at the data is to export it. From the Data Management window, click on the “wheel” and select *Export data*. Selecting XML (.collect-data) allows to share 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).

Selecting CSV and clicking “Export all entities” will create CSV file(s) for each of the entity in the survey. Select a data *Step*. Click *Export.* When the message “Data export complete” appears, click *Download.* The result is a Zip file containing a CSV file for each entity (eg. cluster.csv, plot.csv, tree.csv)

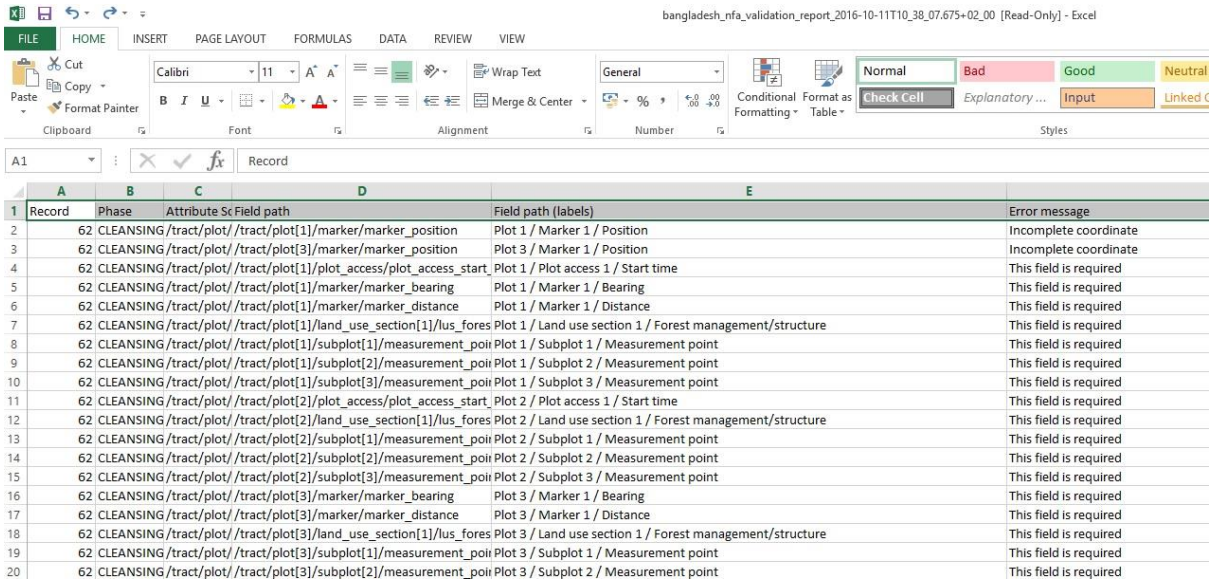


8. DATA CLEANSING

The Data Cleansing button on the Home page of *OF Collect* 

contains features that are most applicable when using *OF Collect Earth*. It allows to run queries on the database and perform 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.



	A	B	C	D	E	
1	Record	Phase	Attribute Sc	Field path	Field path (labels)	Error message
2	62	CLEANSING	/tract/plot/	/tract/plot[1]/marker/marker_position	Plot 1 / Marker 1 / Position	Incomplete coordinate
3	62	CLEANSING	/tract/plot/	/tract/plot[3]/marker/marker_position	Plot 3 / Marker 1 / Position	Incomplete coordinate
4	62	CLEANSING	/tract/plot/	/tract/plot[1]/plot_access/plot_access_start_	Plot 1 / Plot access 1 / Start time	This field is required
5	62	CLEANSING	/tract/plot/	/tract/plot[1]/marker/marker_bearing	Plot 1 / Marker 1 / Bearing	This field is required
6	62	CLEANSING	/tract/plot/	/tract/plot[1]/marker/marker_distance	Plot 1 / Marker 1 / Distance	This field is required
7	62	CLEANSING	/tract/plot/	/tract/plot[1]/land_use_section[1]/lus_fores	Plot 1 / Land use section 1 / Forest management/structure	This field is required
8	62	CLEANSING	/tract/plot/	/tract/plot[1]/subplot[1]/measurement_poi	Plot 1 / Subplot 1 / Measurement point	This field is required
9	62	CLEANSING	/tract/plot/	/tract/plot[1]/subplot[2]/measurement_poi	Plot 1 / Subplot 2 / Measurement point	This field is required
10	62	CLEANSING	/tract/plot/	/tract/plot[1]/subplot[3]/measurement_poi	Plot 1 / Subplot 3 / Measurement point	This field is required
11	62	CLEANSING	/tract/plot/	/tract/plot[2]/plot_access/plot_access_start_	Plot 2 / Plot access 1 / Start time	This field is required
12	62	CLEANSING	/tract/plot/	/tract/plot[2]/land_use_section[1]/lus_fores	Plot 2 / Land use section 1 / Forest management/structure	This field is required
13	62	CLEANSING	/tract/plot/	/tract/plot[2]/subplot[1]/measurement_poi	Plot 2 / Subplot 1 / Measurement point	This field is required
14	62	CLEANSING	/tract/plot/	/tract/plot[2]/subplot[2]/measurement_poi	Plot 2 / Subplot 2 / Measurement point	This field is required
15	62	CLEANSING	/tract/plot/	/tract/plot[2]/subplot[3]/measurement_poi	Plot 2 / Subplot 3 / Measurement point	This field is required
16	62	CLEANSING	/tract/plot/	/tract/plot[3]/marker/marker_bearing	Plot 3 / Marker 1 / Bearing	This field is required
17	62	CLEANSING	/tract/plot/	/tract/plot[3]/marker/marker_distance	Plot 3 / Marker 1 / Distance	This field is required
18	62	CLEANSING	/tract/plot/	/tract/plot[3]/land_use_section[1]/lus_fores	Plot 3 / Land use section 1 / Forest management/structure	This field is required
19	62	CLEANSING	/tract/plot/	/tract/plot[3]/subplot[1]/measurement_poi	Plot 3 / Subplot 1 / Measurement point	This field is required
20	62	CLEANSING	/tract/plot/	/tract/plot[3]/subplot[2]/measurement_poi	Plot 3 / Subplot 2 / Measurement point	This field is required

