Open Foris Collect

Handbook

9 February 2021

http://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

3.1 PREREQUISITES

1. Web browser: **Google Chrome** is recommended.

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**, with 2 items: **Open Foris Collect** and **Update 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*.

3.3.4 Setup Collect in a server for a multi-user environment

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.

If you want to use Collect in a multi-user environment, you can think about installing it in a server remotely accessible (or in the same local network) from different client computers.
Collect is just a web application that runs in a Application Server (e.g. Apache Tomcat).
You need to follow these steps:

- install Apache Tomcat 7+ application server (Apace Tomcat 9 is suggested) (download Tomcat):
- install Java Runtime Environment (JRE) 8+ (download Java)
- install PostgreSQL 8+ (download PostgreSQL)
- create an empty database called "collect"
- download:
    - the latest version of the PostgreSQL JDBC driver
    - the latest version of the SQLite JDBC driver
    - the latest version of the Apache Commons DBCP Component
    - put them into tomcat/libs folder
- download the latest version of the Collect war file
- put the war file into the tomcat/webapps folder and rename it into collect.war
- download the sample collect.xml file
- modify the collect.xml file with the correct database connection parameters
- put it inside tomcat/conf/Catalina/localhost

If you followed correctly all these steps, you can startup Tomcat and Collect will be available (usually) under http://localhost:8080/collect (if you keep 8080 as default port in Tomcat and collect.war as name of the downloaded Collect war file).

If Collect doesn't start up, have a look to the log files in the tomcat/logs folder (usually there can be problems in the configuration of the connection with the database).

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

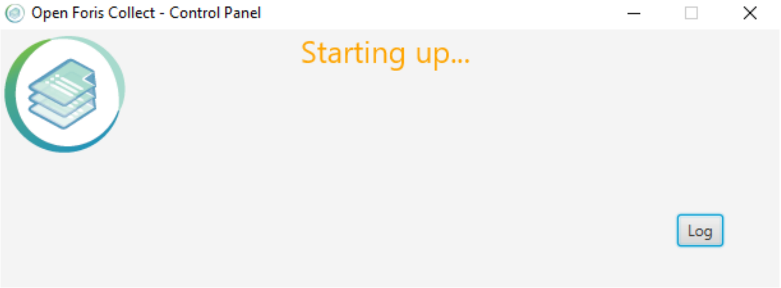
- Double click on the ***Launch Open Foris Collect*** icon in the desktop
- The ***Collect Control Panel*** will appear. See Using Collect Control Panel

4.3 Mac OSx systems

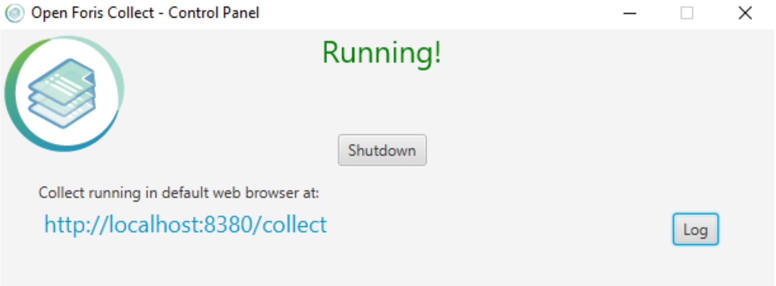
- Double click on the ***Launch Open Foris Collect*** icon in the desktop
- The ***Collect Control Panel*** will appear. See Using Collect Control Panel

5 Using Collect Control Panel

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



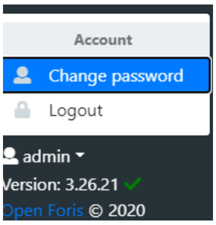
Please wait until the "**Starting up…**" message disappears and the "**Running!**" message appears:



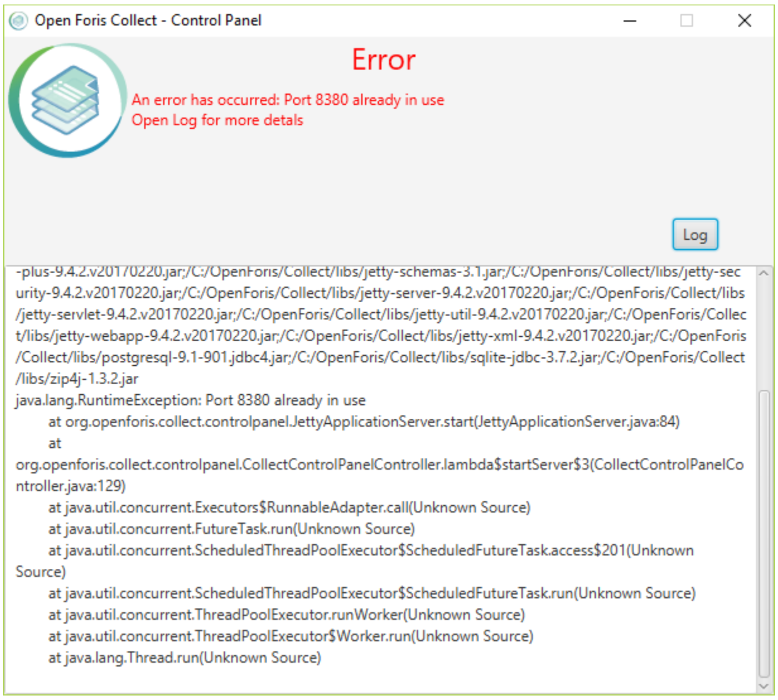
Collect running in default web browser at:
http://localhost:8380/collect

- Collect will open automatically a new window in the default web browser, linking the address: 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 home page should appear on the web browser. [If nothing happens, open a new browser window and access this url: http://localhost:8380/collect]

- By default, Collect runs with the user “**admin**” with administrative rights and you don’t see the login screen. If you add more users or you change the password for the “admin” user, next time you run Collect you will see a login page. The default password for the “admin” user is “**Collect123**”. You can change it by clicking on the dropdown menu button in the bottom left corner and selecting “**Change password**”:



- 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 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.



Error

An error has occurred: Port 8380 already in use
Open Log for more detals

```
-plus-9.4.2.v20170220.jar;/C:/OpenForis/Collect/libs/jetty-schemas-3.1.jar;/C:/OpenForis/Collect/libs/jetty-sec
urity-9.4.2.v20170220.jar;/C:/OpenForis/Collect/libs/jetty-server-9.4.2.v20170220.jar;/C:/OpenForis/Collect/libs
/jetty-servlet-9.4.2.v20170220.jar;/C:/OpenForis/Collect/libs/jetty-util-9.4.2.v20170220.jar;/C:/OpenForis/Collec
t/libs/jetty-webapp-9.4.2.v20170220.jar;/C:/OpenForis/Collect/libs/jetty-xml-9.4.2.v20170220.jar;/C:/OpenForis
/Collect/libs/postgresql-9.1-901.jdbc4.jar;/C:/OpenForis/Collect/libs/sqlite-jdbc-3.7.2.jar;/C:/OpenForis/Collect
/libs/zip4j-1.3.2.jar
java.lang.RuntimeException: Port 8380 already in use
	at org.openforis.collect.controlpanel.JettyApplicationServer.start(JettyApplicationServer.java:84)
	at
org.openforis.collect.controlpanel.CollectControlPanelController.lambda$startServer$3(CollectControlPanelCo
ntroller.java:129)
	at java.util.concurrent.Executors$RunnableAdapter.call(Unknown Source)
	at java.util.concurrent.FutureTask.run(Unknown Source)
	at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$201(Unknown
Source)
	at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(Unknown Source)
	at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
	at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
	at java.lang.Thread.run(Unknown Source)
```

- 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).

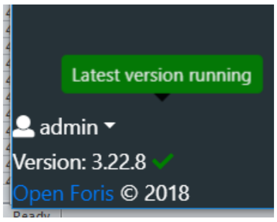
6 Shutdown Collect

- Close all the opened browser windows that are using Collect
- Close the Open Foris Collect Control Panel by clicking on the Shutdown button or just close it by using the close icon button in the top right corner.

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.



Latest version running

admin

Version: 3.22.8 ✓

Open Foris © 2018

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:

   https://oss.sonatype.org/service/local/artifact/maven/redirect?r=releases&g=org.openforis.collect&a=collect-updater&v=LATEST&c=windows&e=exe

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

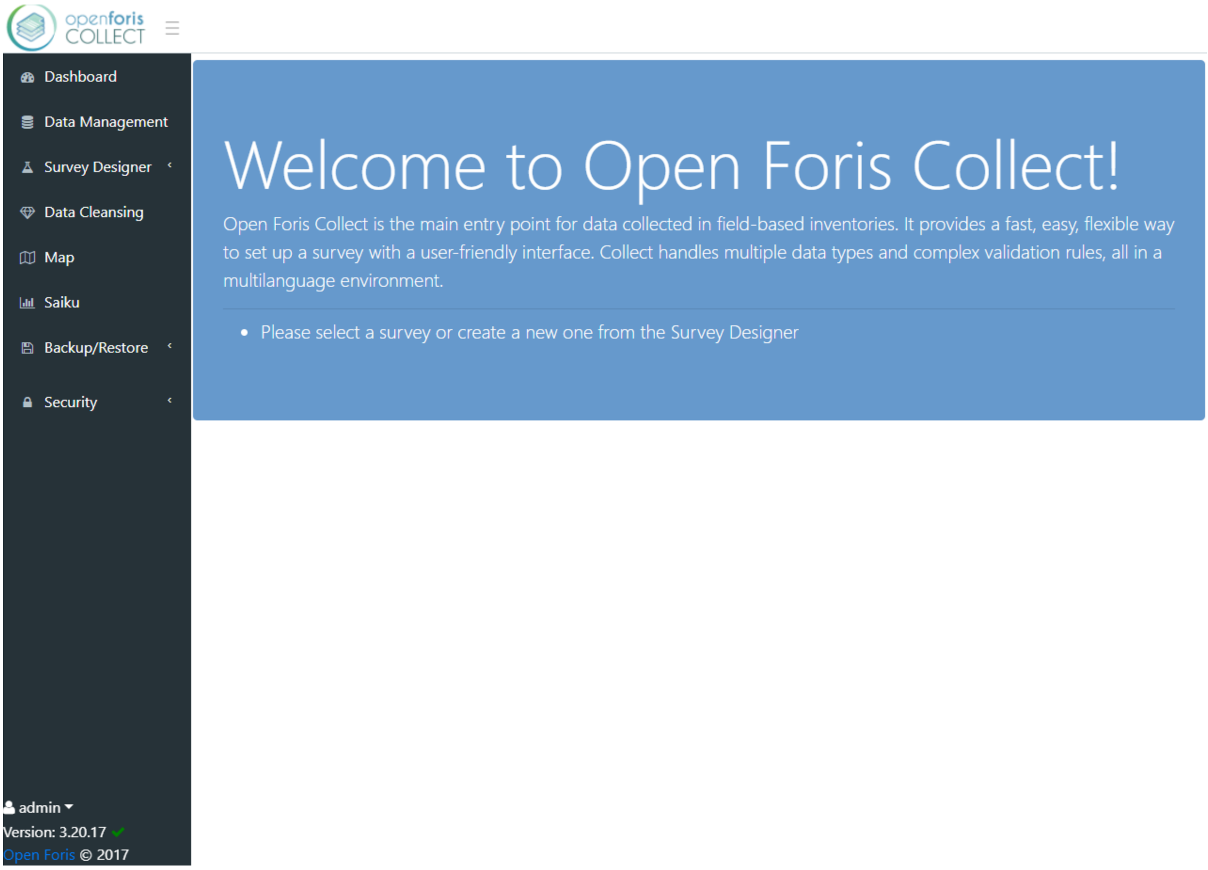
Warning: if you installed Collect before April 2017 (i.e. you are using older version than 3.12.30) and you are using SQLite database in your local machine, copy your data before uninstalling Collect, otherwise you will lose all your Collect data! If you are using PostgreSQL database for storing Collect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

7.4 Update a multi-user installation of Collect (for expert users only)

If you have already installed Collect for a multi-user environment and you want to update it to the latest version, follow these steps:

- shutdown Tomcat
- take a full backup of your database (if you are using PostgreSQL, use PGAdmin and take a backup with Custom format)
- take a backup of the collect.war that is inside the tomcat/webapps folder and move it into another location outside of the Tomcat folder
- download the latest version of the Collect war file
- put it inside tomcat/webapps folder and rename it into collect.war
- startup Tomcat

8 Collect Home



Welcome to Open Foris Collect!

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 multilanguage environment.

- Please select a survey or create a new one from the Survey Designer

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

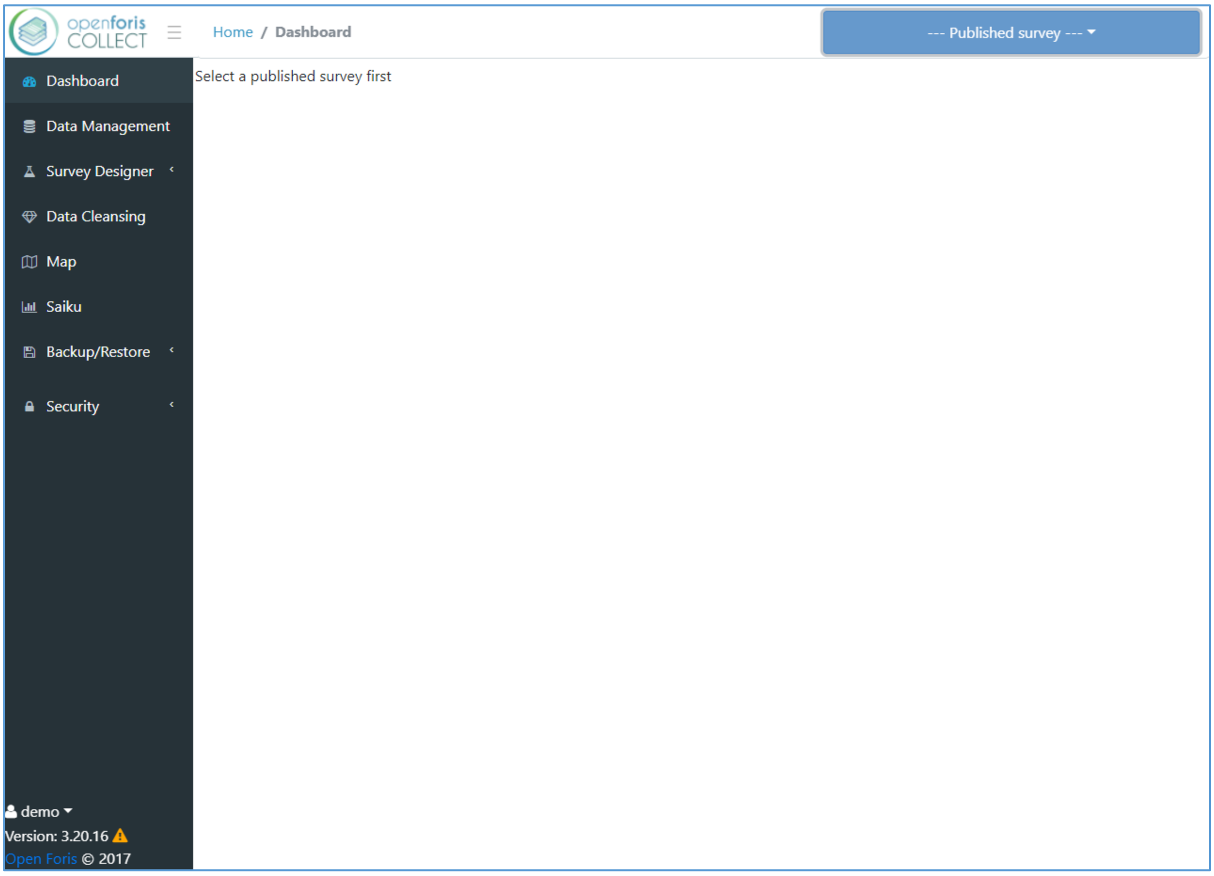
- **Dashboard**: for each published survey, it allows to get an overview on the data collection process, by showing statistics on the records already created, modified, entered, and/or cleansed.
- **Data management**: once a survey has been created, select Data management to start entering data and manage 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

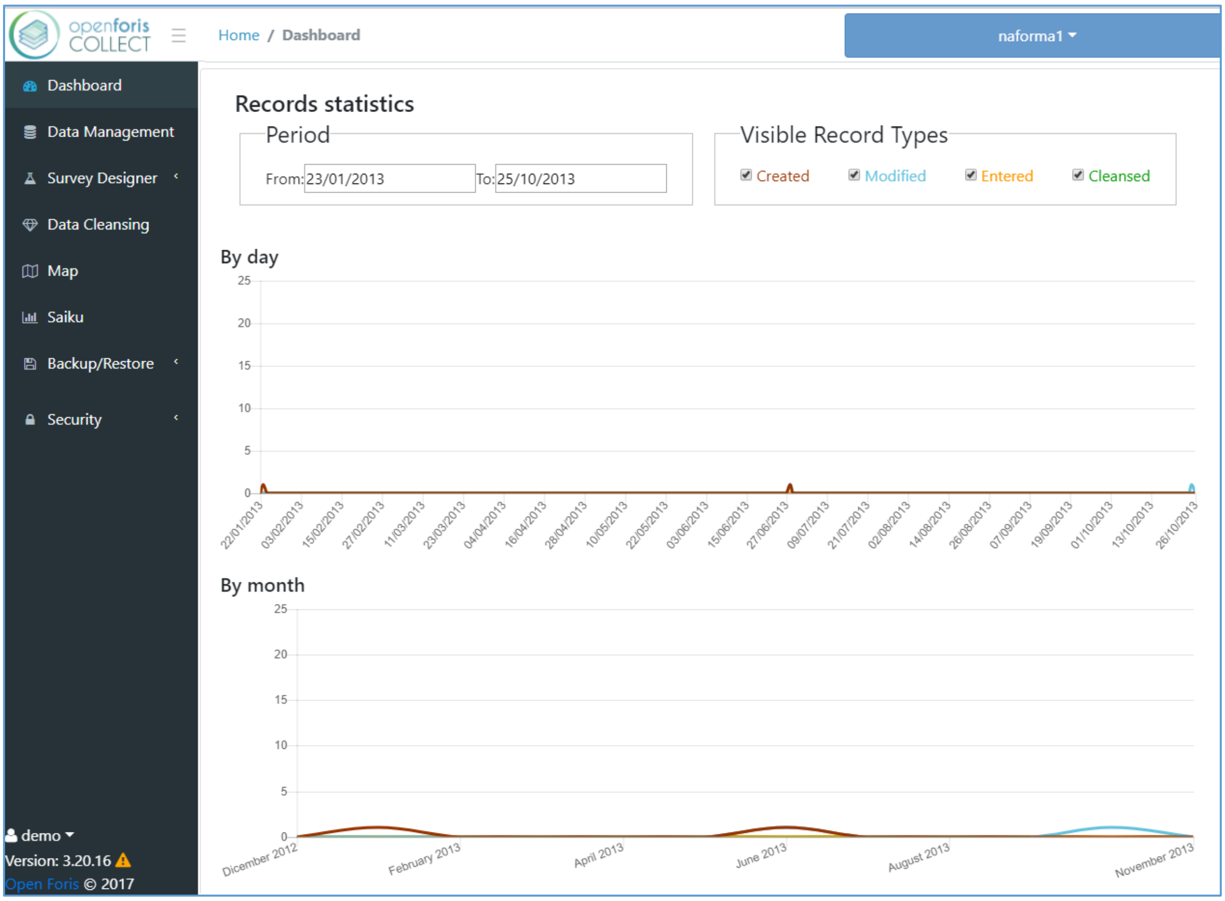
9 DASHBOARD

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.



Select a published survey first

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.



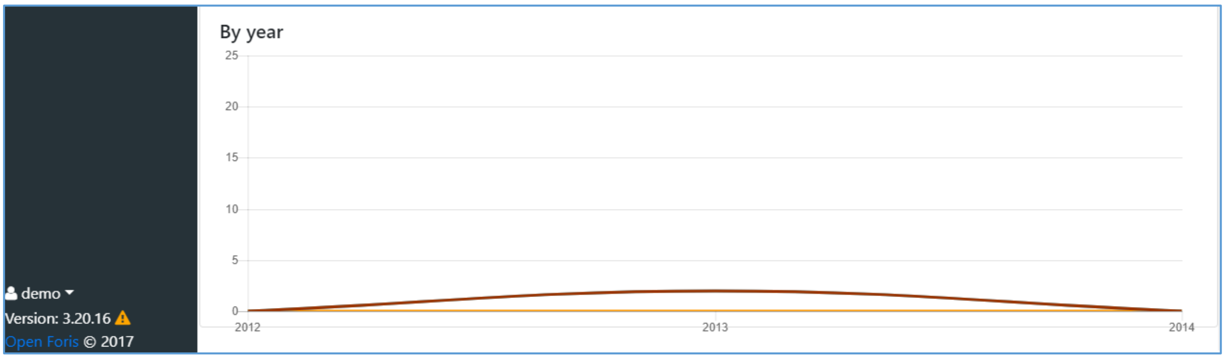
Records statistics

Period

From: 23/01/2013 To: 25/10/2013

Visible Record Types

☑ Created ☑ Modified ☑ Entered ☑ Cleansed

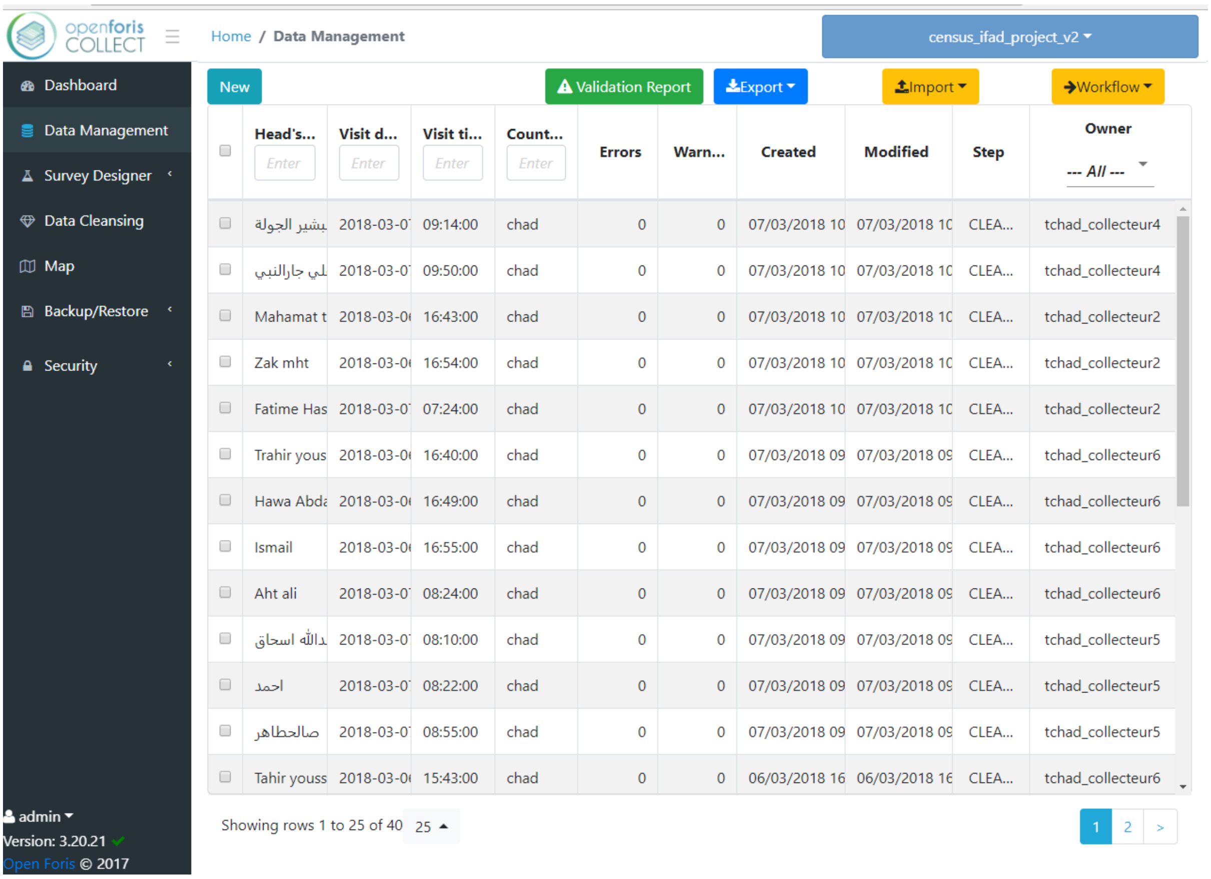


10 Data Management

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.



□	Head's...	Visit d...	Visit ti...	Count...	Errors	Warn...	Created	Modified	Step	Owner
	Enter	Enter	Enter	Enter						--- All ---
□	بشير الجولة	2018-03-07	09:14:00	chad	0	0	07/03/2018 10	07/03/2018 10	CLEA...	tchad_collecteur4
□	لي جارالنبي	2018-03-07	09:50:00	chad	0	0	07/03/2018 10	07/03/2018 10	CLEA...	tchad_collecteur4
□	Mahamat t	2018-03-06	16:43:00	chad	0	0	07/03/2018 10	07/03/2018 10	CLEA...	tchad_collecteur2
□	Zak mht	2018-03-06	16:54:00	chad	0	0	07/03/2018 10	07/03/2018 10	CLEA...	tchad_collecteur2
□	Fatime Has	2018-03-07	07:24:00	chad	0	0	07/03/2018 10	07/03/2018 10	CLEA...	tchad_collecteur2
□	Trahir yous	2018-03-06	16:40:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur6
□	Hawa Abda	2018-03-06	16:49:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur6
□	Ismail	2018-03-06	16:55:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur6
□	Aht ali	2018-03-07	08:24:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur6
□	دالله اسحاق	2018-03-07	08:10:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur5
□	احمد	2018-03-07	08:22:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur5
□	صالحطاهر	2018-03-07	08:55:00	chad	0	0	07/03/2018 09	07/03/2018 09	CLEA...	tchad_collecteur5
□	Tahir youss	2018-03-06	15:43:00	chad	0	0	06/03/2018 16	06/03/2018 16	CLEA...	tchad_collecteur6

Showing rows 1 to 25 of 40 25

1 2 >

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.

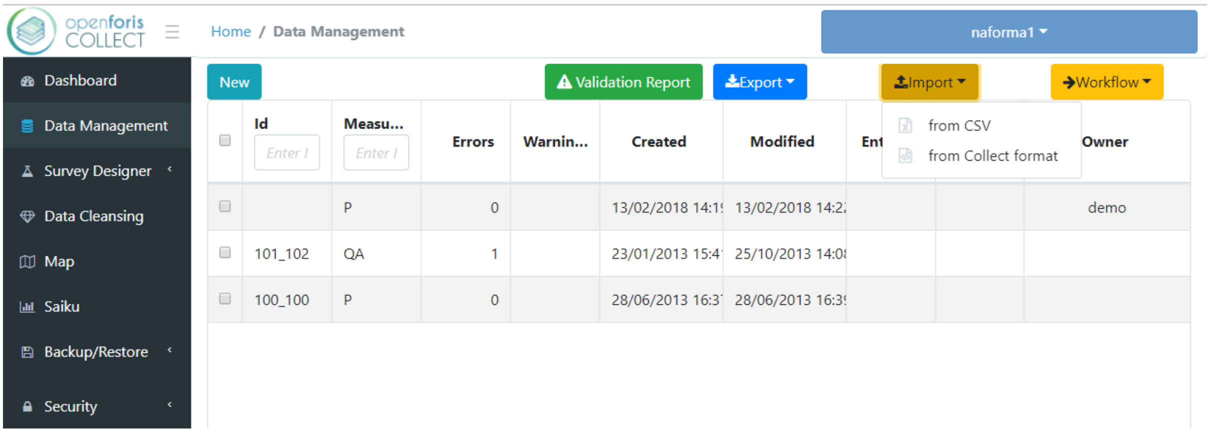
Data Archiving

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

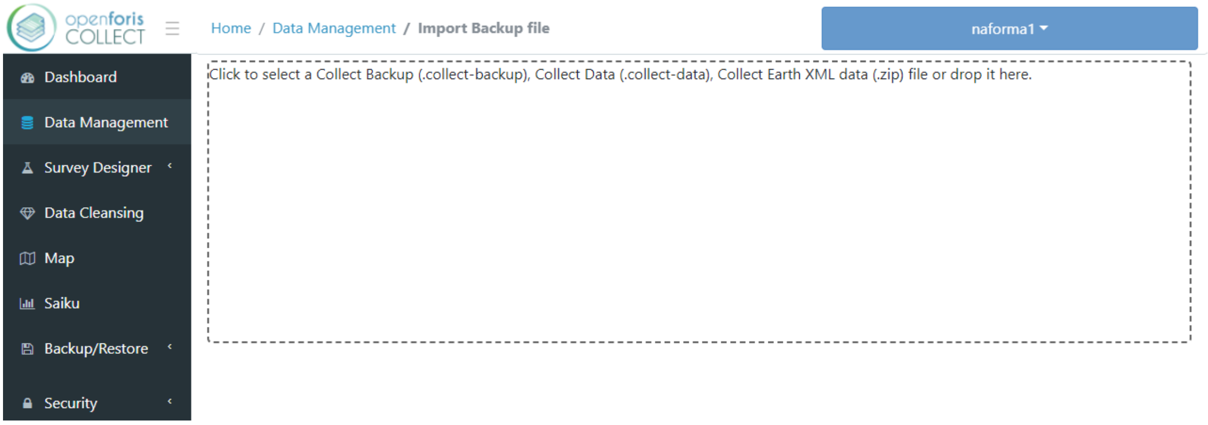
10.1.1Data upload (from Collect Mobile)

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).



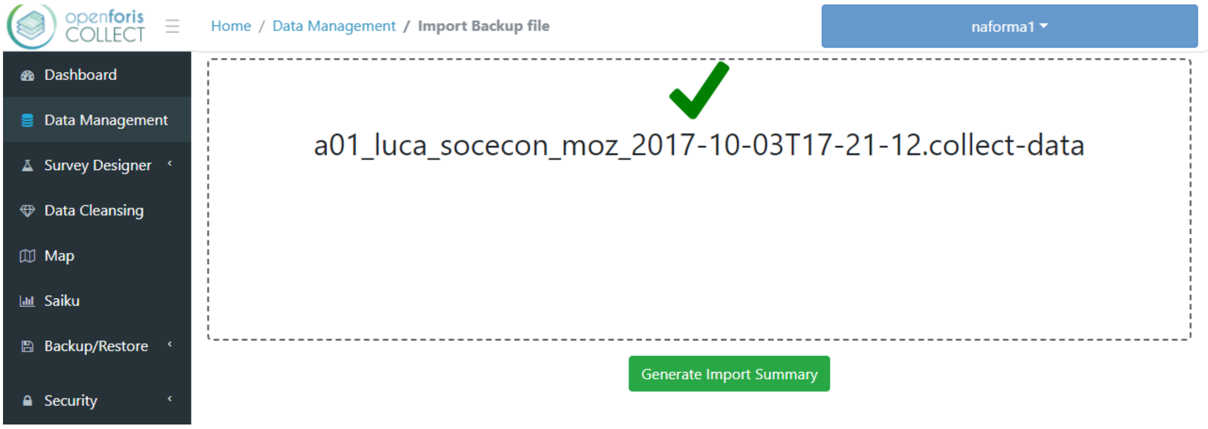
□	Id	Measu...	Errors	Warnin...	Created	Modified	Ent		Owner
	Enter I	Enter I							
□		P	0		13/02/2018 14:1!	13/02/2018 14:2;			demo
□	101_102	QA	1		23/01/2013 15:4	25/10/2013 14:0!			
□	100_100	P	0		28/06/2013 16:3	28/06/2013 16:3!			

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.



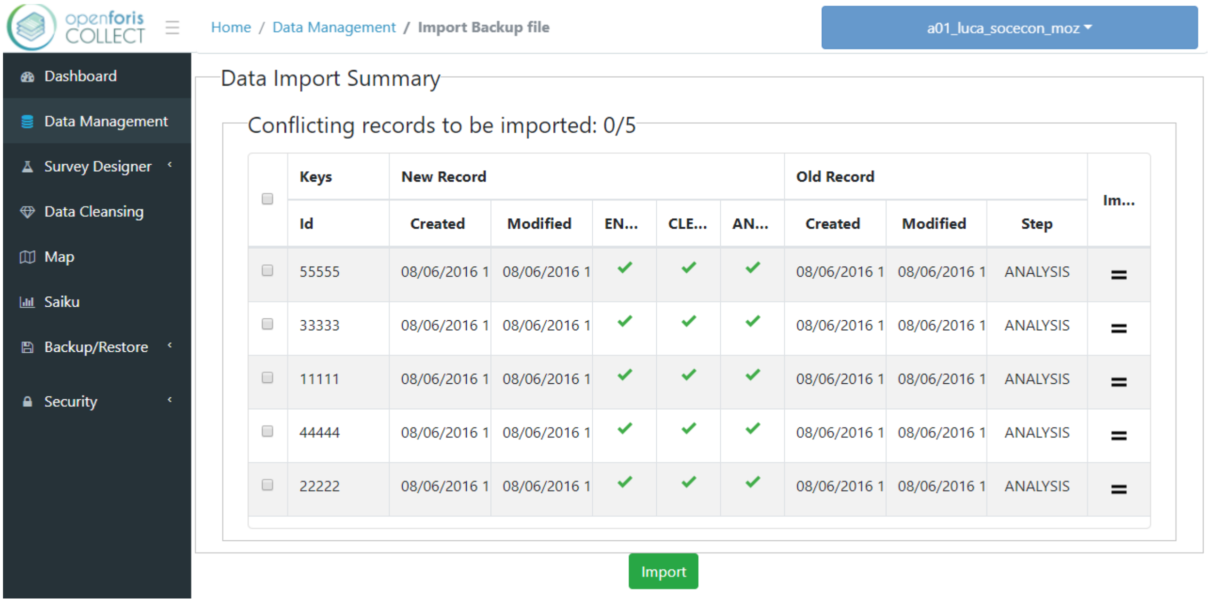
Click to select a Collect Backup (.collect-backup), Collect Data (.collect-data), Collect Earth XML data (.zip) file or drop it here.

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.



a01_luca_socecon_moz_2017-10-03T17-21-12.collect-data

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.



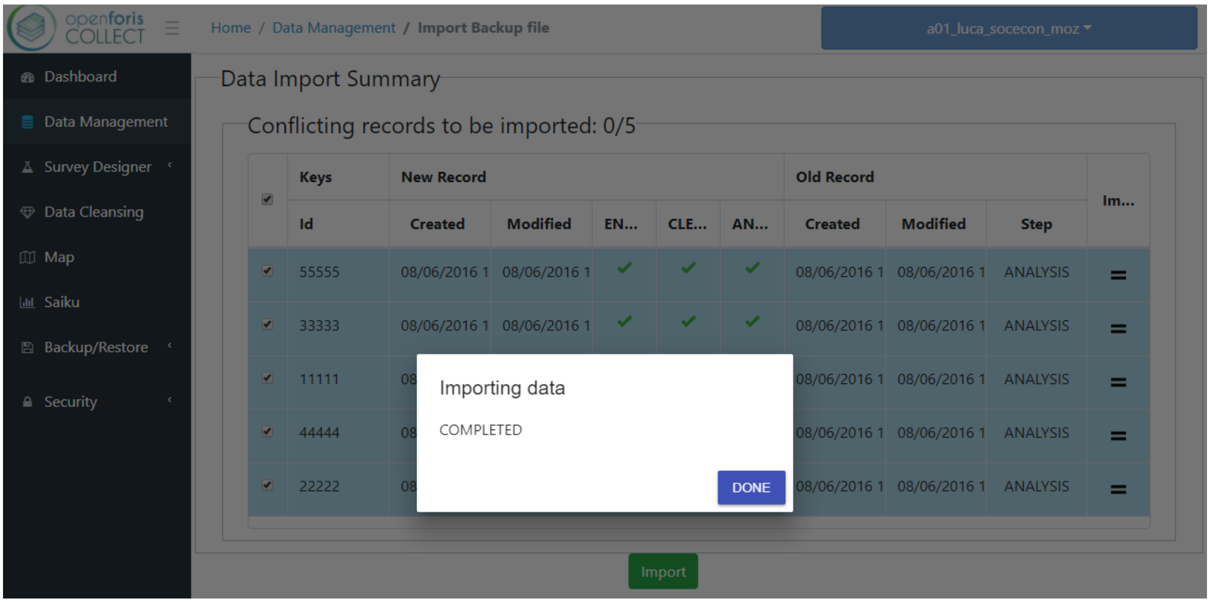
Data Import Summary

Conflicting records to be imported: 0/5

☐	Keys	New Record					Old Record			Im...
	Id	Created	Modified	EN...	CLE...	AN...	Created	Modified	Step	
☐	55555	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☐	33333	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☐	11111	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☐	44444	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☐	22222	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=

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.



Data Import Summary

Conflicting records to be imported: 0/5

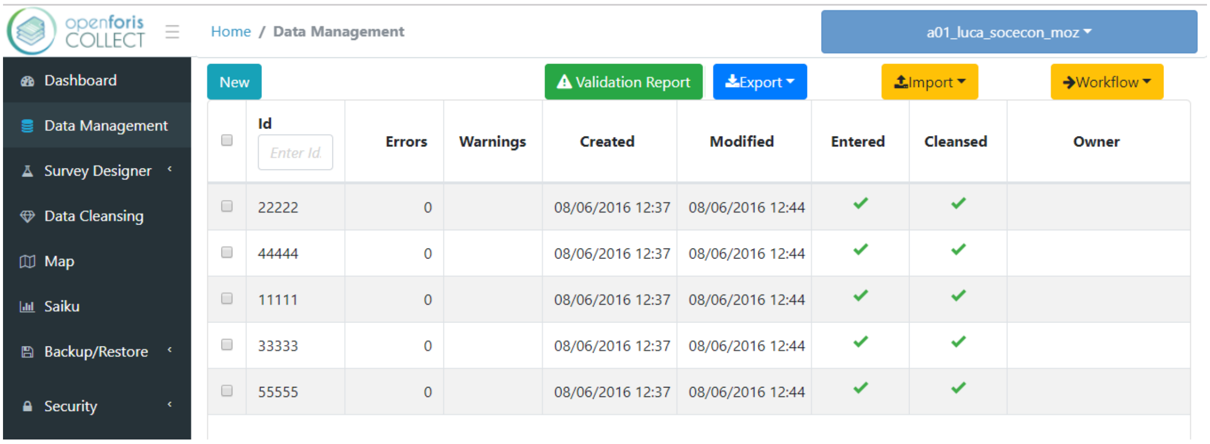
☑	Keys	New Record					Old Record			Im...
	Id	Created	Modified	EN...	CLE...	AN...	Created	Modified	Step	
☑	55555	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☑	33333	08/06/2016 1	08/06/2016 1	✓	✓	✓	08/06/2016 1	08/06/2016 1	ANALYSIS	=
☑	11111	08					08/06/2016 1	08/06/2016 1	ANALYSIS	=
☑	44444	08					08/06/2016 1	08/06/2016 1	ANALYSIS	=
☑	22222	08					08/06/2016 1	08/06/2016 1	ANALYSIS	=

Importing data

COMPLETED

DONE

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.

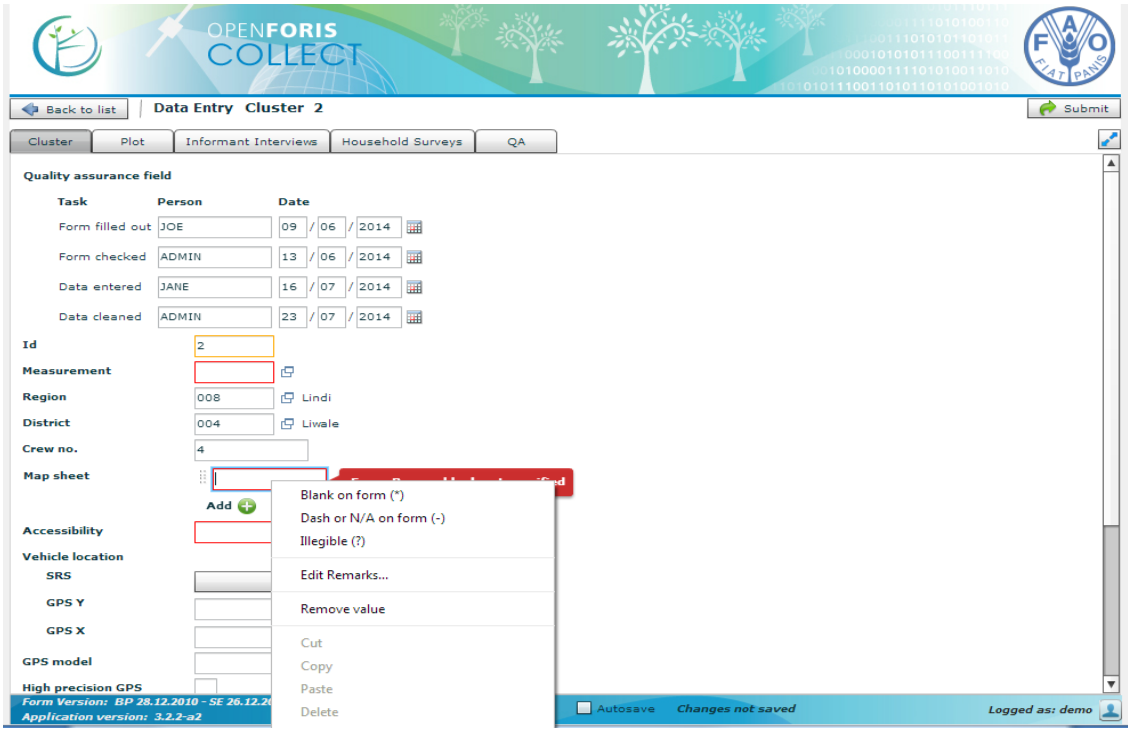


□	Id Enter Id	Errors	Warnings	Created	Modified	Entered	Cleansed	Owner
□	22222	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	44444	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	11111	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	33333	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	55555	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	

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).