As shown before, running a Validation report allows to work 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.

8.1 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.

Concept	**Explanation**
Data Query Type	Identifies what kind of queries will be available, useful when grouping queries into categories (typos, incomplete records, changed code).

Data Query	Defines a query on the data, a criteria to find values among the collected records.
Data Query Group	A group of queries that can be used to run them in bulk to save time.
Data Report	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.
Data Cleansing Step	Changes the values of attributes to another value according to conditions of a query (constant or expression)
Data Cleansing Chain	Groups a set of Cleansing Steps for executing and gives a summary of records involved in the cleansing process.

8.2 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
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

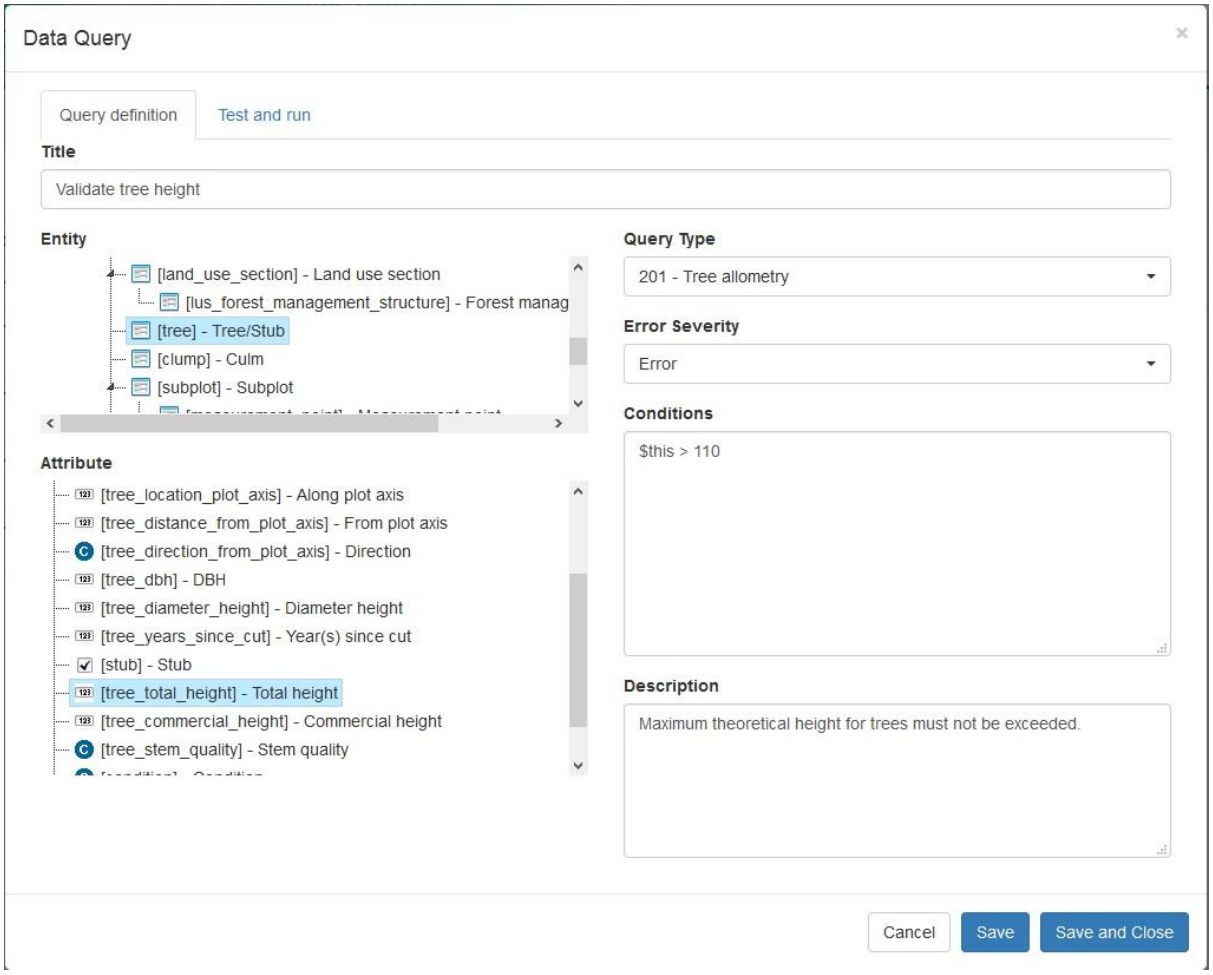
8.2.1 DATA BACKUP

See section six 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.