Data Entry Cluster 2

Quality assurance field

Task	Person	Date
Form filled out	JOE	09 / 06 / 2014
Form checked	ADMIN	13 / 06 / 2014
Data entered	JANE	16 / 07 / 2014
Data cleaned	ADMIN	23 / 07 / 2014

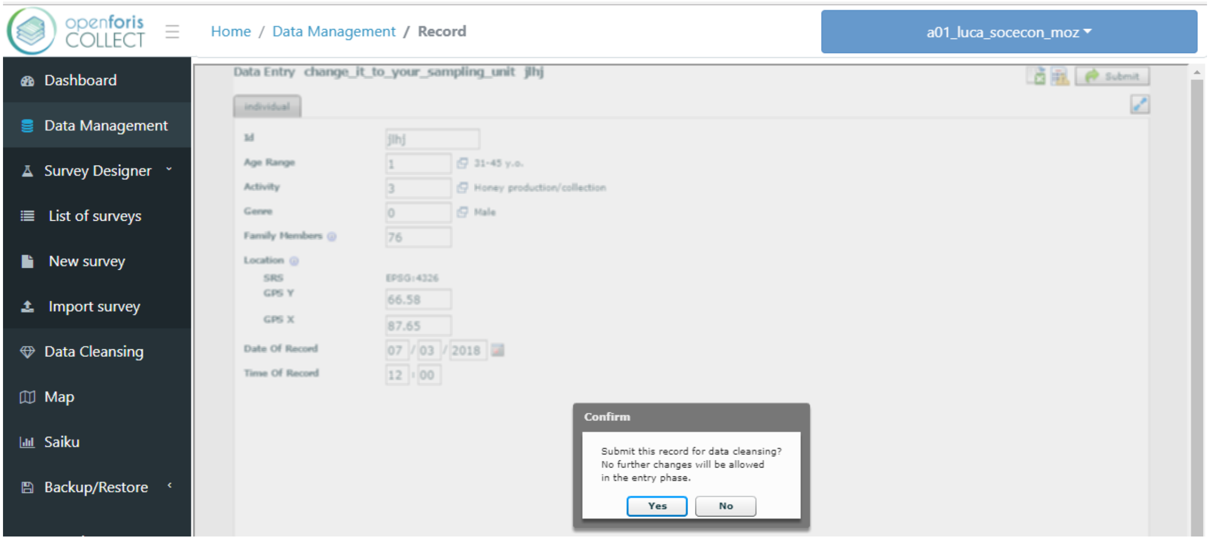
Id	2	
Measurement		
Region	008	Lindi
District	004	Liwale
Crew no.	4	
Map sheet		
Accessibility		
Vehicle location		
SRS		
GPS Y		
GPS X		
GPS model		
High precision GPS		

Form Version: BP 28.12.2010 - SE 26.12.2
Application version: 3.2.2-a2

Changes not saved

Logged as: demo

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.



Data Entry change_it_to_your_sampling_unit jlhj

Id	jlhj	
Age Range	1	31-45 y.o.
Activity	3	Honey production/collection
Genre	0	Male
Family Members	76	
Location		
SRS	EPSG:4326	
GPS Y	66.58	
GPS X	87.65	
Date Of Record	07 / 03 / 2018	
Time Of Record	12 : 00	

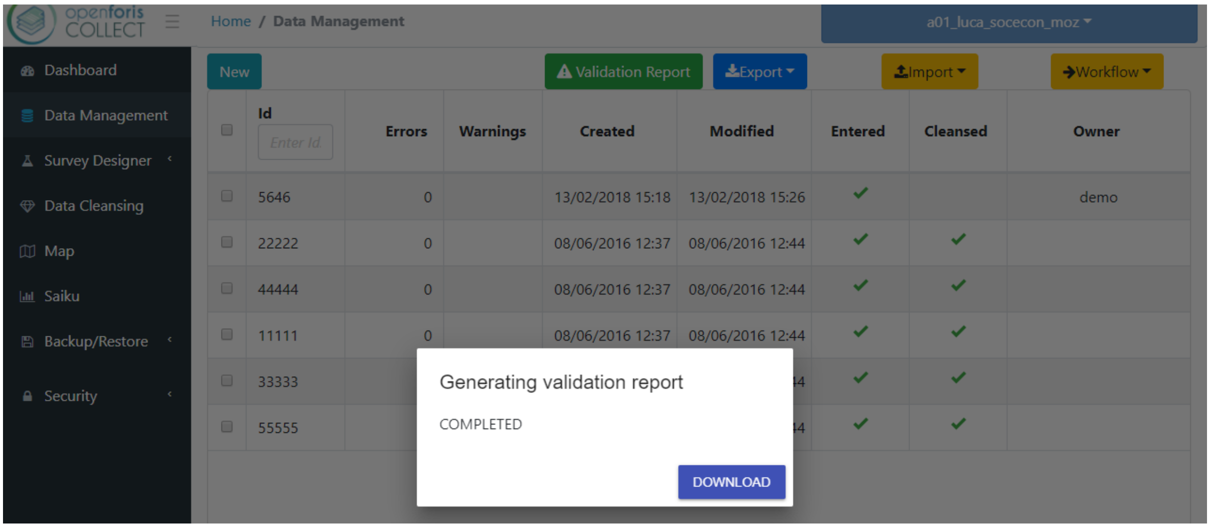
Confirm

Submit this record for data cleansing? No further changes will be allowed in the entry phase.

Yes No

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.



□	Id	Errors	Warnings	Created	Modified	Entered	Cleansed	Owner
□	5646	0		13/02/2018 15:18	13/02/2018 15:26	✓		demo
□	22222	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	44444	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	11111	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	33333				4	✓	✓	
□	55555				4	✓	✓	

Generating validation report

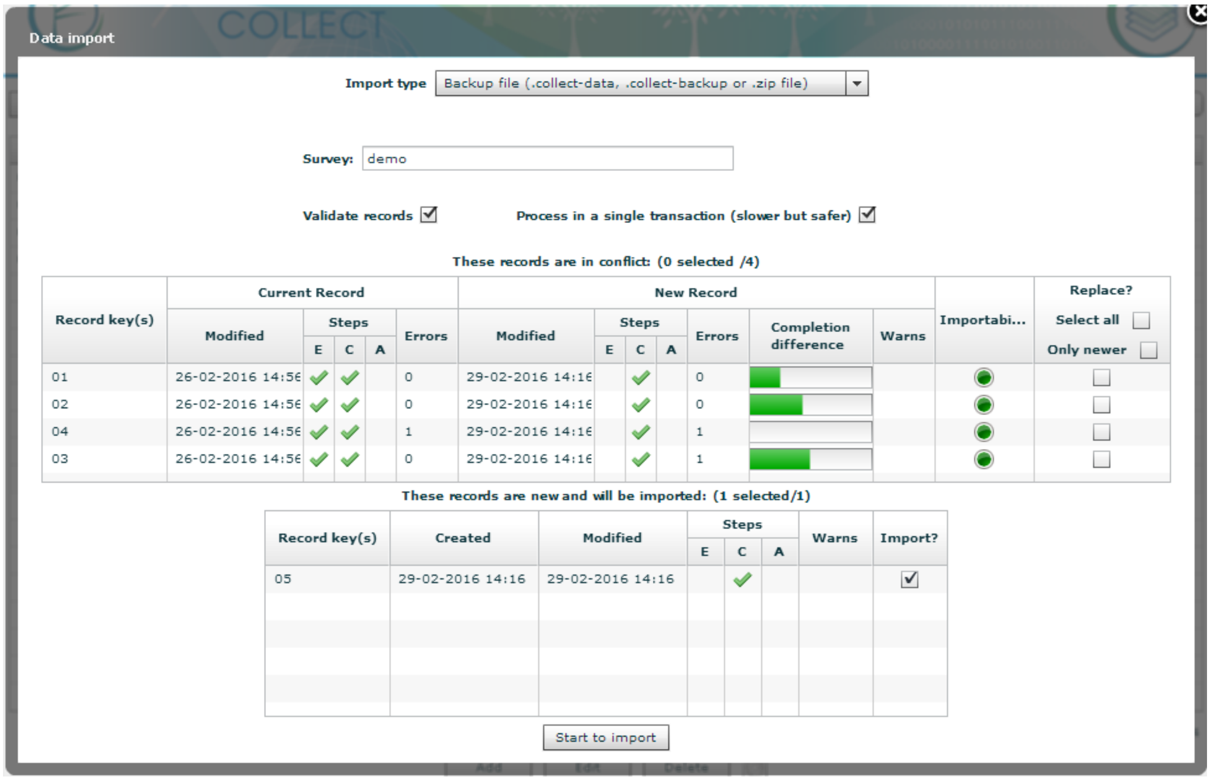
COMPLETED

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.



Import type Backup file (.collect-data, .collect-backup or .zip file)

Survey: demo

Validate records ☑ Process in a single transaction (slower but safer) ☑

These records are in conflict: (0 selected /4)

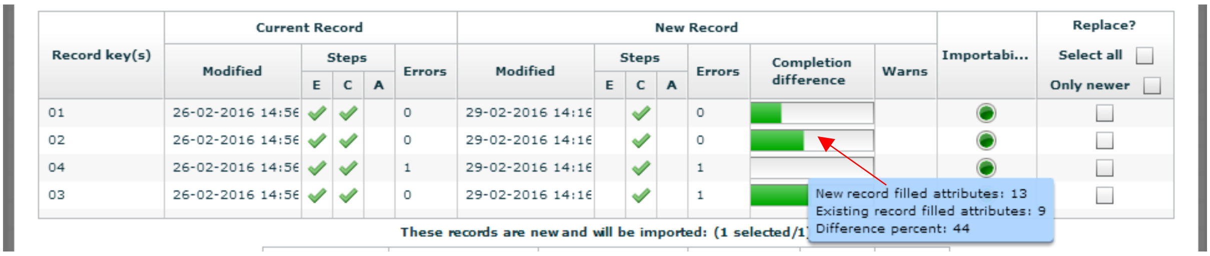
Record key(s)	Current Record					New Record							Importabi...	Replace?
	Modified	Steps			Errors	Modified	Steps			Errors	Completion difference	Warns		Select all □ Only newer □
		E	C	A			E	C	A					
01	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
02	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
04	26-02-2016 14:5€	✓	✓		1	29-02-2016 14:1€		✓		1			●	□
03	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		1	🟩		●	□

These records are new and will be imported: (1 selected/1)

Record key(s)	Created	Modified	Steps			Warns	Import?
			E	C	A		
05	29-02-2016 14:16	29-02-2016 14:16		✓			☑

Start to import

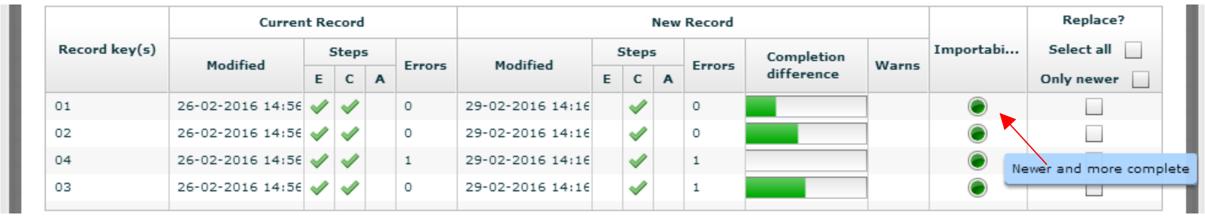
In the top table we see that for records #01, 02 and 03 a new and more complete version was sent with the second data export. As before, modification date and error numbers is presented. The column Completion difference (green increment bar) indicates the number of filled attributes of the new record 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.



Record key(s)	Current Record					New Record							Importabi...	Replace?
	Modified	Steps			Errors	Modified	Steps			Errors	Completion difference	Warns		Select all □ Only newer □
		E	C	A			E	C	A					
01	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
02	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
04	26-02-2016 14:5€	✓	✓		1	29-02-2016 14:1€		✓		1			●	□
03	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		1	🟩			□

New record filled attributes: 13
Existing record filled attributes: 9
Difference percent: 44

These records are new and will be imported: (1 selected/1



Record key(s)	Current Record					New Record							Importabi...	Replace?
	Modified	Steps			Errors	Modified	Steps			Errors	Completion difference	Warns		Select all □ Only newer □
		E	C	A			E	C	A					
01	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
02	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		0	🟩		●	□
04	26-02-2016 14:5€	✓	✓		1	29-02-2016 14:1€		✓		1			●	
03	26-02-2016 14:5€	✓	✓		0	29-02-2016 14:1€		✓		1	🟩		●	□

Newer and more complete

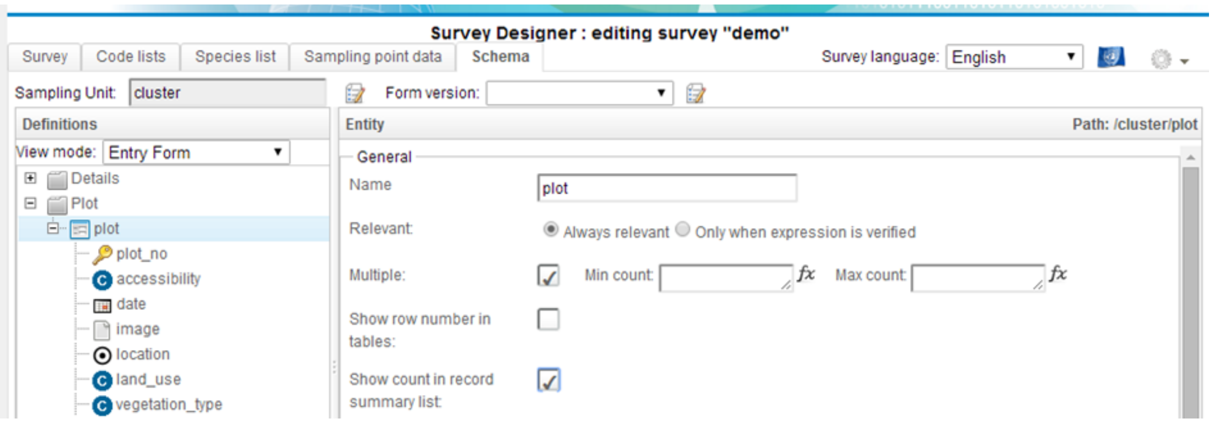
The last column on the right: "Replace?" is used to select those records that should be uploaded to replace older (and less complete) versions of the same records. Click to select record to be uploaded choosing from: select all, only newer (green dot), or select records individually. The bottom table, 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.



Survey Designer : editing survey "demo"

Survey language: English

Sampling Unit: cluster

Form version:

Definitions

View mode: Entry Form

Entity — Path: /cluster/plot

General

Name: plot

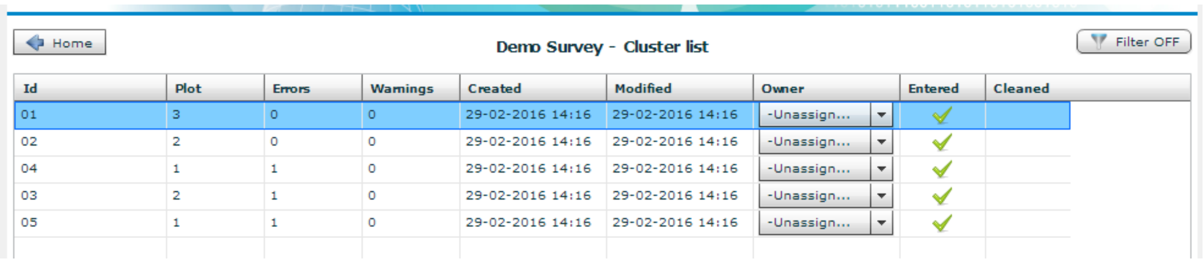
Relevant: ● Always relevant ○ Only when expression is verified

Multiple: ☑ Min count: Max count:

Show row number in tables: □

Show count in record summary list: ☑

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.

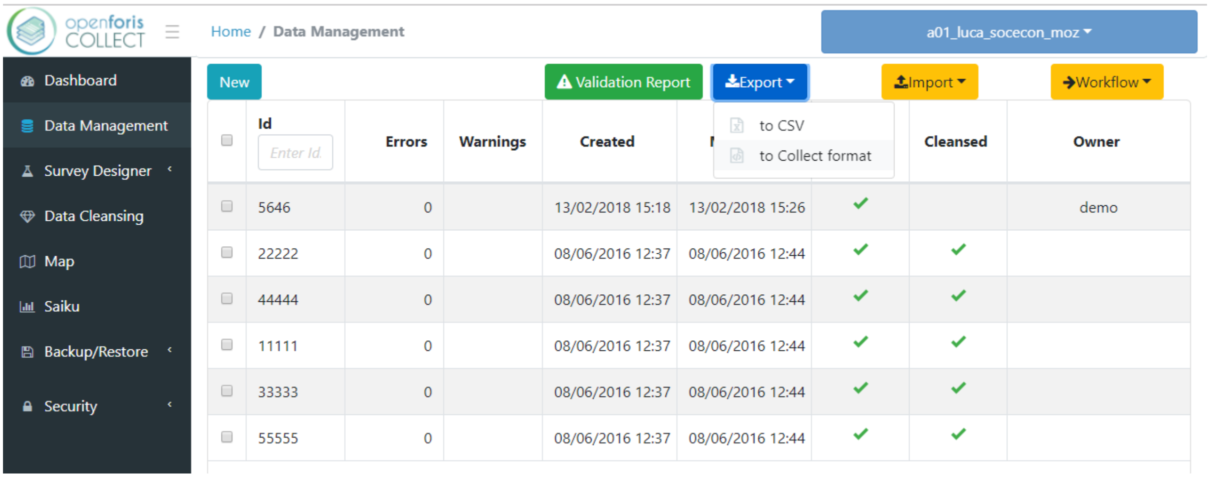


Demo Survey - Cluster list

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...	✓	

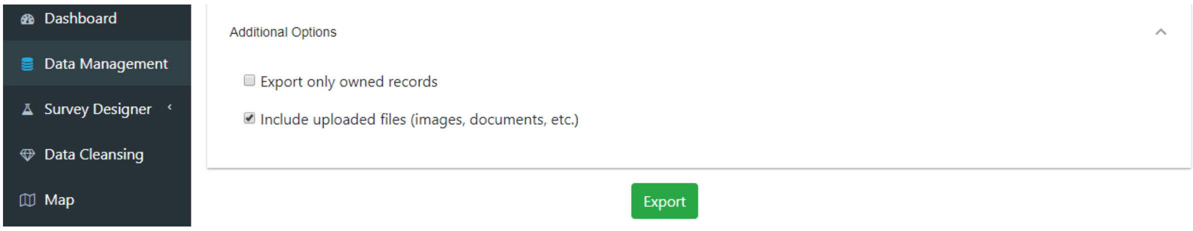
10.5 Data export

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.



□	Id	Errors	Warnings	Created	I		Cleansed	Owner
□	5646	0		13/02/2018 15:18	13/02/2018 15:26	✓		demo
□	22222	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	44444	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	11111	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	33333	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	
□	55555	0		08/06/2016 12:37	08/06/2016 12:44	✓	✓	

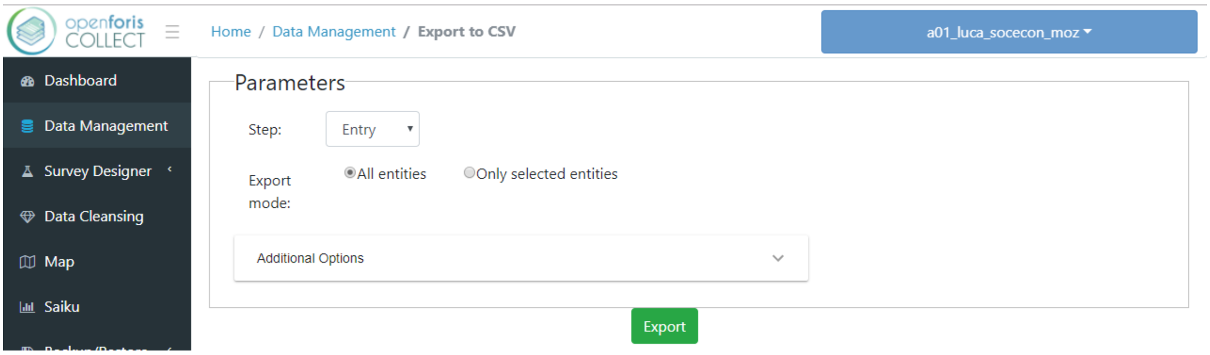
This function can also be used to perform a backup, with or without uploaded files (such as images – see below).



□ Export only owned records

☑ Include uploaded files (images, documents, etc.)

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



Step: Entry

Export mode: ● All entities ○ Only selected entities

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).

10.6 Data Workflow

Collect implements a workflow which divides data processing into three phases: data entry, data cleansing, and data analysis.