8.2.2 DEFINE DATA QUERY TYPES

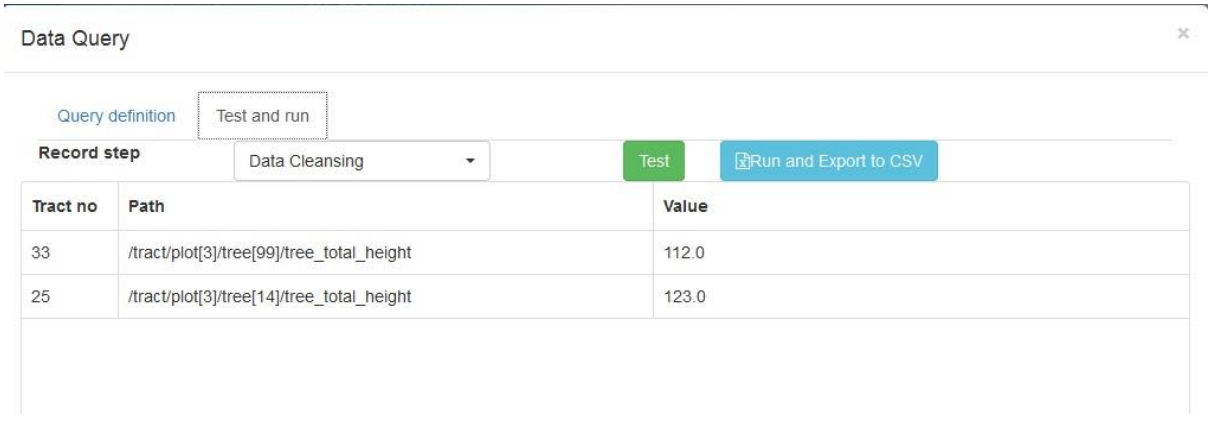
Select the green “New” button to add a Data Query type. Remember that these should be general **types** of mistakes, not specific things to correct. Specify the name and code for this Query Type in the pop up window. Note that the code can be either letters or numbers, but it has to be unique for that 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.

8.2.3 Define Data Queries

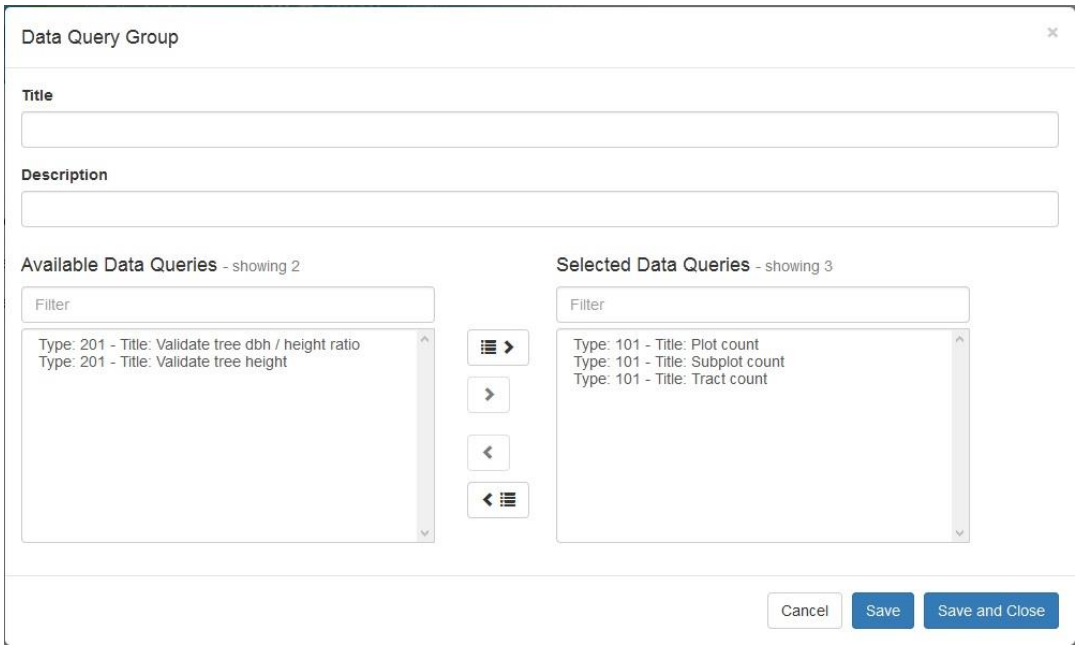


Data Queries are defined as XPath expressions, written into the "Conditions" box. Start by selecting the node you want to target from the Entity and Attribute lists. Then select what type of query this should be (from the ones created in Query Types) and select if this error is a warning or an error. For help in programming the condition, see Appendix 3.