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)), 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

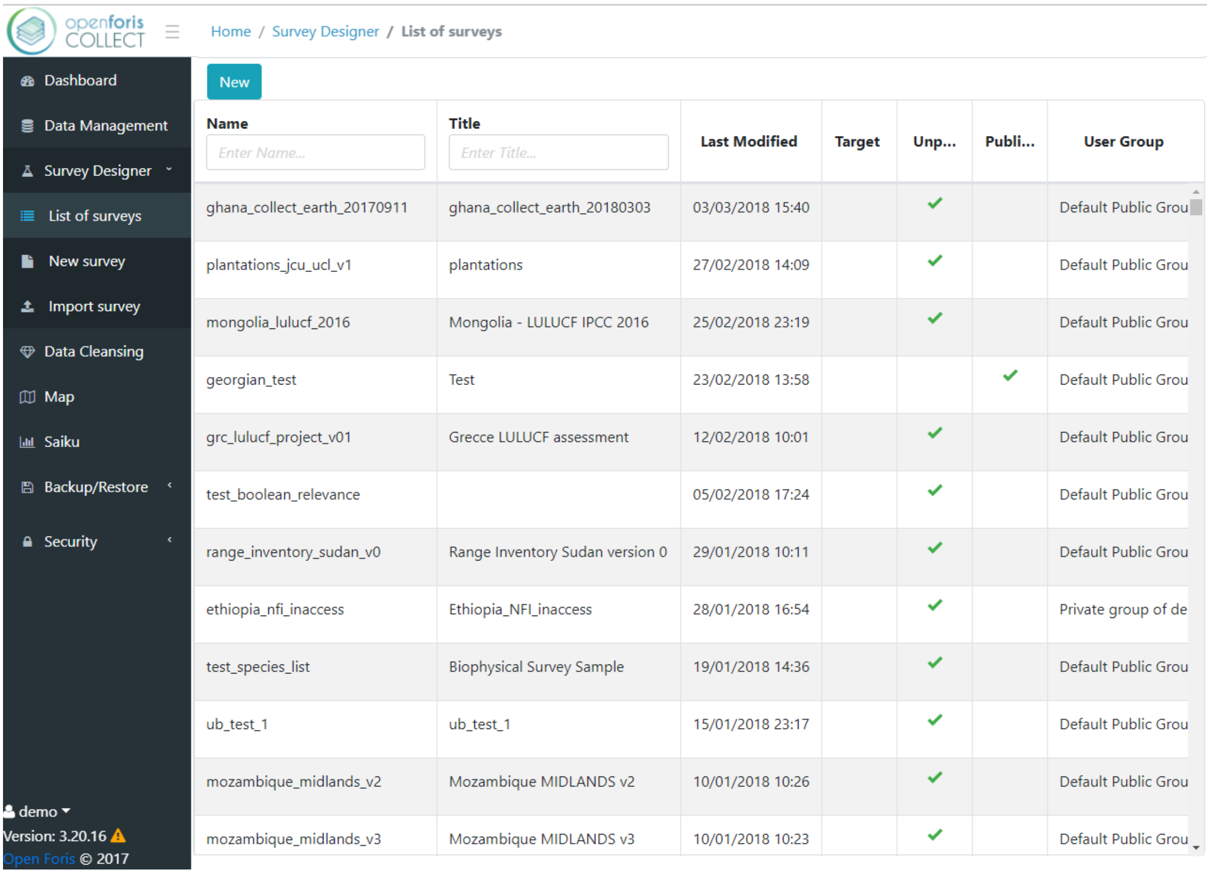
- Experts with knowledge of field conditions and data work with a copy of the data entered in phase 1 to correct or remove missing or invalid values. Estimated and imputed values may be entered in this phase and unavailable values removed.
- If the data cleansing expert deems that the data entry task 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.

10.6.3DATA 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.

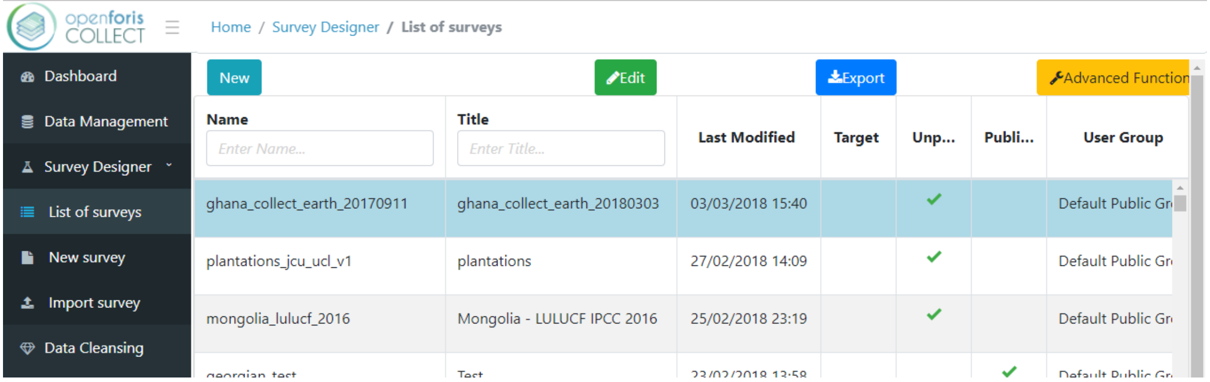
11 Survey Designer

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.



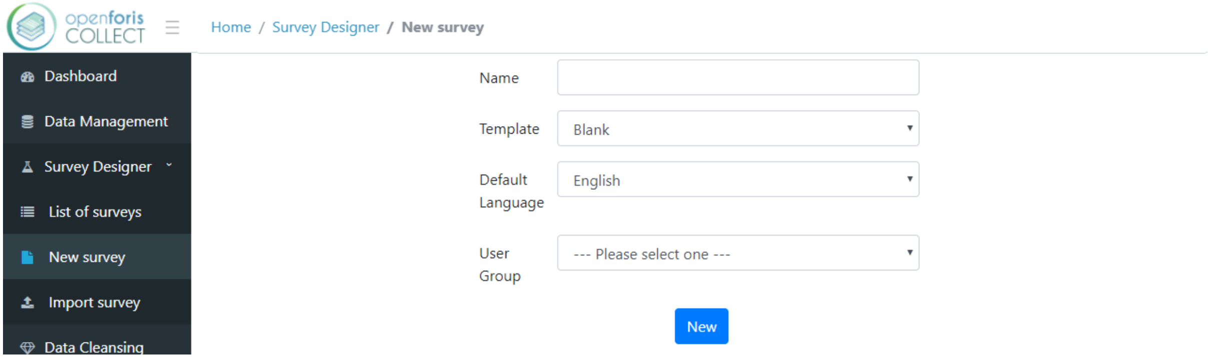
Name	Title	Last Modified	Target	Unp...	Publi...	User Group
Enter Name...	Enter Title...					
ghana_collect_earth_20170911	ghana_collect_earth_20180303	03/03/2018 15:40		✓		Default Public Grou
plantations_jcu_ucl_v1	plantations	27/02/2018 14:09		✓		Default Public Grou
mongolia_lulucf_2016	Mongolia - LULUCF IPCC 2016	25/02/2018 23:19		✓		Default Public Grou
georgian_test	Test	23/02/2018 13:58			✓	Default Public Grou
grc_lulucf_project_v01	Grecce LULUCF assessment	12/02/2018 10:01		✓		Default Public Grou
test_boolean_relevance		05/02/2018 17:24		✓		Default Public Grou
range_inventory_sudan_v0	Range Inventory Sudan version 0	29/01/2018 10:11		✓		Default Public Grou
ethiopia_nfi_inaccess	Ethiopia_NFI_inaccess	28/01/2018 16:54		✓		Private group of de
test_species_list	Biophysical Survey Sample	19/01/2018 14:36		✓		Default Public Grou
ub_test_1	ub_test_1	15/01/2018 23:17		✓		Default Public Grou
mozambique_midlands_v2	Mozambique MIDLANDS v2	10/01/2018 10:26		✓		Default Public Grou
mozambique_midlands_v3	Mozambique MIDLANDS v3	10/01/2018 10:23		✓		Default Public Grou

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.



Name	Title	Last Modified	Target	Unp...	Publi...	User Group
Enter Name...	Enter Title...					
ghana_collect_earth_20170911	ghana_collect_earth_20180303	03/03/2018 15:40		✓		Default Public Gr
plantations_jcu_ucl_v1	plantations	27/02/2018 14:09		✓		Default Public Gr
mongolia_lulucf_2016	Mongolia - LULUCF IPCC 2016	25/02/2018 23:19		✓		Default Public Gr
georgian_test	Test	23/02/2018 13:58			✓	Default Public Gr

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.



Name:

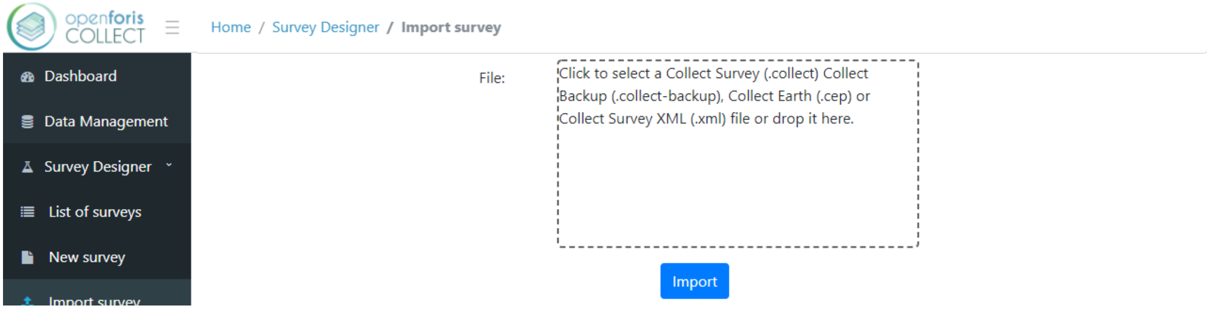
Template: Blank

Default Language: English

User Group: --- Please select one ---

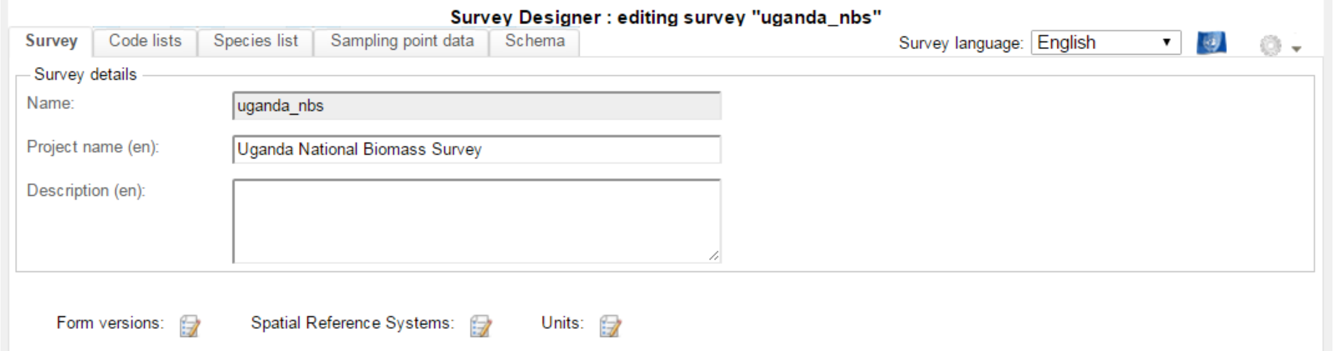
Note: additional languages can also be added at a later stage by clicking on the UN flag (top right corner) and select and add additional languages for the survey. The user can then switch from one language to another by selecting a language in the drop-down menu on the top right corner of the screen. 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).



File: Click to select a Collect Survey (.collect) Collect Backup (.collect-backup), Collect Earth (.cep) or Collect Survey XML (.xml) file or drop it here.

11.1 New Survey



Survey Designer : editing survey "uganda_nbs"

Survey language: English

Survey details

Name: uganda_nbs

Project name (en): Uganda National Biomass Survey

Description (en):

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.

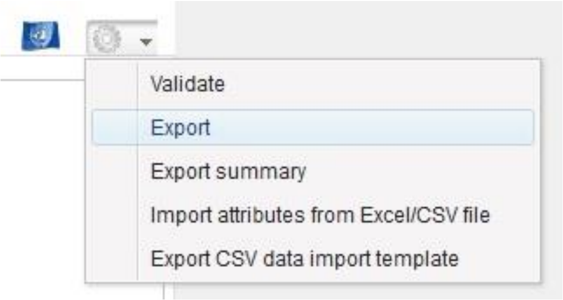
NOTE: Try to avoid using different form version in you survey. This feature can cause troubles in data analysis when using Open Foris Calc phase if different versions have different variables and/or data types.

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
Export
Export summary
Import attributes from Excel/CSV file
Export CSV data import template

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.

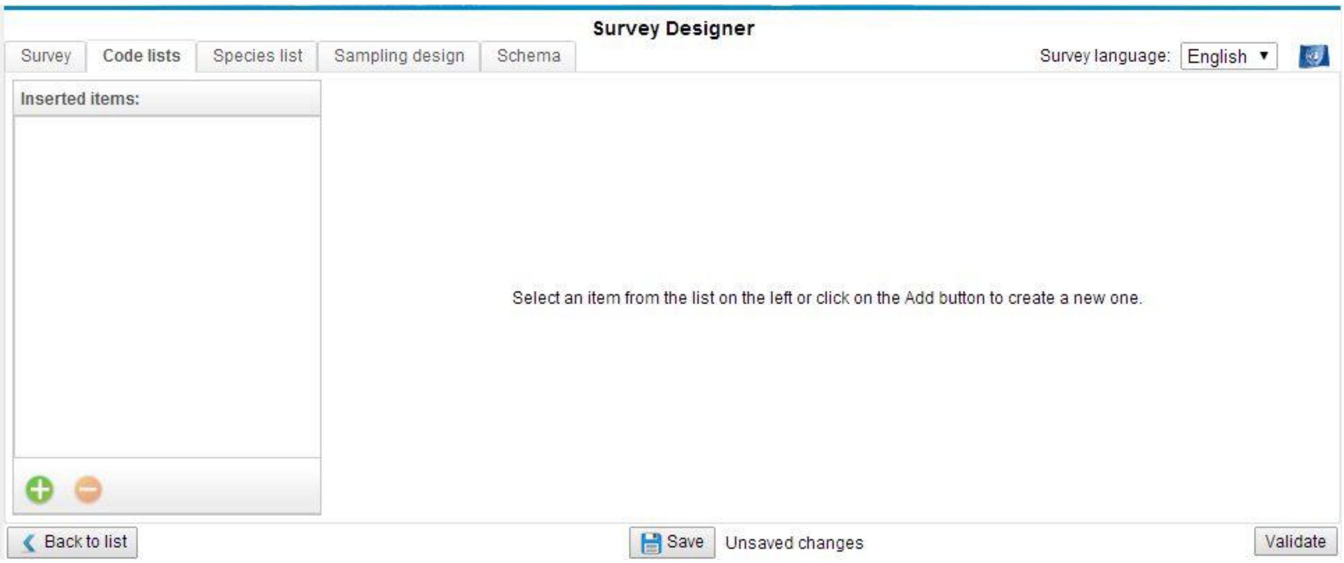
Export schema summary: Creates a CSV file that summarizes the data structure of the survey. This is useful for defining conditions for attributes' relevance, for example.

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.



Survey Designer

Select an item from the list on the left or click on the Add button to create a new one.

Unsaved changes

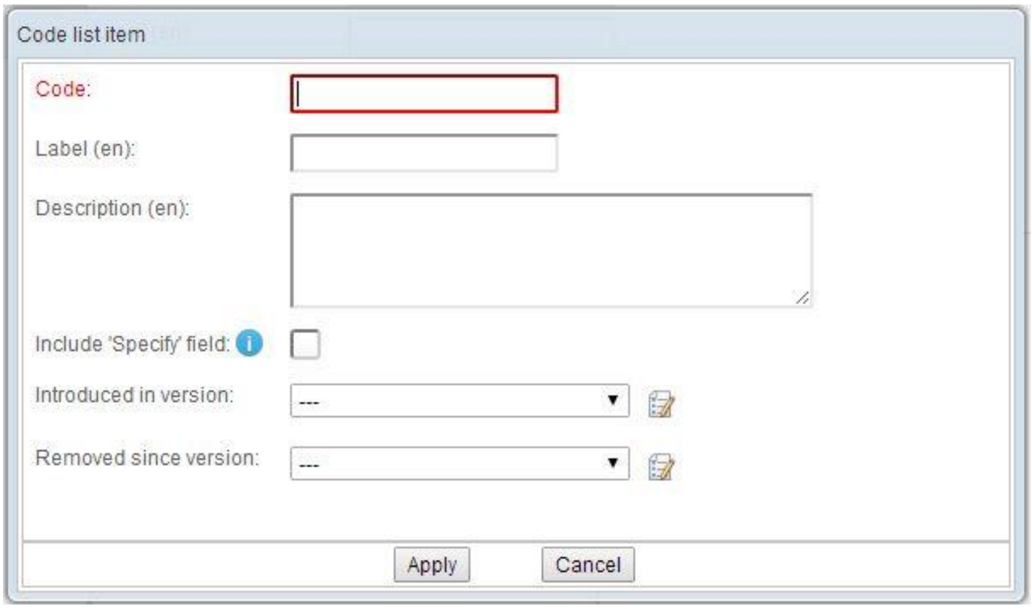
Adding a Code list

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.



Code list item

Code:

Label (en):

Description (en):

Include 'Specify' field: □

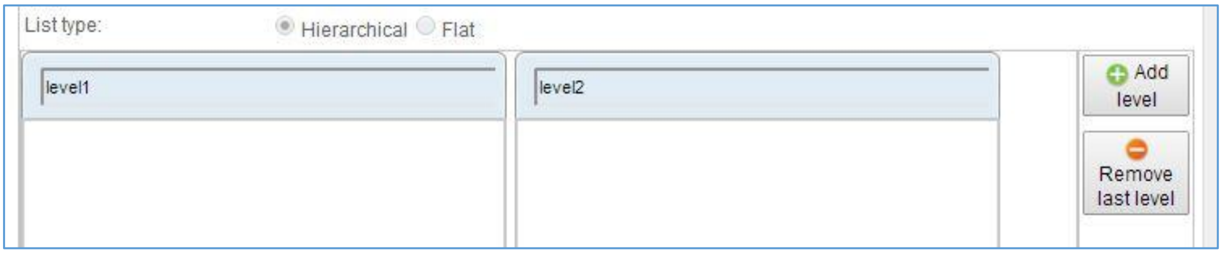
Introduced in version: ---

Removed since version: ---

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.



List type: ● Hierarchical ○ Flat

level1

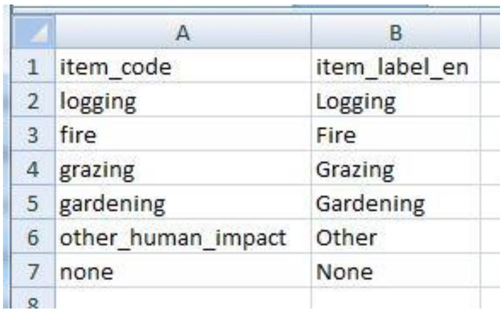
level2

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.

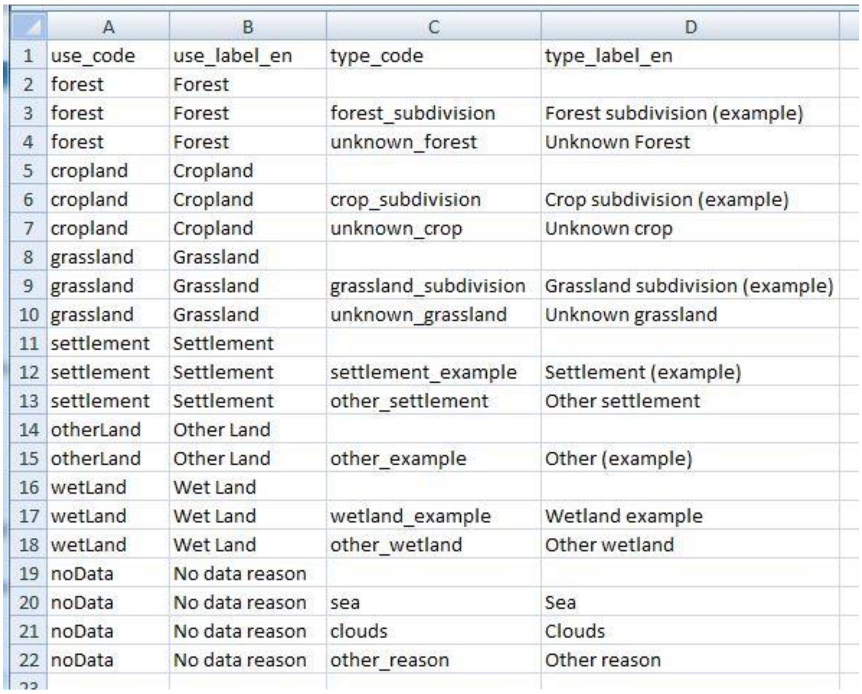
To import multiple csv files at the same time, create a compressed folder containing the files and select “Batch import”. The code lists will be visible once the import is completed like when importing single files.

Example of Flat Code list



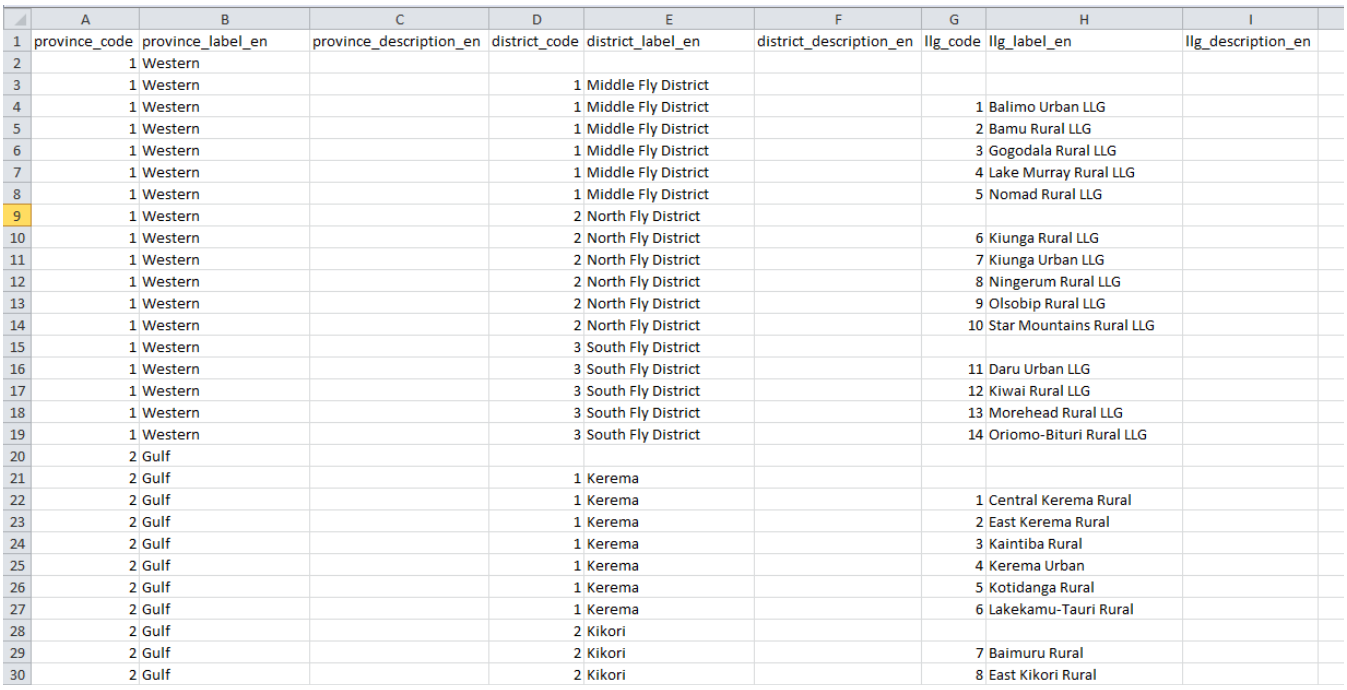
	A	B
1	item_code	item_label_en
2	logging	Logging
3	fire	Fire
4	grazing	Grazing
5	gardening	Gardening
6	other_human_impact	Other
7	none	None

Example of Hierarchical Code list - Vegetation types



	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

Example of Hierarchical Code list - Administrative units



	A	B	C	D	E	F	G	H	I
1	province_code	province_label_en	province_description_en	district_code	district_label_en	district_description_en	llg_code	llg_label_en	llg_description_en
2	1	Western							
3	1	Western		1	Middle Fly District				
4	1	Western		1	Middle Fly District		1	Balimo Urban LLG	
5	1	Western		1	Middle Fly District		2	Bamu Rural LLG	
6	1	Western		1	Middle Fly District		3	Gogodala Rural LLG	
7	1	Western		1	Middle Fly District		4	Lake Murray Rural LLG	
8	1	Western		1	Middle Fly District		5	Nomad Rural LLG	
9	1	Western		2	North Fly District				
10	1	Western		2	North Fly District		6	Kiunga Rural LLG	
11	1	Western		2	North Fly District		7	Kiunga Urban LLG	
12	1	Western		2	North Fly District		8	Ningerum Rural LLG	
13	1	Western		2	North Fly District		9	Olsobip Rural LLG	
14	1	Western		2	North Fly District		10	Star Mountains Rural LLG	
15	1	Western		3	South Fly District				
16	1	Western		3	South Fly District		11	Daru Urban LLG	
17	1	Western		3	South Fly District		12	Kiwai Rural LLG	
18	1	Western		3	South Fly District		13	Morehead Rural LLG	
19	1	Western		3	South Fly District		14	Oriomo-Bituri Rural LLG	
20	2	Gulf							
21	2	Gulf		1	Kerema				
22	2	Gulf		1	Kerema		1	Central Kerema Rural	
23	2	Gulf		1	Kerema		2	East Kerema Rural	
24	2	Gulf		1	Kerema		3	Kaintiba Rural	
25	2	Gulf		1	Kerema		4	Kerema Urban	
26	2	Gulf		1	Kerema		5	Kotidanga Rural	
27	2	Gulf		1	Kerema		6	Lakekamu-Tauri Rural	
28	2	Gulf		2	Kikori				
29	2	Gulf		2	Kikori		7	Baimuru Rural	
30	2	Gulf		2	Kikori		8	East Kikori Rural	

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:

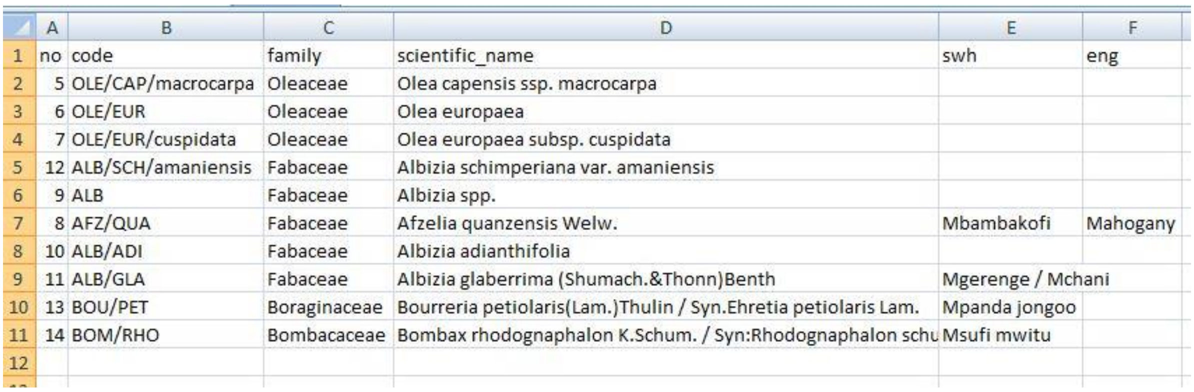
The species list (CSV file) file should be prepared in advance following the formatting requirements indicated below.

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.



	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.

11.4 Sampling point data

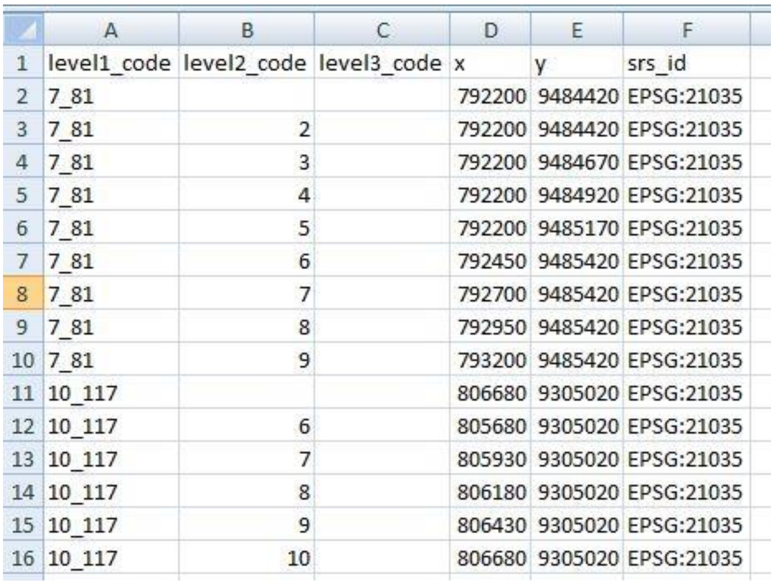
This tab is used to define the list of coordinates of each sample point location. This can be done by importing a CSV file that should be prepared following the formatting indicated below:

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.

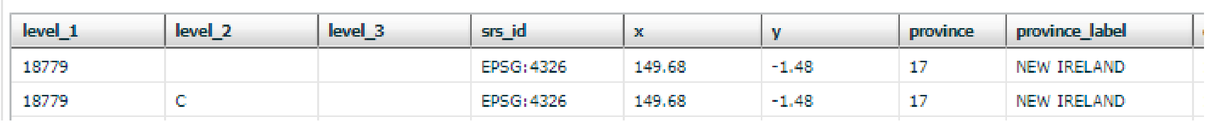
If needed, the file can also be Exported.



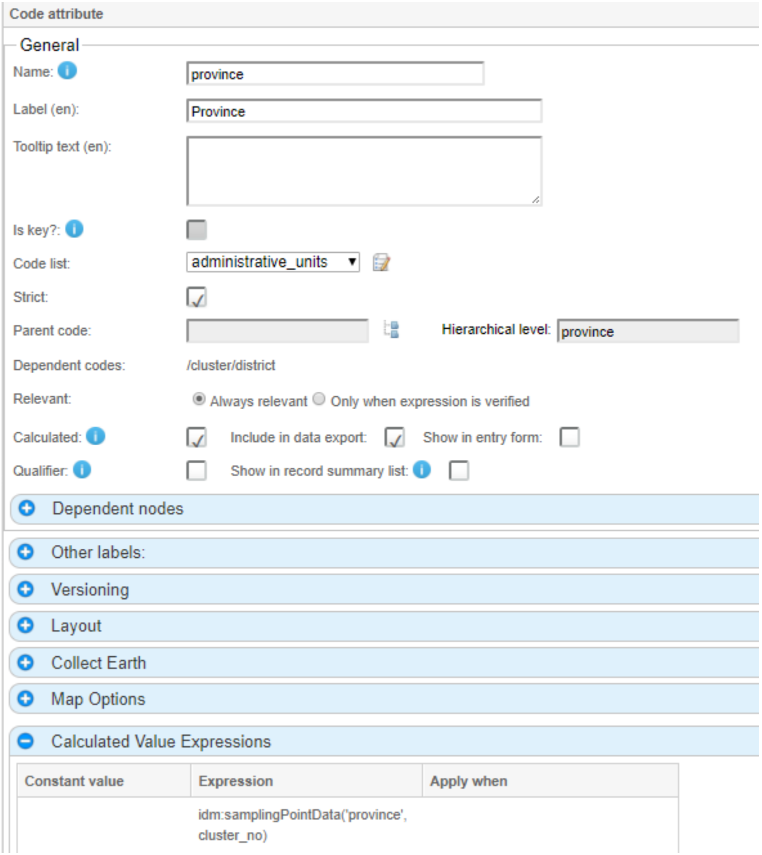
	A	B	C	D	E	F
1	level1_code	level2_code	level3_code	x	y	srs_id
2	7_81			792200	9484420	EPSG:21035
3	7_81	2		792200	9484420	EPSG:21035
4	7_81	3		792200	9484670	EPSG:21035
5	7_81	4		792200	9484920	EPSG:21035
6	7_81	5		792200	9485170	EPSG:21035
7	7_81	6		792450	9485420	EPSG:21035
8	7_81	7		792700	9485420	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 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).



level_1	level_2	level_3	srs_id	x	y	province	province_label
18779			EPSG:4326	149.68	-1.48	17	NEW IRELAND
18779	C		EPSG:4326	149.68	-1.48	17	NEW IRELAND



Code attribute

General

Name:	province
Label (en):	Province
Tooltip text (en):	
Is key?:	□
Code list:	administrative_units
Strict:	☑
Parent code:	Hierarchical level: province
Dependent codes:	/cluster/district
Relevant:	● Always relevant ○ Only when expression is verified
Calculated:	☑ Include in data export: ☑ Show in entry form: □
Qualifier:	□ Show in record summary list: □

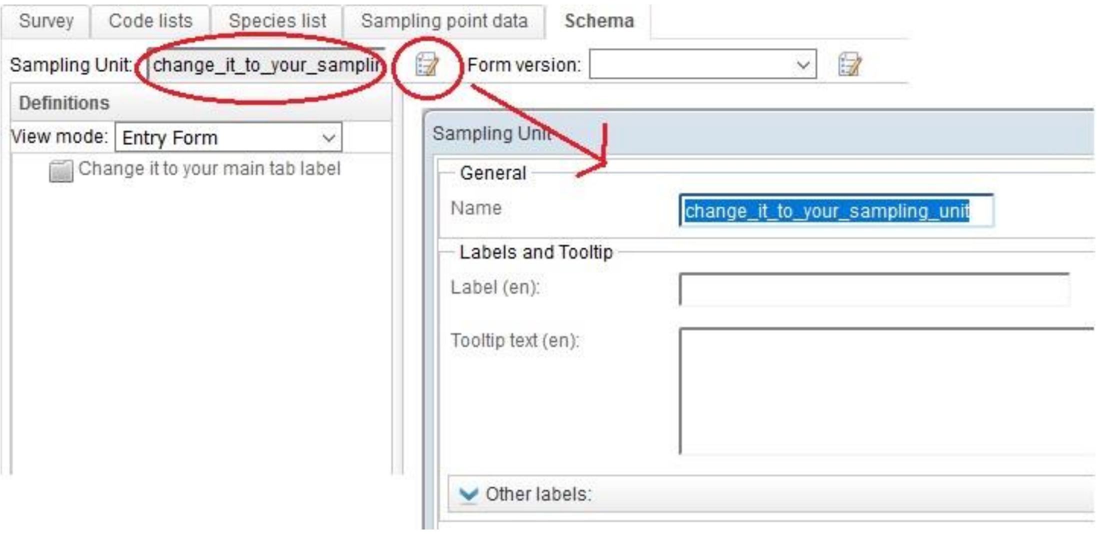
Constant value	Expression	Apply when
	idm:samplingPointData('province', cluster_no)	

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.



Sampling Unit: change_it_to_your_samplin

View mode: Entry Form

Change it to your main tab label

Name: change_it_to_your_sampling_unit

Label (en):

Tooltip text (en):

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

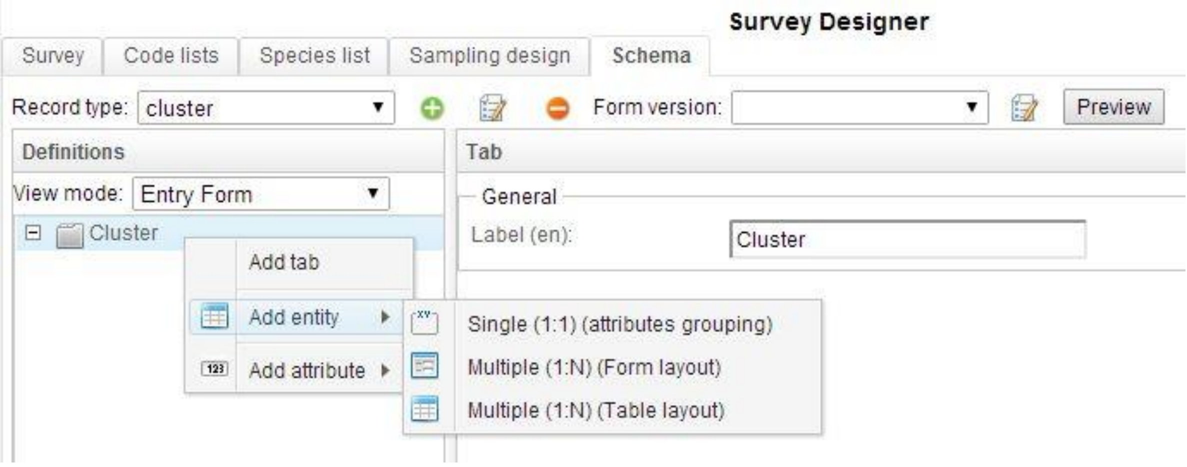
For each of the items that the user wishes to measure in the field, it is necessary to determine whether it is best expressed as an entity or an attribute and of which kind. As an example, a few variables measured at the Cluster level are listed below, indicating the type of considerations needed to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

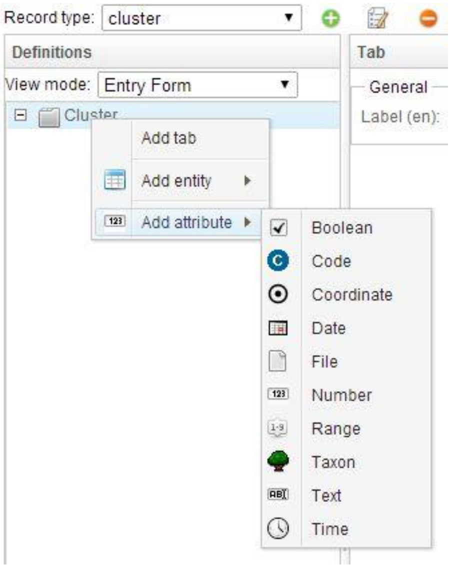
Adding entities:


Record type: cluster

View mode: Entry Form

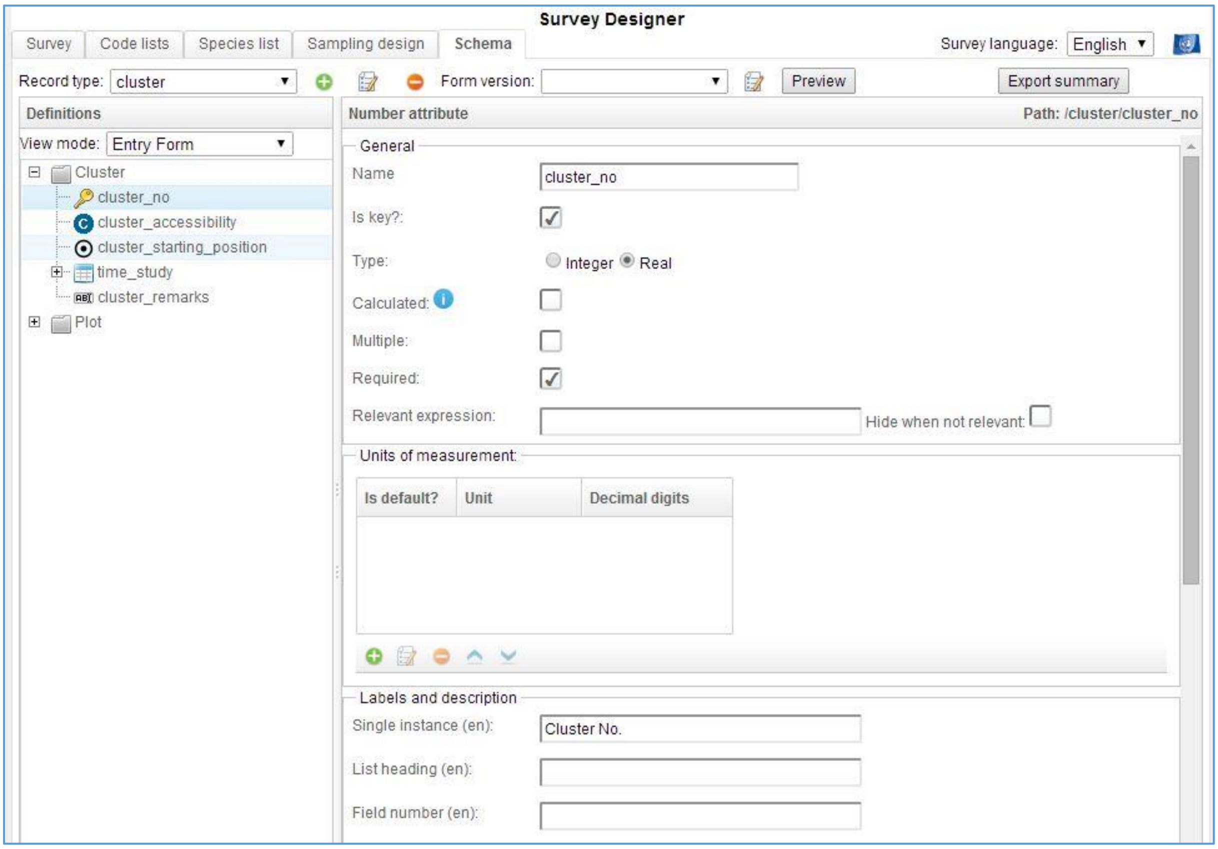
Label (en): Cluster

Adding attributes:


Record type: cluster

View mode: Entry Form

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.


Survey Designer

Number attribute

Path: /cluster/cluster_no

General

Name: cluster_no

Is key?: ☑

Type: ○ Integer ● Real

Calculated: □

Multiple: □

Required: ☑

Relevant expression: Hide when not relevant: □

Units of measurement

Is default?	Unit	Decimal digits

Labels and description

Single instance (en): Cluster No.