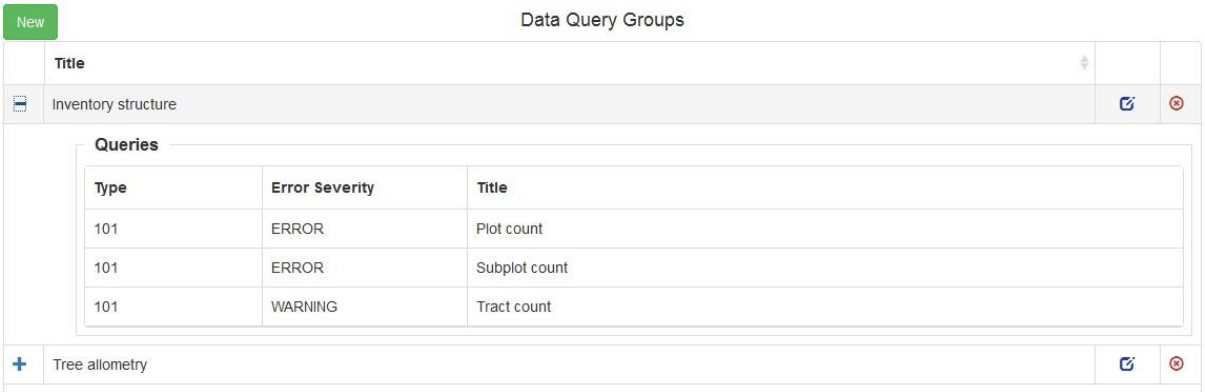
To see if a Query works, go to the "Test and Run" tab. Select the appropriate record step (the phase of data management at which your data is now, usually Data Cleansing now) and click "Run". A summary of successfully located records appears on the window.



8.2.4 Group Queries into Data Query Groups



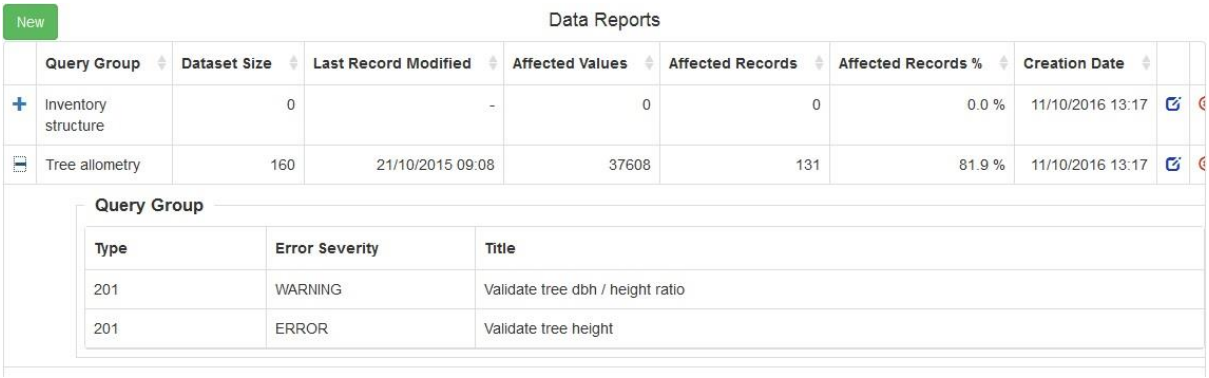
For efficiency, Data Queries will be run as groups. To create a group, click on the green "New" button and a pop up will appear. You form a group by selecting the relevant Queries from the left box and clicking on the arrows in the middle. To move all Queries at once, click on the uppermost or downmost arrow. You can select specific Queries by holding down the CTRL-key and clicking them.



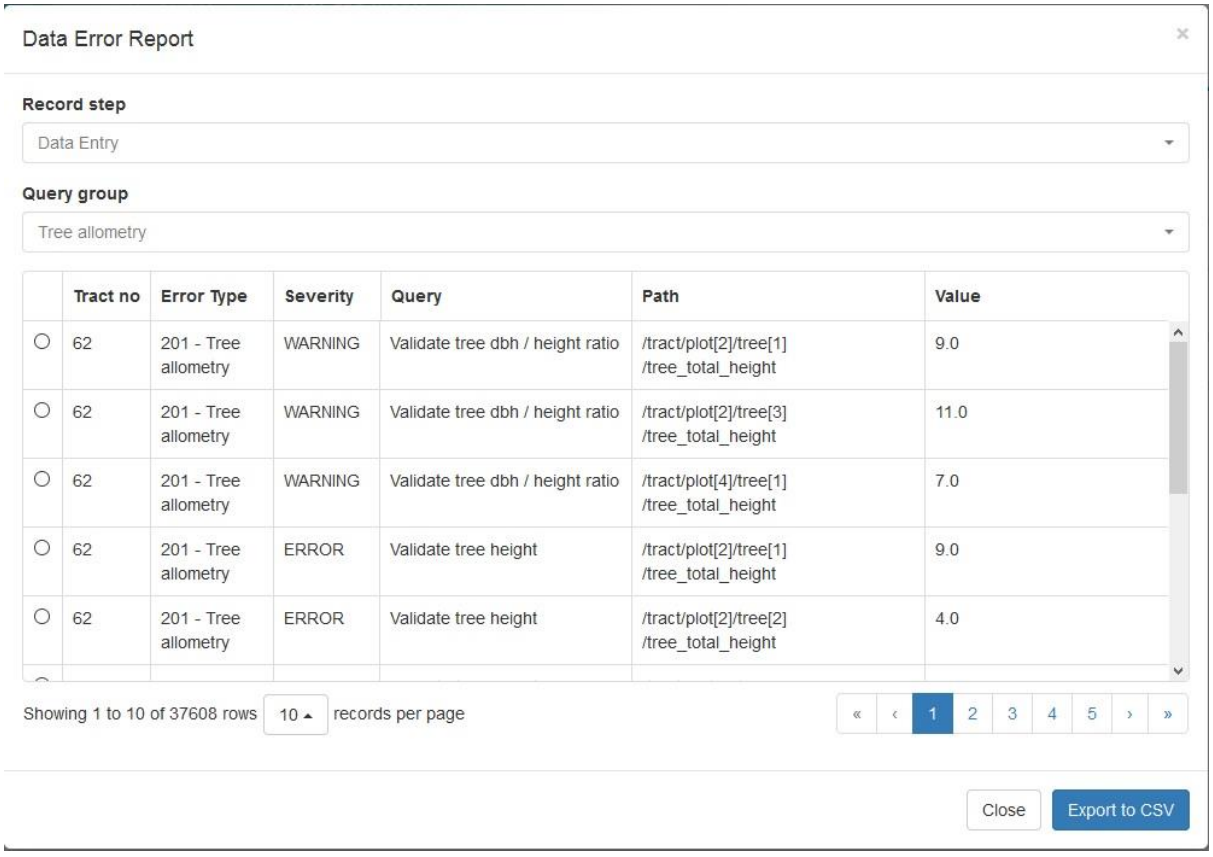
You can view the included Queries by clicking on the plus sign next to the Query Group's name.