List heading (en):

Field number (en):

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.

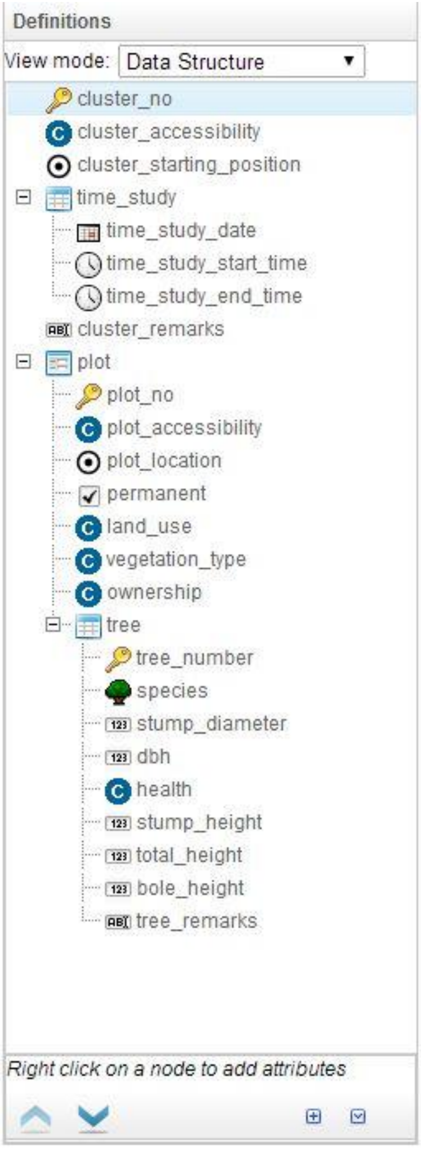
General

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	If checked, the field will be hidden in the data entry form when the

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
File type	Image, Audio, Video, Document Note: in Collect desktop, you can only upload a file, but in Collect Mobile you can capture image, audio recording or video.
- 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 keeping track of since which survey version a specific Entity/Attribute has been added to the schema.

Removed since version	Allows keeping 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

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



Definitions

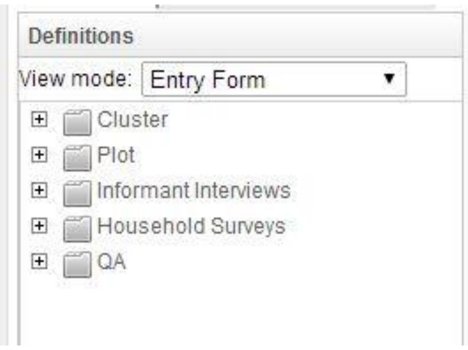
View mode: Data Structure ▼

- cluster_no
- cluster_accessibility
- cluster_starting_position
- time_study
  - time_study_date
  - time_study_start_time
  - time_study_end_time
- cluster_remarks
- plot
  - plot_no
  - plot_accessibility
  - plot_location
  - permanent
  - land_use
  - vegetation_type
  - ownership
  - tree
    - tree_number
    - species
    - stump_diameter
    - dbh
    - health
    - stump_height
    - total_height
    - bole_height
    - tree_remarks

Right click on a node to add attributes

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 [+] 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.



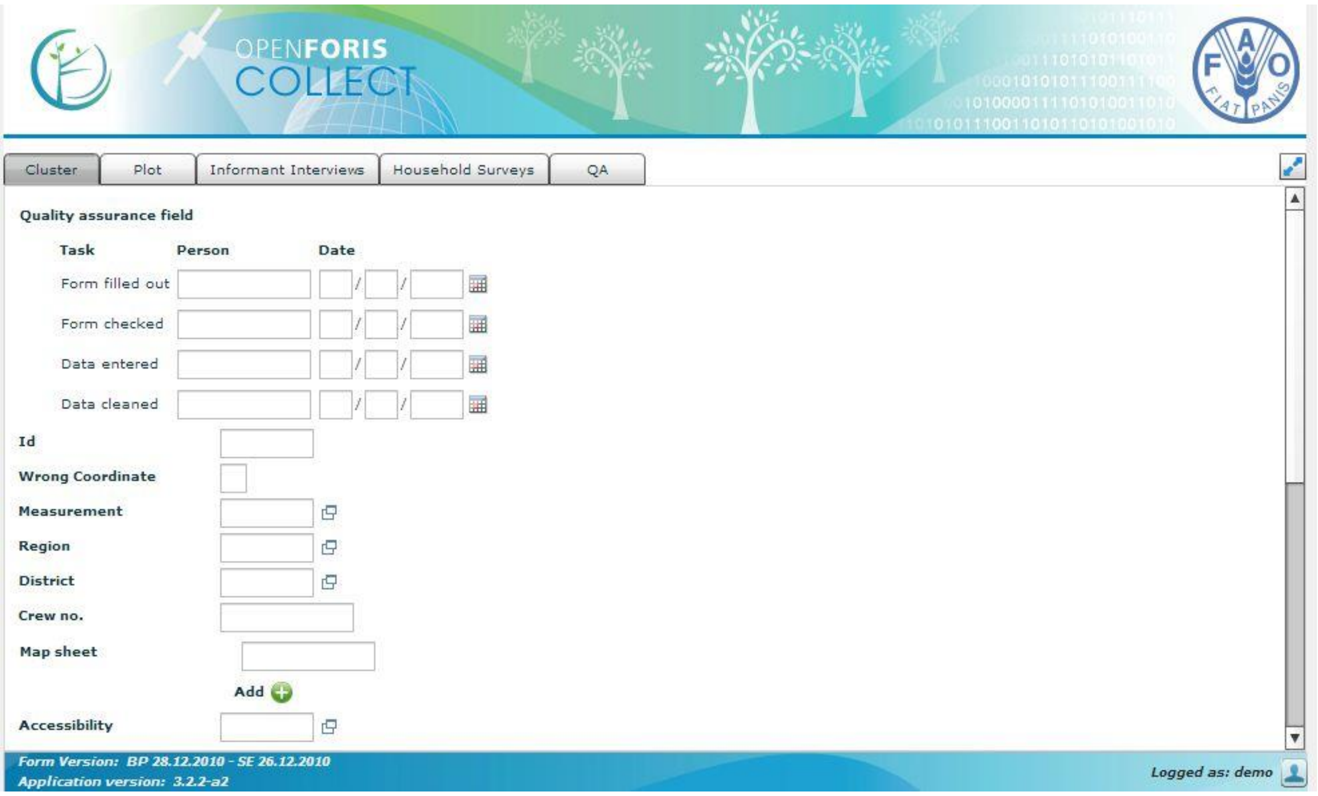
Definitions

View mode: Entry Form ▾

⊞ Cluster
⊞ Plot
⊞ Informant Interviews
⊞ Household Surveys
⊞ QA

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.



OPENFORIS COLLECT
FAO FIAT PANIS
Cluster | Plot | Informant Interviews | Household Surveys | QA

Quality assurance field

Task	Person	Date
Form filled out		/ /
Form checked		/ /
Data entered		/ /
Data cleaned		/ /

Id

Wrong Coordinate □

Measurement

Region

District

Crew no.

Map sheet

Add

Accessibility

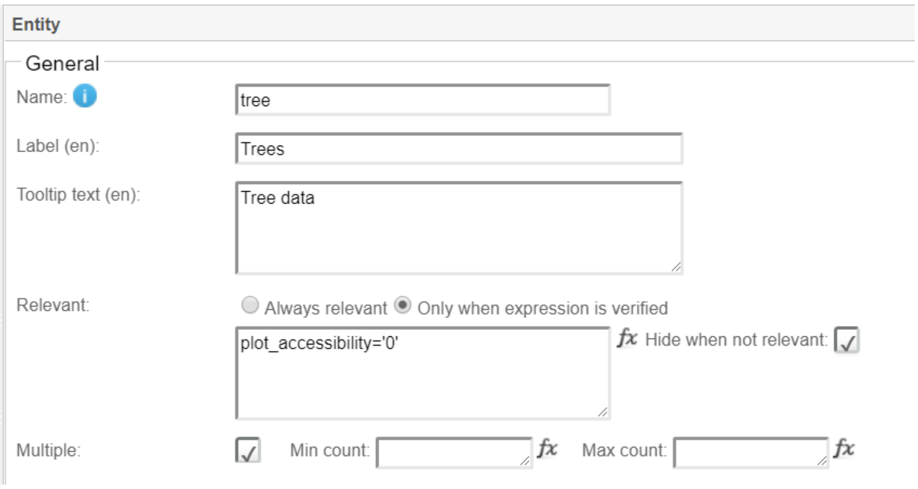
Form Version: BP 28.12.2010 - SE 26.12.2010
Application version: 3.2.2-a2

Logged as: demo

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.



Entity

General

Name: tree

Label (en): Trees

Tooltip text (en): Tree data

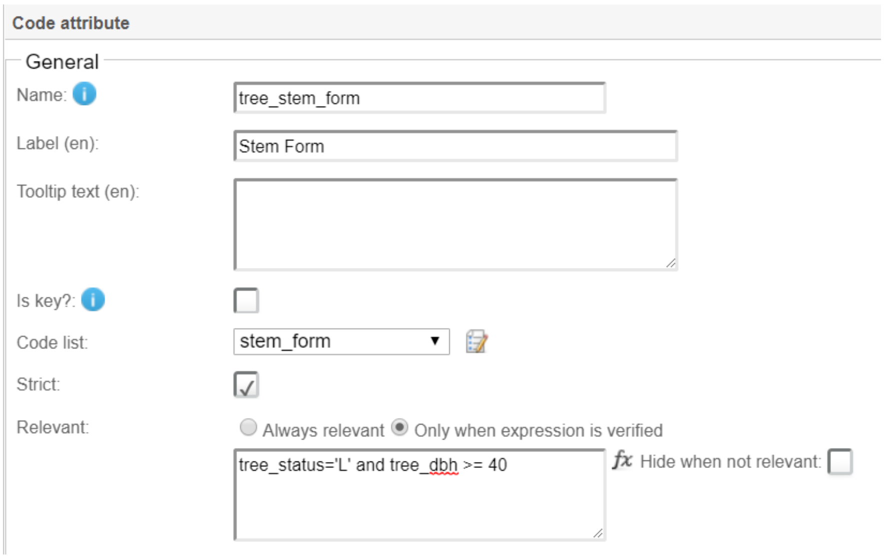
Relevant: ○ Always relevant ● Only when expression is verified

plot_accessibility='0'

fx Hide when not relevant: ☑

Multiple: ☑ Min count: *fx* Max count: *fx*

2) Attribute "stem form" is available if a tree is living (*tree_status='L'*) and DBH more or equal to 40cm (*tree_dbh>=40*).



Code attribute

General

Name: tree_stem_form

Label (en): Stem Form

Tooltip text (en):

Is key?: □

Code list: stem_form ▼

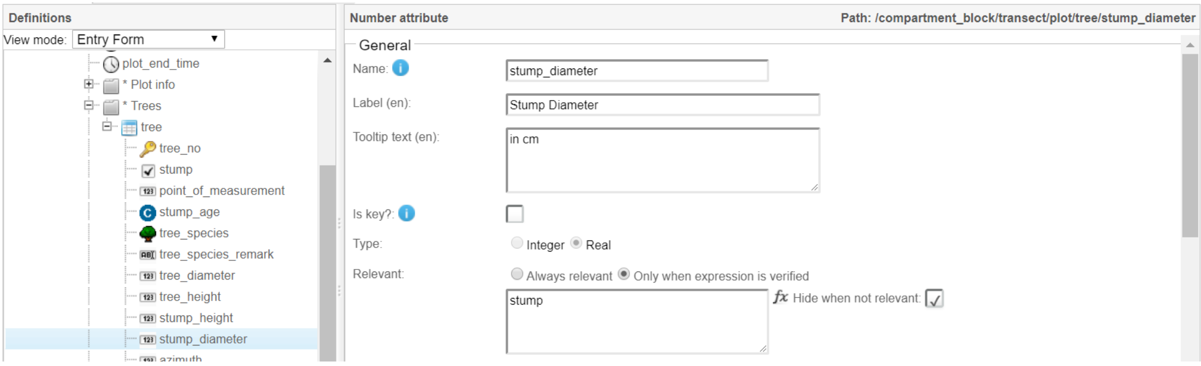
Strict: ☑

Relevant: ○ Always relevant ● Only when expression is verified

tree_status='L' and tree_dbh >= 40

fx Hide when not relevant: □

3) Trees and stumps are collected in the same form. Stump diameter is only recorded for stumps (i.e. attribute 'stump' is TRUE).



Definitions

View mode: Entry Form ▼

Number attribute Path: /compartment_block/transect/plot/tree/stump_diameter

General

Name: stump_diameter

Label (en): Stump Diameter

Tooltip text (en): in cm

Is key?: □

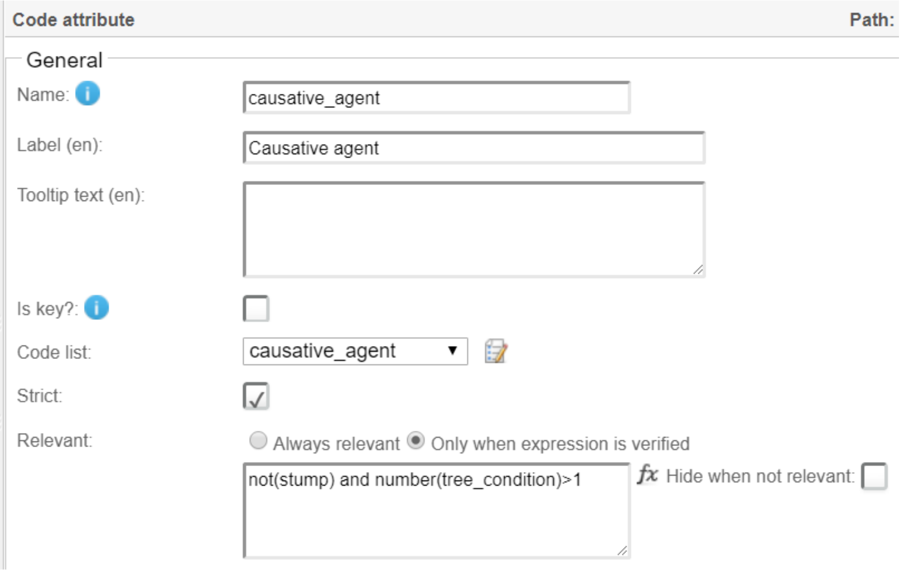
Type: ○ Integer ● Real

Relevant: ○ Always relevant ● Only when expression is verified

stump

fx Hide when not relevant: ☑

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).



Code attribute | Path:

General

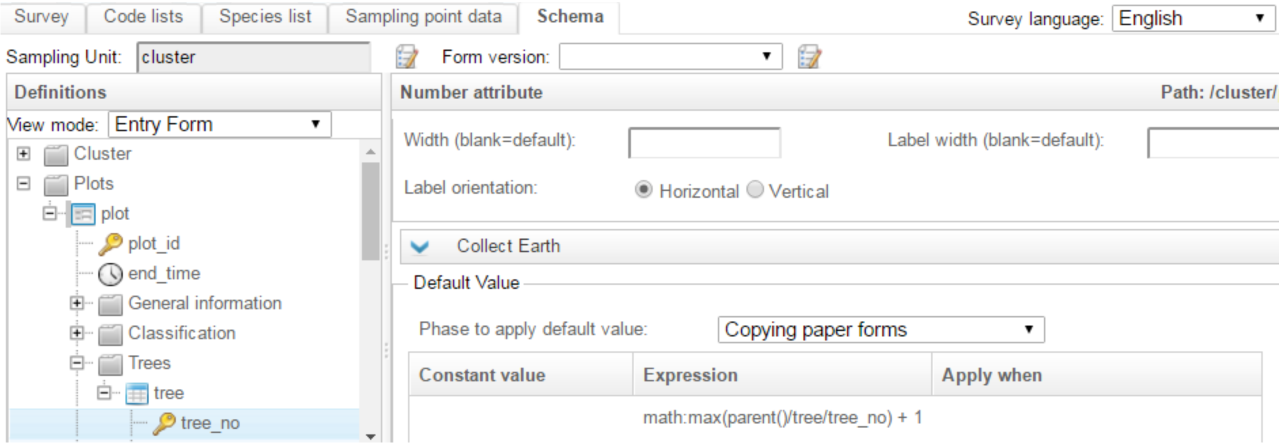
Name:	causative_agent
Label (en):	Causative agent
Tooltip text (en):	
Is key?:	□
Code list:	causative_agent
Strict:	☑
Relevant:	○ Always relevant ● Only when expression is verified not(stump) and number(tree_condition)>1 Hide when not relevant: □

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 *cluster/plot/tree* and we need to get tree numbers automatically.

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



Survey language: English

Sampling Unit: cluster

Form version:

View mode: Entry Form

Number attribute

Path: /cluster/

Width (blank=default):

Label width (blank=default):

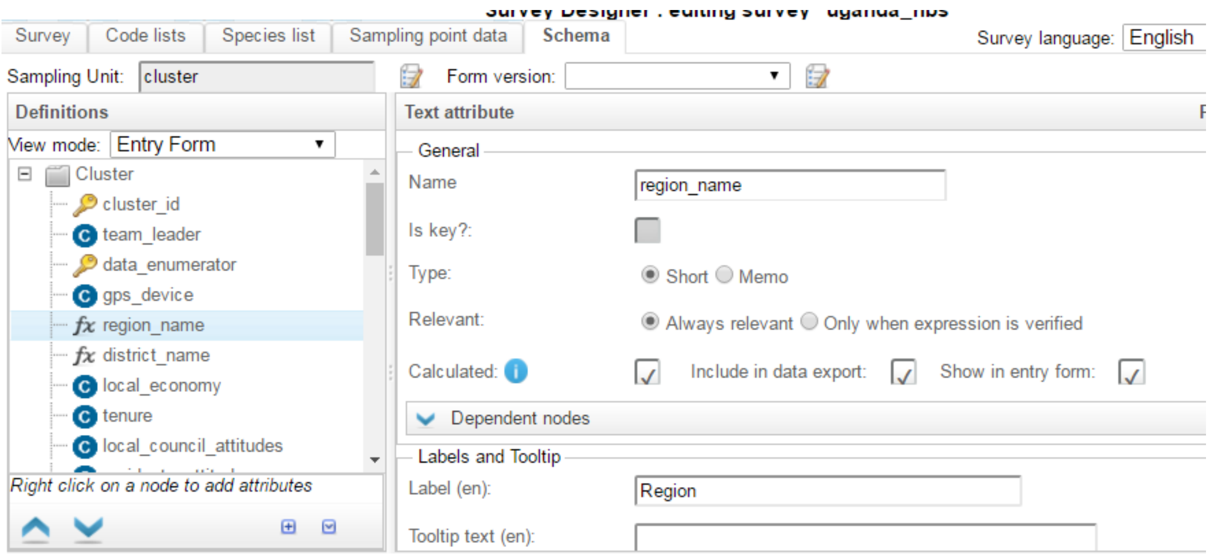
Label orientation: ● Horizontal ○ Vertical

Default Value

Phase to apply default value: Copying paper forms

Constant value	Expression	Apply when
	math:max(parent()/tree/tree_no) + 1	

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'.