8.2.5 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.



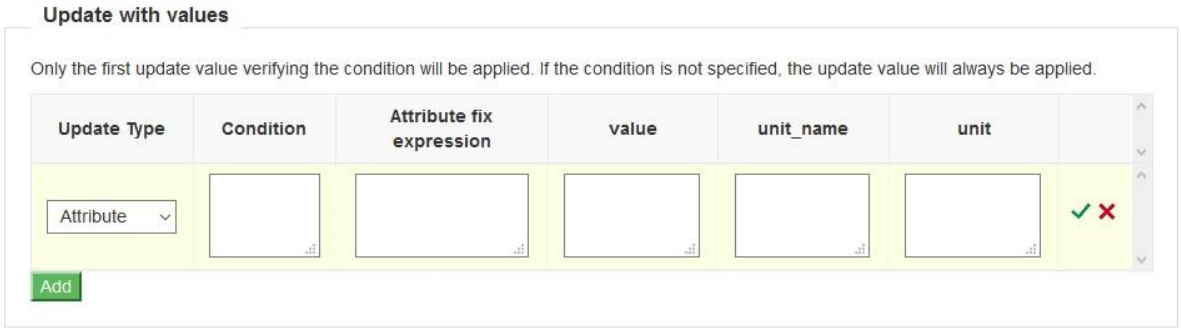
You can view the results in browser, or export them as a CSV. The report is stored in the system and is used for checking that the data cleansing operation works as expected. After running your Data Cleansing Chains, produce a new Data Report and check if the records have been fixed.

8.2.6 Defining Cleansing Steps for Queries

Cleansing Steps are created just like Queries, only that now you assign Queries to cleansing

steps. A cleansing steps searches for records based on its Query, but it also changes or deletes the record according to its Cleansing type:

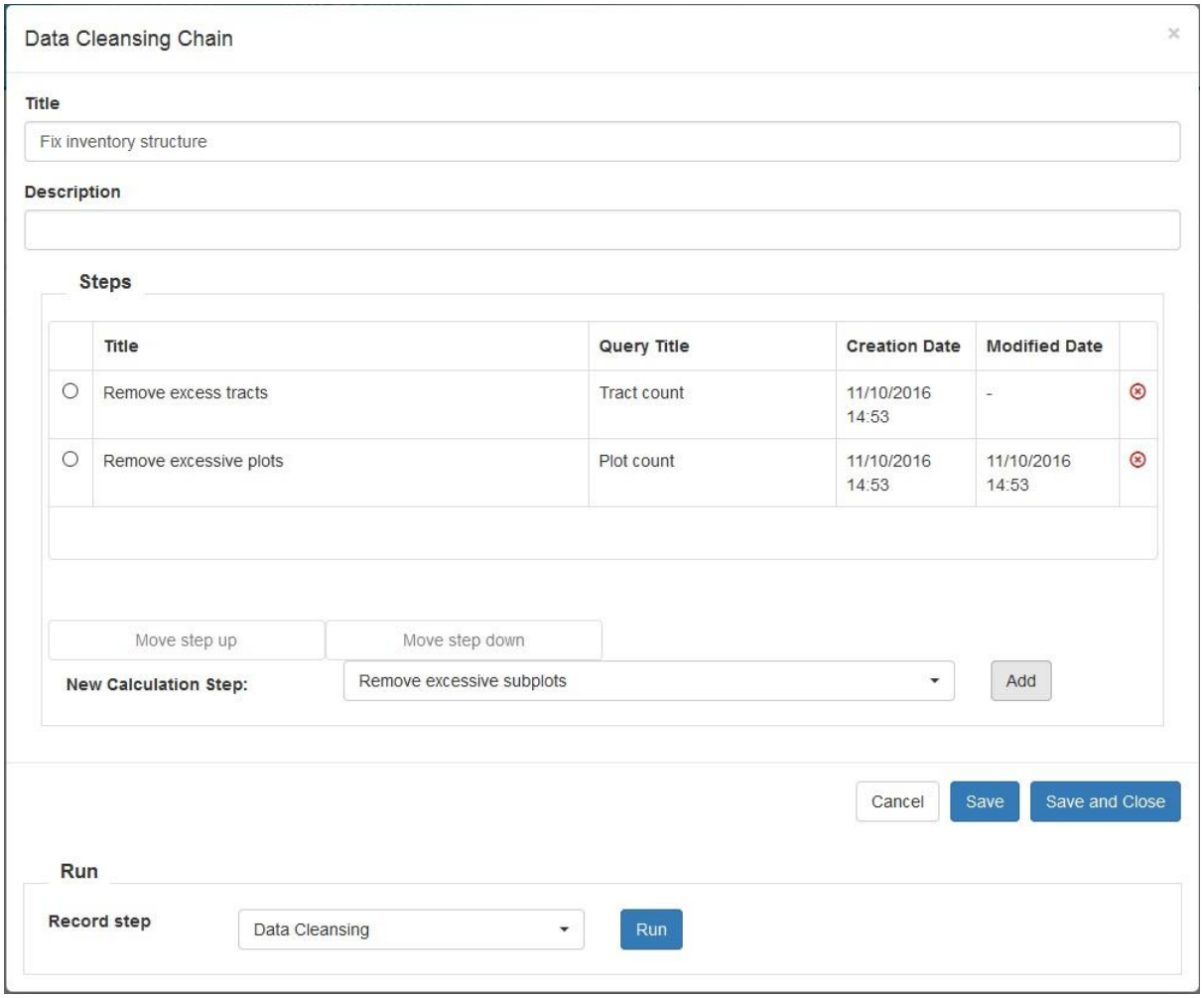
Update attribute: Changes the value of any attribute that matches the condition set by the Query being used. The replacement can be a single value or



Delete entity: Deletes any **entities** that match the condition set by the Query being used. Useful for removing large batches of irrelevant data.

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

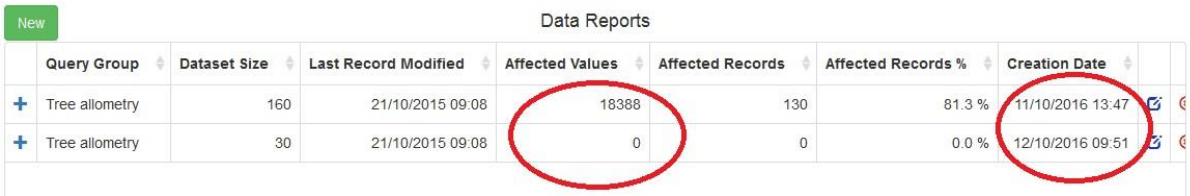
8.2.7 Defining a Data Cleansing Chain



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.



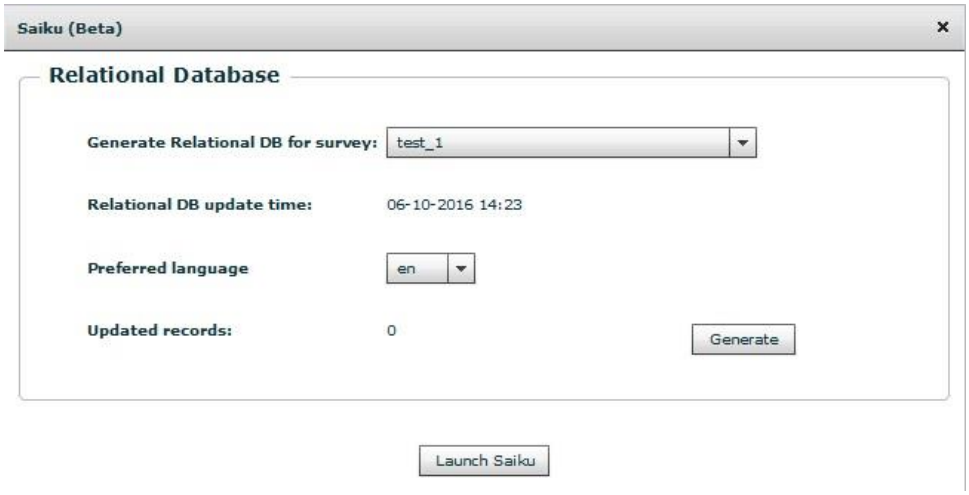
As you can see, the Data Cleansing chain has removed all the faulty records. **Note that the report only shows the kinds of records that are relevant to the conditions of the Queries in them.** In the picture above, for example, you can only see the count of records that are

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.

9. SAIKU

Saiku is a web-based open source software that facilitates data visualization and data querying. To lauch 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.

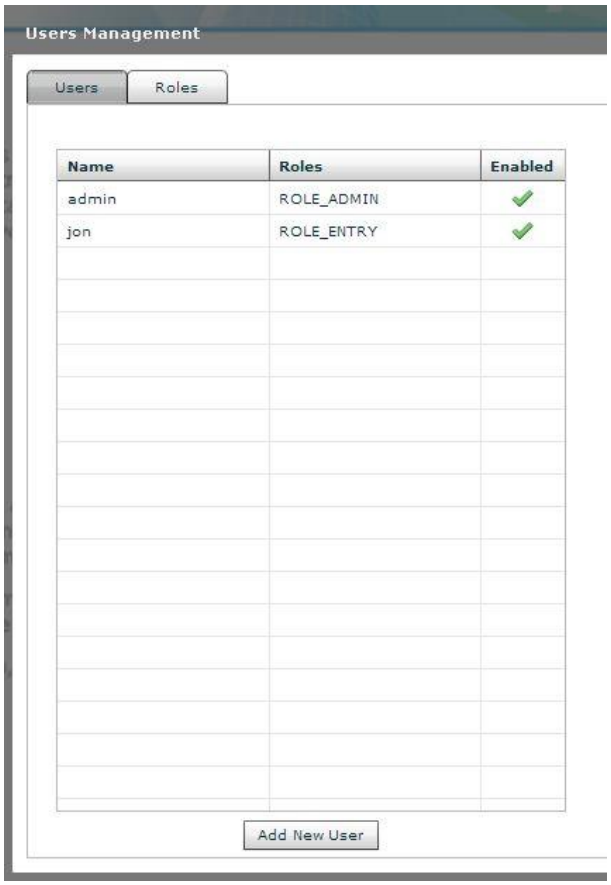
For detailed instructions on how to use Saiku, refer to the Collect Earth documentation.

http://www.openforis.org/tools/collect-earth/tutorials/saiku.html

10. USERS ACCOUNTS

Users accounts 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.

From the home panel, clicking on **Users accounts** will open a window with a list of current users: Name and Role.



Clicking on a user will show User's details.

To add a new user click on **Add New User** at the bottom of the window and fill in user's 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:

- **Data Entry**: Lowest level of permission. Data Entry users have access only to data entry phase, they are allowed to enter new records, edit own records, submit them for cleansing and export records.
- **Data Cleansing**: Same as Data entry + permission to edit records in cleansing phase and submit them to data analysis.
- **Data Analysis**: All previous rights + the ability to unlock records and re-submit them to data cleansing phase.
- **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.

Users-Roles management can also be done from the Roles tab.

11. SETTINGS

From Collect Home, clicking on Settings allows the user to modify *Storage paths* customizations and *Layout images* customization.

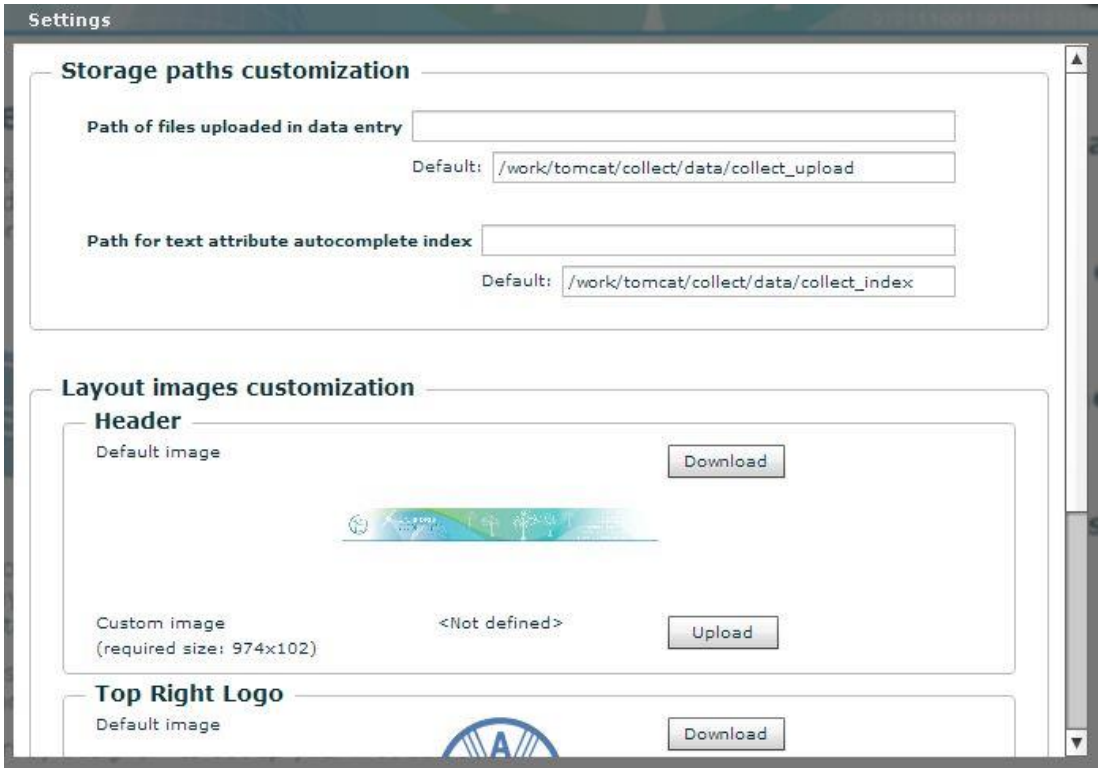
Storage paths customization:

- **Path of files uploaded in data entry** - this allows modifying the default path of the folder where files uploaded during data entry will be saved. For example in the case of attributes of type: file.
- **Path for the text attribute autocomplete index** - Every value entered in text fields are stored and they form the index of values available for autocomplete. Here it is possible to change the default path location.