Survey language: English

Sampling Unit: cluster

Form version:

View mode: Entry Form

Text attribute

General

Name: region_name

Is key?: □

Type: ● Short ○ Memo

Relevant: ● Always relevant ○ Only when expression is verified

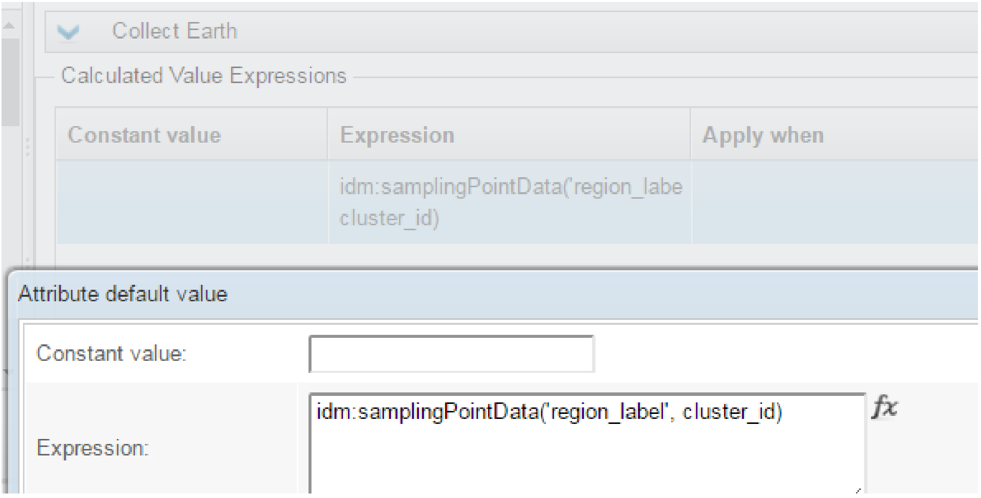
Calculated: ☑ Include in data export: ☑ Show in entry form: ☑

Labels and Tooltip

Label (en): Region

Tooltip text (en):

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



Collect Earth

Calculated Value Expressions

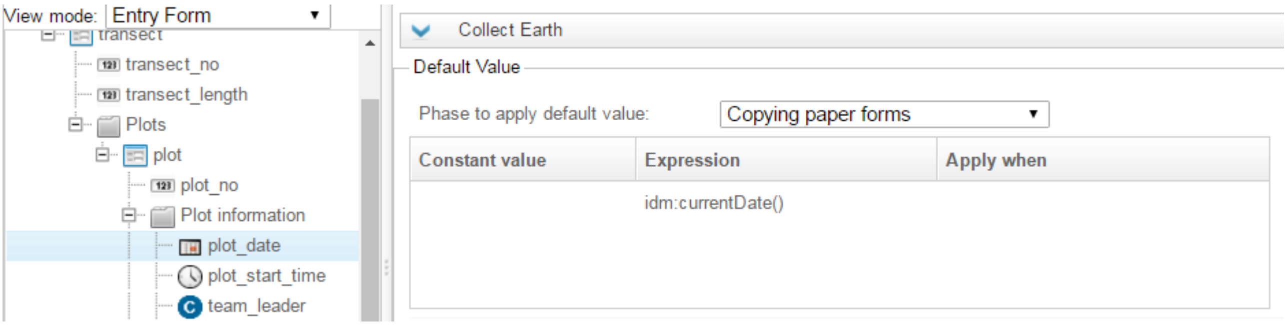
Constant value	Expression	Apply when
	idm:samplingPointData('region_labe cluster_id)	

Attribute default value

Constant value:

Expression: idm:samplingPointData('region_label', cluster_id)

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



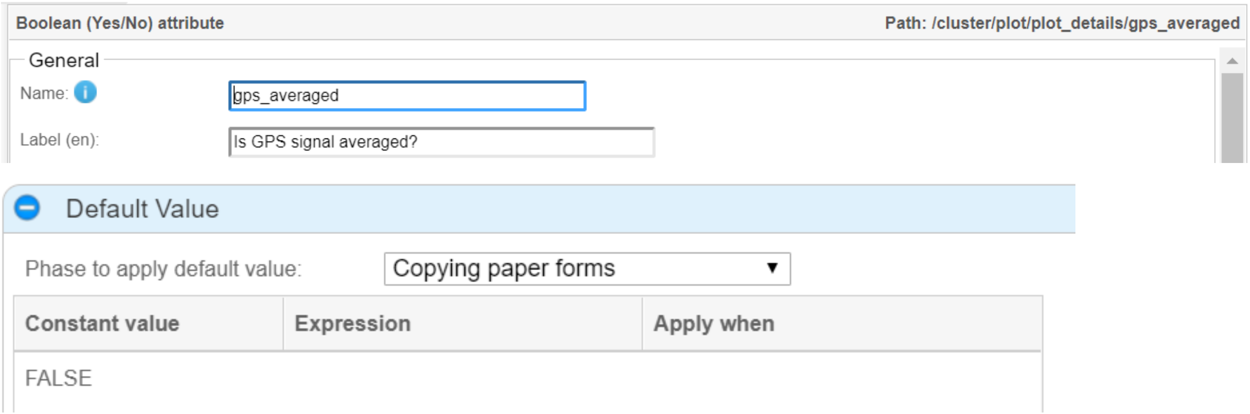
Default Value

Phase to apply default value: Copying paper forms

Constant value	Expression	Apply when
	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.



Boolean (Yes/No) attribute   Path: /cluster/plot/plot_details/gps_averaged

General

Name: gps_averaged

Label (en): Is GPS signal averaged?

Default Value

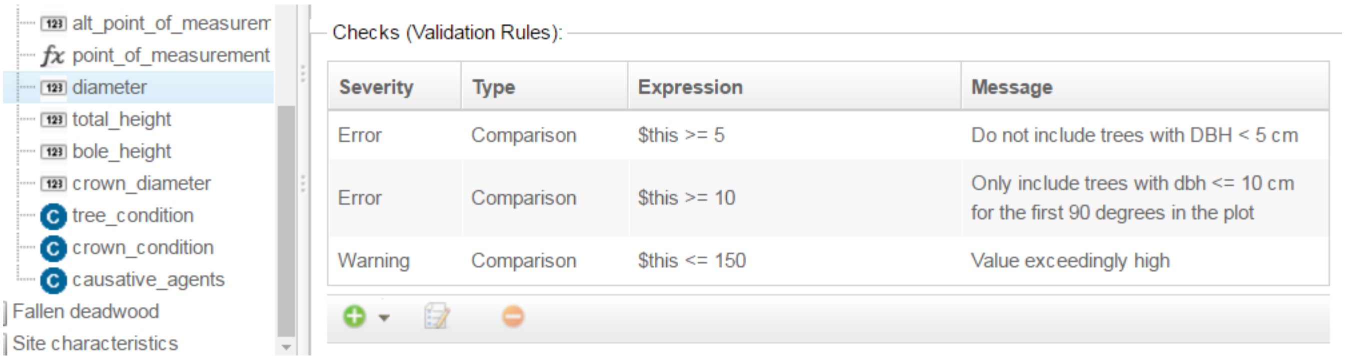
Phase to apply default value: Copying paper forms

Constant value	Expression	Apply when
FALSE		

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.

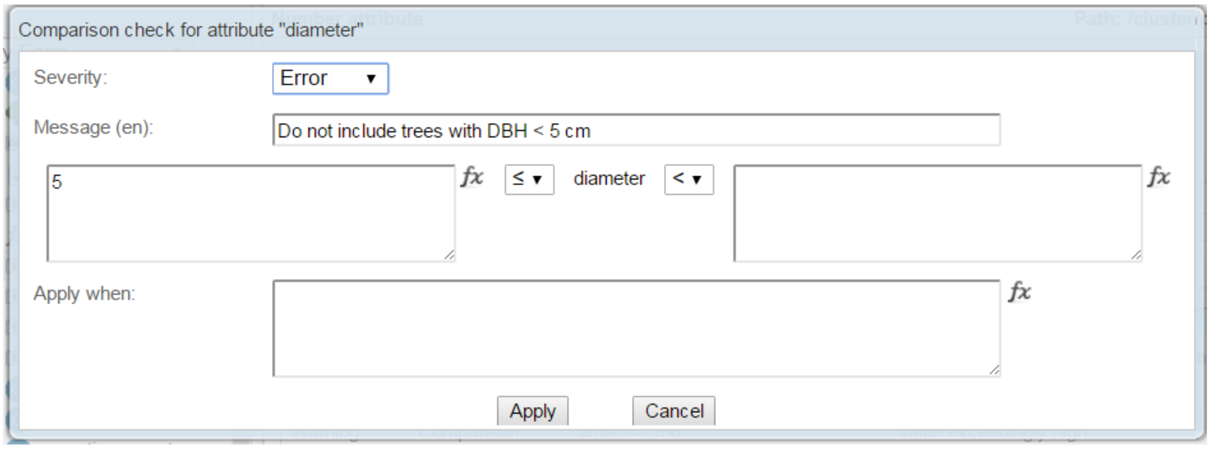
1) Tree diameter (DBH) check. Diameter is a numeric attribute (real).



Checks (Validation Rules):

Severity	Type	Expression	Message
Error	Comparison	$this >= 5	Do not include trees with DBH < 5 cm
Error	Comparison	$this >= 10	Only include trees with dbh <= 10 cm for the first 90 degrees in the plot
Warning	Comparison	$this <= 150	Value exceedingly high

And the first rule.



Comparison check for attribute "diameter"

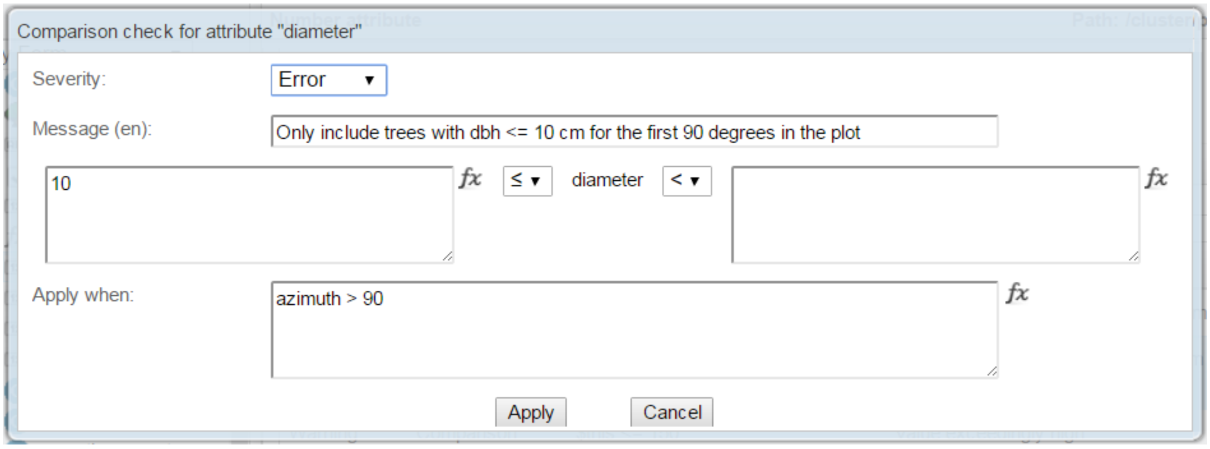
Severity: Error

Message (en): Do not include trees with DBH < 5 cm

5 ≤ diameter <

Apply when:

The second rule.



Comparison check for attribute "diameter"

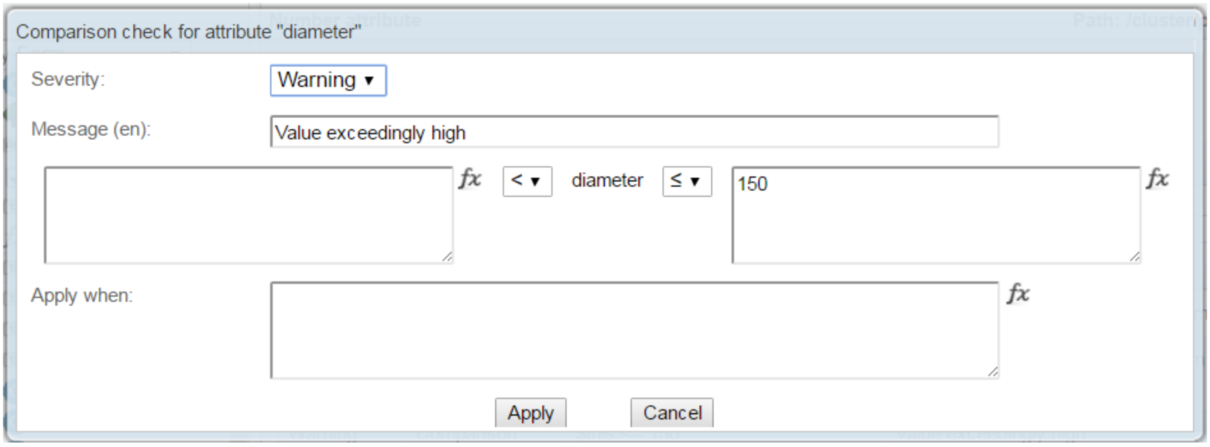
Severity: Error

Message (en): Only include trees with dbh <= 10 cm for the first 90 degrees in the plot

10 ≤ diameter <

Apply when: azimuth > 90

And the third rule.



Comparison check for attribute "diameter"

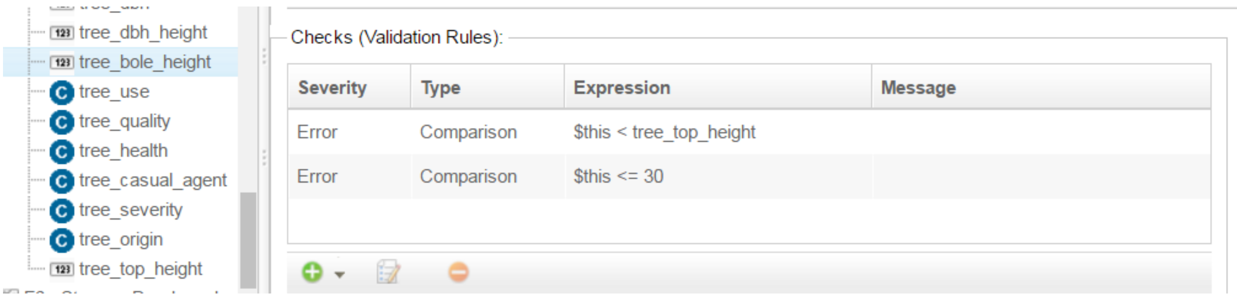
Severity: Warning ▾

Message (en): Value exceedingly high

Apply when:

2) Bole height validation rules are as follows:

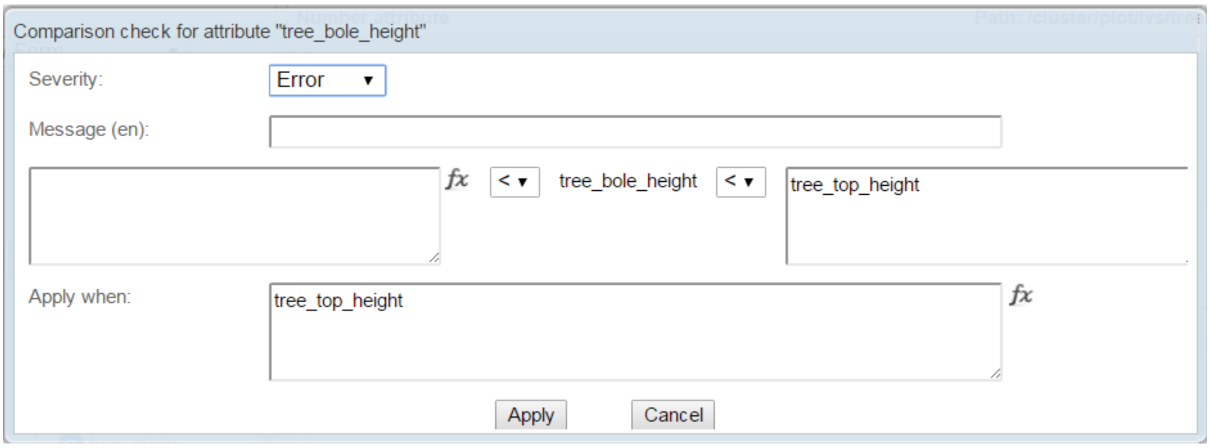
- Should be always less than tree top height (‘tree_top_height’) if top height is given, and
- Should be always less than 30 m.



Checks (Validation Rules):

Severity	Type	Expression	Message
Error	Comparison	$this < tree_top_height	
Error	Comparison	$this <= 30	

And in the edit mode this look as follows:



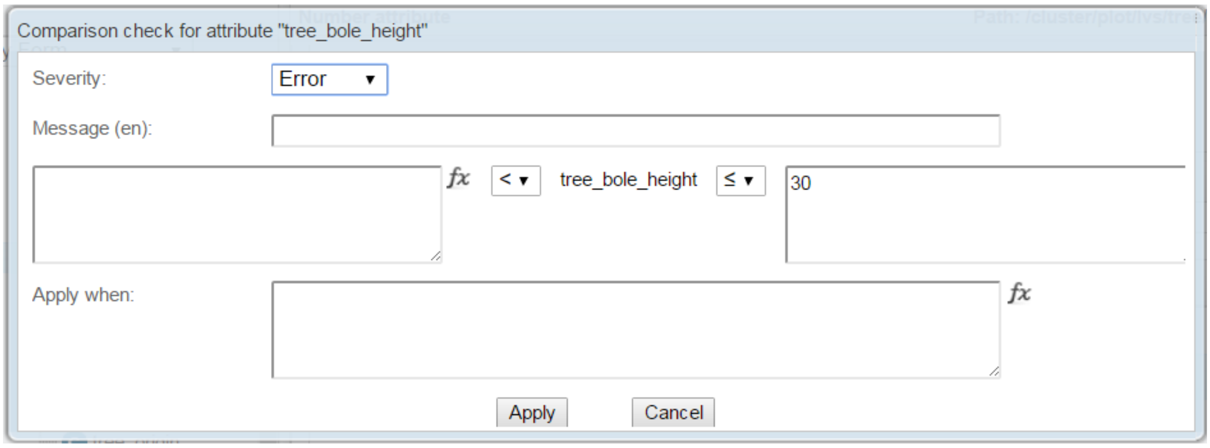
Comparison check for attribute "tree_bole_height"

Severity: Error ▾

Message (en):

Apply when: tree_top_height

And the second rule.

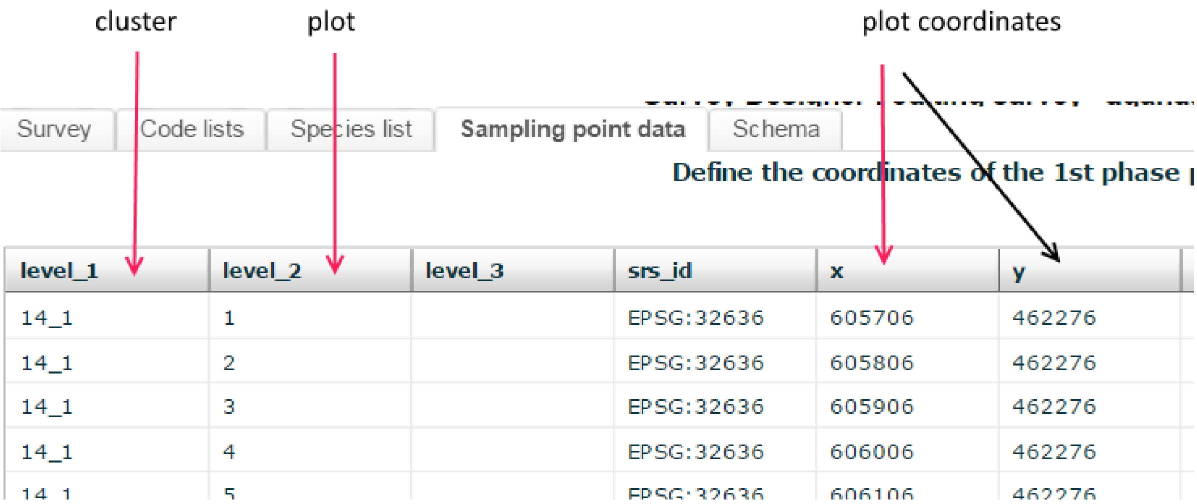


Comparison check for attribute "tree_bole_height"

Severity:	Error
Message (en):	
≤	30
Apply when:	

3) Plot location check for inputted coordinates.

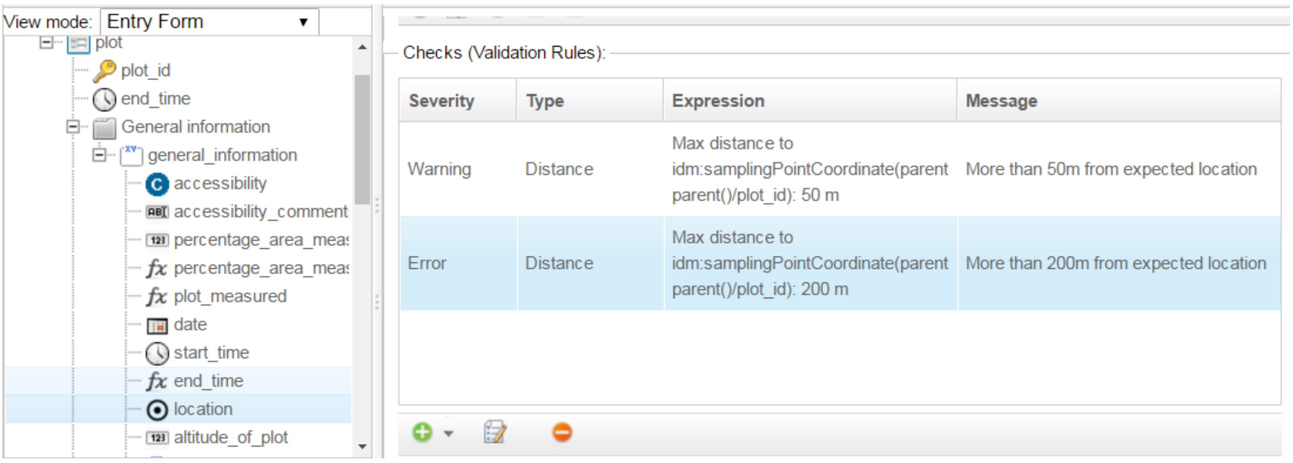
In ‘Sampling point data’ there is given a list of clusters, plots and plot coordinates, as follows.