Note: both customizations are mostly applicable when Collect is installed on a server in a multi-user environment. Unless needed, it is suggested not to modify the default paths.

Layout images customization:

- This allows the users to modify the graphical appearance of Collect by modifying the Header, the top-right logo and the footer. Users can download and upload their own images paying attention to the required sizes (in pixels).



ANNEX 1. POSTGRESQL WITH COLLECT

Note: see also

http://www.openforis.org/OFwiki/index.php/Open_Foris_Collect_Installation#Install_PostgreSQL_as_Database

Install PostgreSQL (v. 9.4 or newer), so that database password is ***postgres*** (as it is recommended by the installer). 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 install StackBuilder.

After installing Collect, change database connection settings in this file:
C:\opt\openforis\collect\tomcat\conf\Catalina\localhost\Collect.xml

Use for example NotePad++ as a text editor (or any other suitable XML editor).

Comment SQLite, and activate PostgreSQL connection, as follows.

```
<?xml version="1.0" encoding="UTF-8"?>
<Context path="/collect" reloadable="false">

    <!-- Parameters -->
    <Parameter name="collect.simple_editor" value="false" /> <!-- Simplifies Survey Editor UI -->

    <!-- SQLite database stored in CATALINA_HOME/data/collect.db file -->
    <!--Resource
       name="jdbc/collectDs"
       auth="Container"
       type="javax.sql.DataSource"
       factory="org.apache.commons.dbcp.BasicDataSourceFactory"
       driverClassName="org.sqlite.JDBC"
       url="jdbc:sqlite:${catalina.home}/data/collect.db">
     </Resource -->

    <!-- PostgreSQL database configuration -->
    <Resource
       name="jdbc/collectDs"
       auth="Container"
       type="javax.sql.DataSource"
       factory="org.apache.commons.dbcp.BasicDataSourceFactory"
       driverClassName="org.postgresql.Driver"
       url="jdbc:postgresql://localhost:5432/collect"
       username="collect"
       password="collect123"
       initialSize="5"
       maxActive="20"
       maxIdle="5">
     </Resource>


</Context>
```

Then start Collect. If Tomcat gives an error in the log window and complains about permissions user rights, check using pgAdmin that PostgreSQL database password (*postgres*) and localhost number (*5432*) are correct. If this is correct, refresh database view in pgAdmin

and check whether there is database ‘collect’ created. If this is not there, you can do as follows:

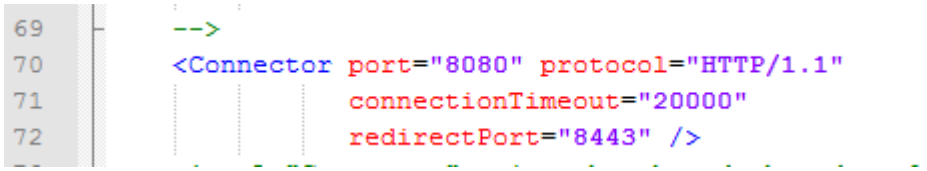
1) Stop Collect (and Tomcat)
2) Using pgAdmin, create username ‘collect’, password: ‘collect123’, with super-user rights
3) Create database ‘collect’, with owner ‘collect’
4) Under this database, create schema ‘collect’, with owner ‘collect’
5) Start Collect
6) After server starting (see Tomcat window), 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.

ANNEX 2. SERVER CONNECTION SETTINGS

See first
http://www.openforis.org/OFwiki/index.php/Open_Foris_Collect_Installation#Collect_Server_Setup

The default port for Tomcat is 8080. The server connection settings are located in the following file:

C:\opt\openforis\collect\tomcat\conf\server.xml



ANNEX 3. EXAMPLES OF CODE IN OPEN FORIS COLLECT

General hierarchical structures (either cluster or plot-based)

Comparisson 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 distance must be positive**
0 **<=** 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**
Message **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**
Message **Impossibly tall tree**
total_height **<** **150**

Severity **Error**
Message **height must be positive**
0 **<=** total_height

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

Severity **Error**
Message **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**
Message **You are more than 50 m from plot location**
Destination point **idm:samplingPointCoordinate(parent()/plot_id)**
Max distance (m) **50**

Severity **Warning**
Message **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 fot the variable) in optional variables from sample design file (i.e., province as a column after design point coordinates in the sampling_design imported file):

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

Requirednes of variable if expression is satisfied (i.e., tree_height only required every three third tree):

Example attribute (text type): **Path: /plot/tree/tree_height**
Required only when expression is verified: **tree_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	**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	**idm:samplingPointCoordinate(parent()/parent()/cluster_no, parent()/plot_no)**
Max distance (m)	**20**

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

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

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

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

Example attribute (number type): **Path: /cluster/plot/tree_count**
Calculated value expressions: Expression: **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: **parent()/plot/details/accessibility=0**

Example attribute (coordinates type): **Path: /cluster/plot/details/position**
Relevant only when expression is verified: **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: **details/accessibility=0**