Define the coordinates of the 1st phase

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

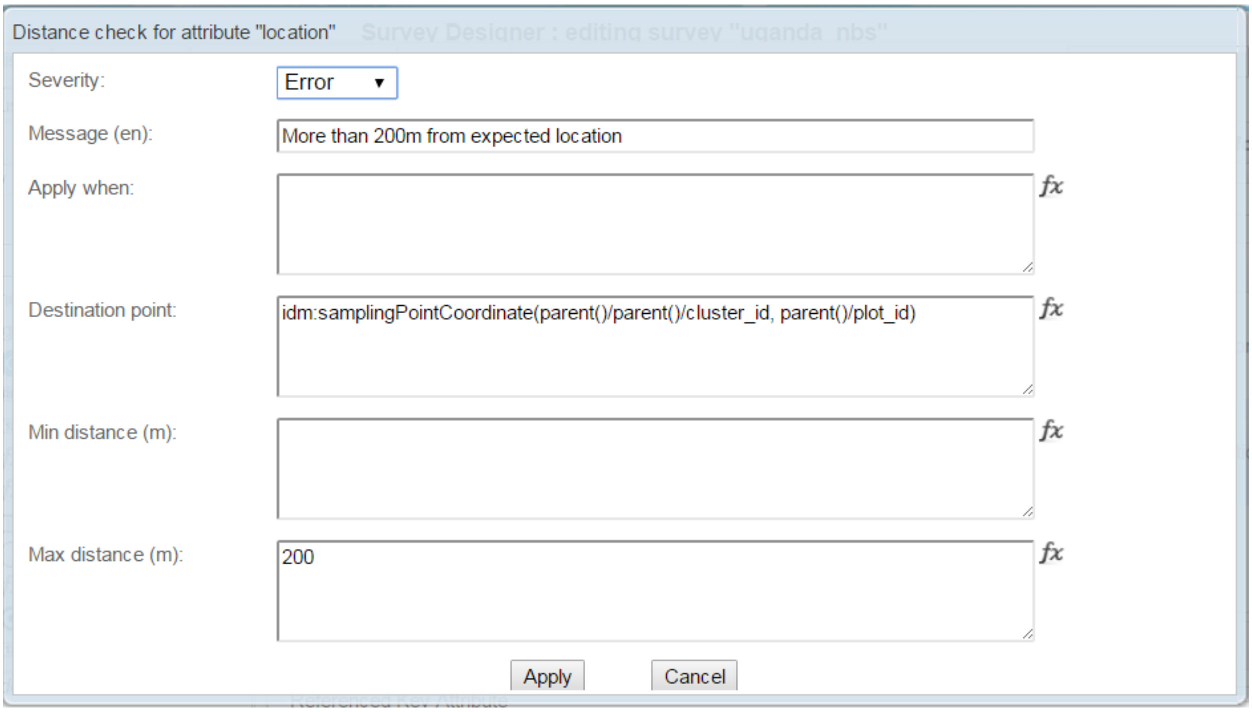
And we need validation rules for attribute ‘location’, i.e. for plot coordinates. Please notice that the attribute ‘location’ belongs to entity ‘general_information’ which is under ‘plot’ in this survey.



Checks (Validation Rules):

Severity	Type	Expression	Message
Warning	Distance	Max distance to idm:samplingPointCoordinate(parent parent()/plot_id): 50 m	More than 50m from expected location
Error	Distance	Max distance to idm:samplingPointCoordinate(parent parent()/plot_id): 200 m	More than 200m from expected location

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


Distance check for attribute "location"

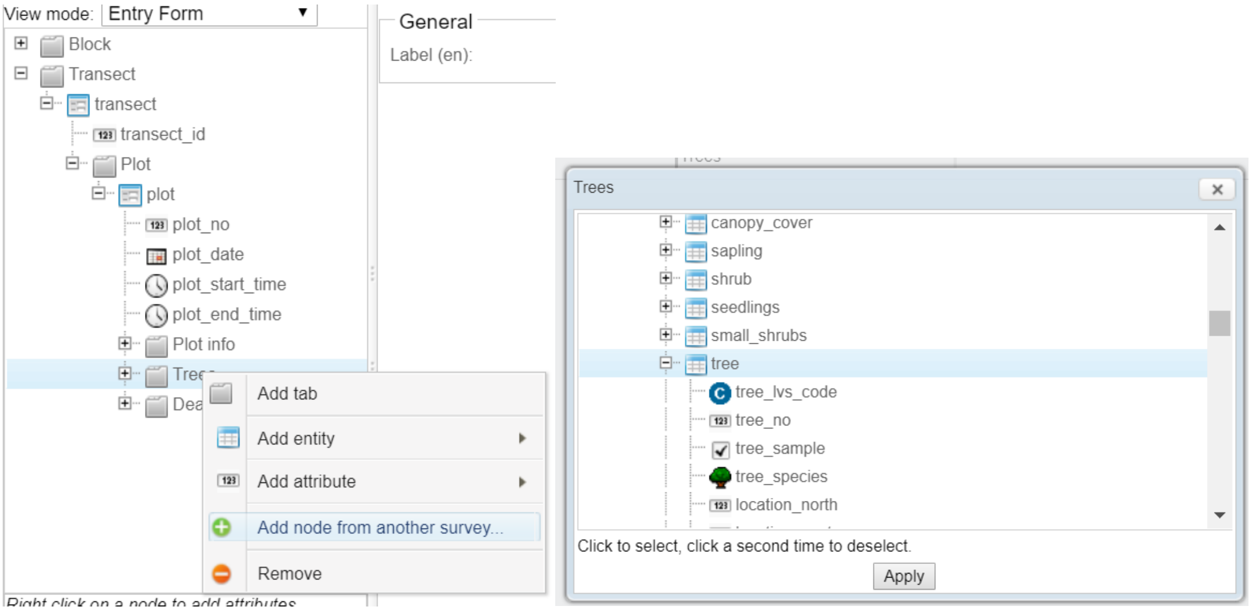
Severity:	Error
Message (en):	More than 200m from expected location
Apply when:	
Destination point:	idm:samplingPointCoordinate(parent()/parent()/cluster_id, parent()/plot_id)
Min distance (m):	
Max distance (m):	200

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).

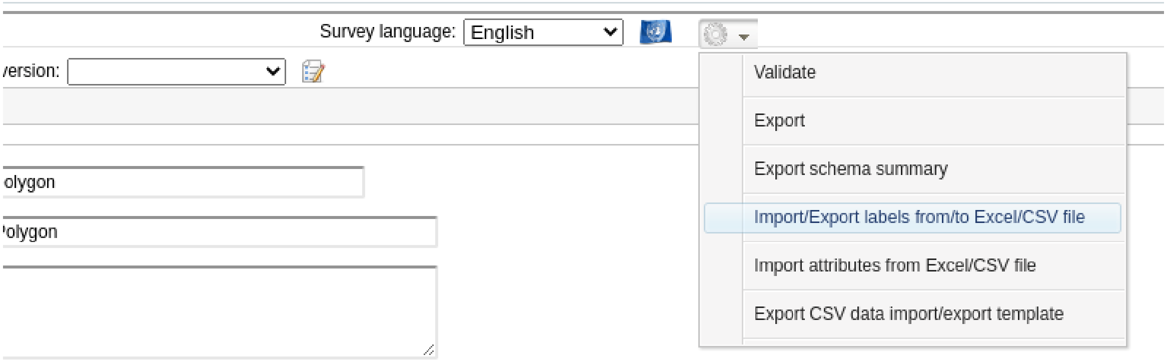
The "another survey" can be published or temporary: if there is only the published copy of the survey, that one will be used, otherwise the survey name will have a suffix "(temporary)" and that version of the survey will be used.



Click to select, click a second time to deselect.

11.5.6 IMPORTING NEW LABELS

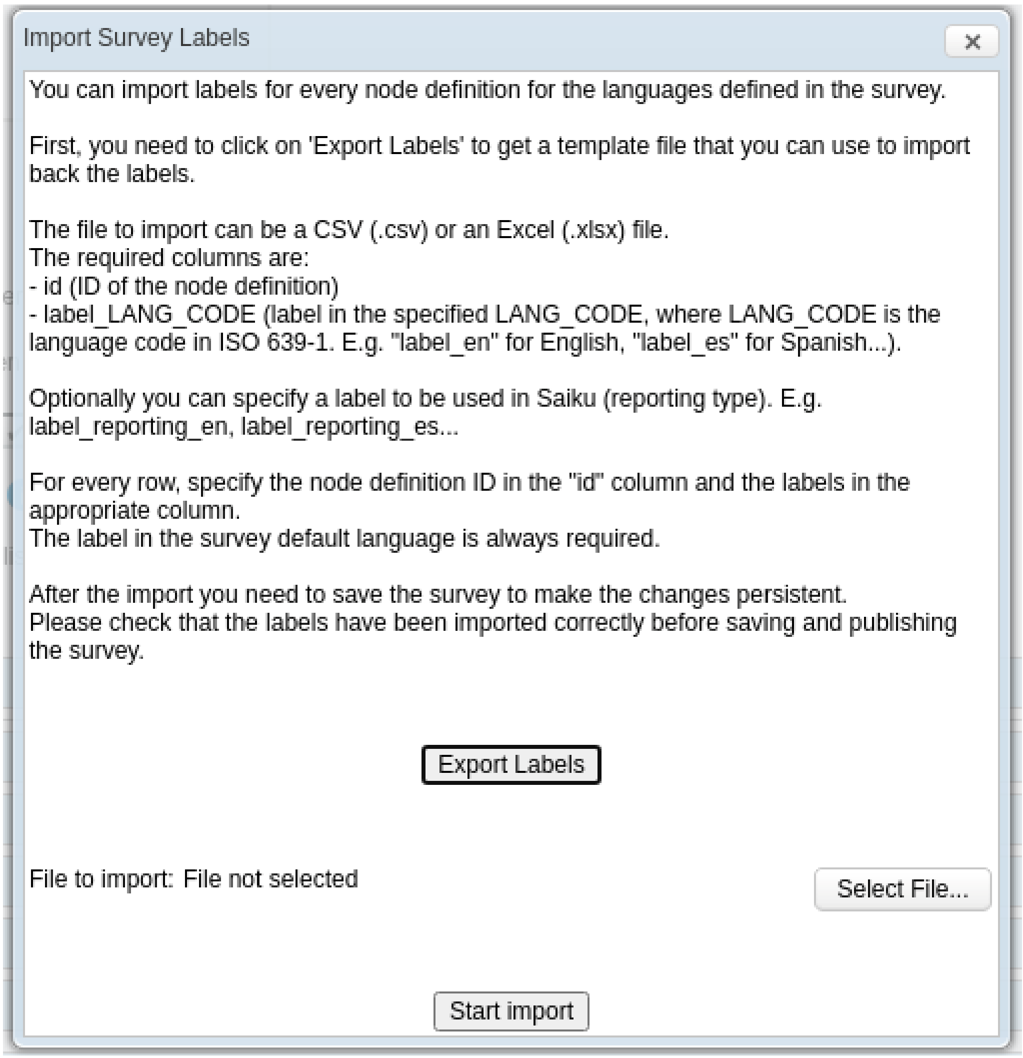
In Collect v4.0.12 there is a new feature, accessible from the top right gear button of the Survey Edit page, in the Survey Designer, it's called **Import/Export labels from/to Excel/CSV file**, as follows:



Survey language: English

/ersion:
olygon
'olygon
Validate
Export
Export schema summary
Import/Export labels from/to Excel/CSV file
Import attributes from Excel/CSV file
Export CSV data import/export template

In the popup that appears, there are some instructions on how to prepare a file that you can use to import new labels into your survey, or to replace the existing ones.



Import Survey Labels

You can import labels for every node definition for the languages defined in the survey.

First, you need to click on 'Export Labels' to get a template file that you can use to import back the labels.

The file to import can be a CSV (.csv) or an Excel (.xlsx) file.
The required columns are:
- id (ID of the node definition)
- label_LANG_CODE (label in the specified LANG_CODE, where LANG_CODE is the language code in ISO 639-1. E.g. "label_en" for English, "label_es" for Spanish...).

Optionally you can specify a label to be used in Saiku (reporting type). E.g. label_reporting_en, label_reporting_es...

For every row, specify the node definition ID in the "id" column and the labels in the appropriate column.
The label in the survey default language is always required.

After the import you need to save the survey to make the changes persistent.
Please check that the labels have been imported correctly before saving and publishing the survey.

File to import: File not selected

You can also export the current labels as an Excel file, modify them and import it back into Collect to as edited labels.

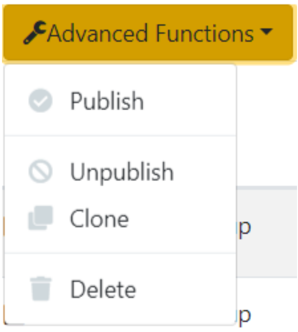
Please consider that in this way you can import only attributes' and entities' labels; **tabs' labels won't be exported/imported!**

If you need to import labels for a new language, add the language(s) first into the survey, then export the labels, edit the Excel file and import all labels back into the survey.

Note: If you use a CSV file for importing the labels, it must be in UTF-8 or otherwise special characters won't be read in correctly.

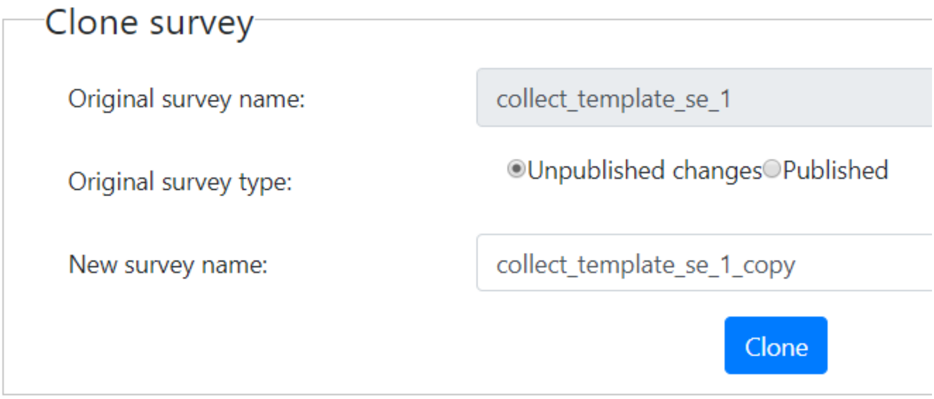
11.7 Advanced functions

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



These are as follows

1. **Publish:** Publish survey for Data Management. Includes any changes you have put into the survey since it was last published.
2. **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.
3. **Clone:** Make a copy of an existing survey.



Clone survey

Original survey name: collect_template_se_1

Original survey type: ● Unpublished changes ○ Published

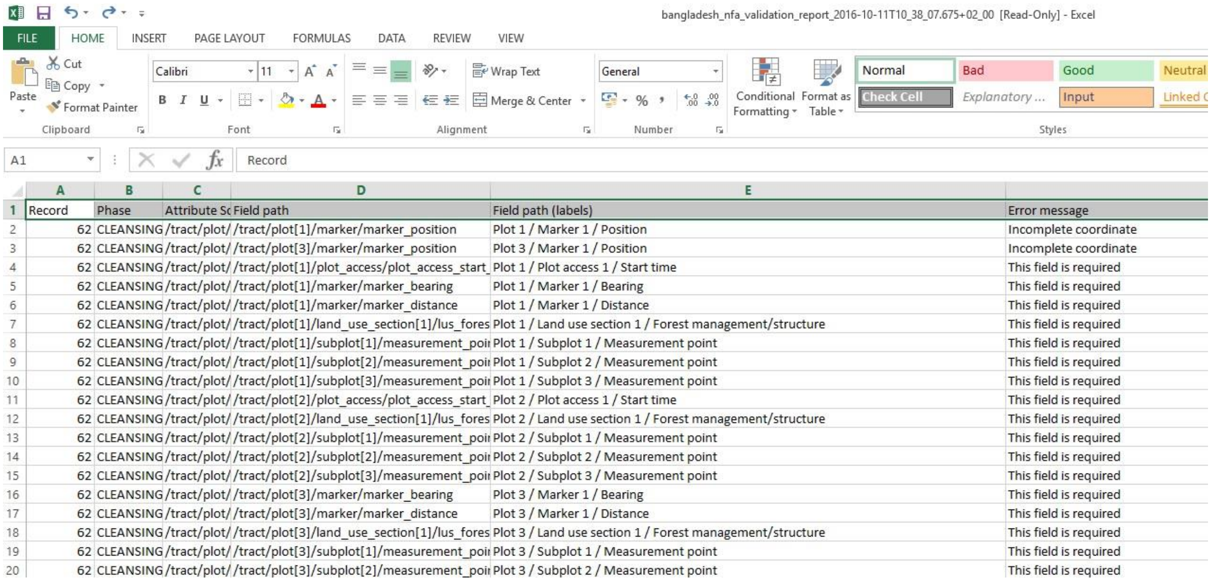
New survey name: collect_template_se_1_copy

4. Delete the selected survey: if the survey has been published, you will be warned that deleting the survey will delete even al 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.



	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 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.

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, 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.

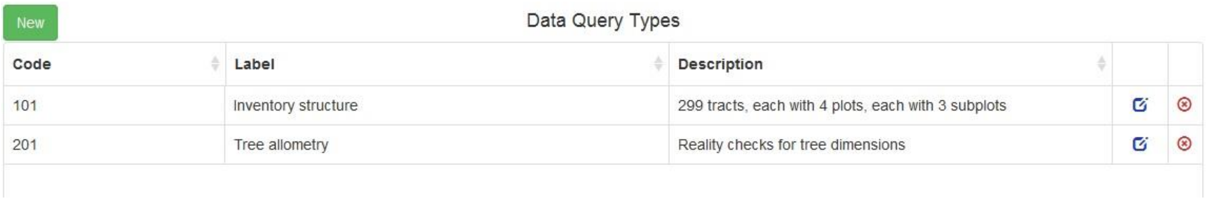
12.2 Data cleansing workflow

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.

12.2.1Data 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

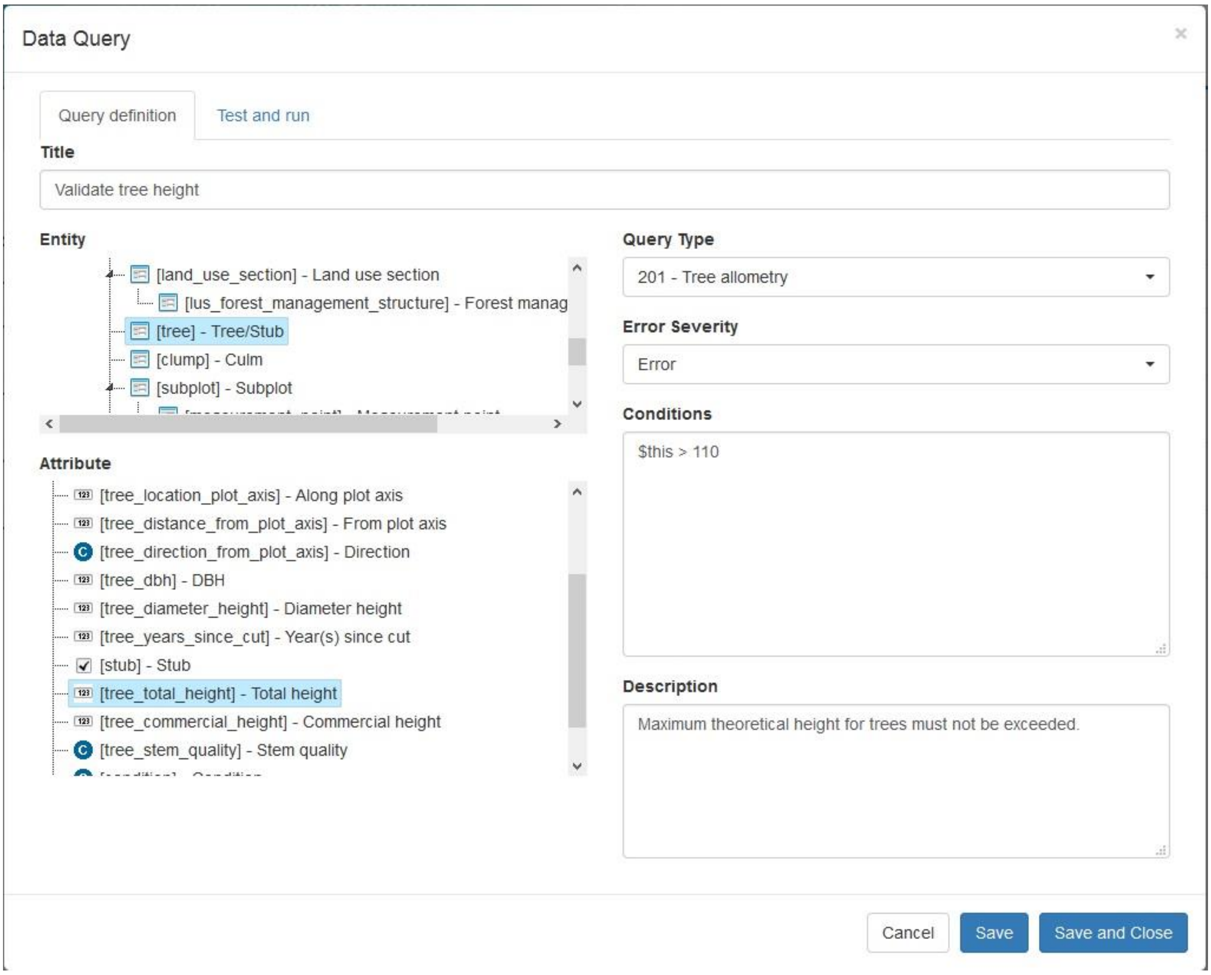


Data Query Types

Code	Label	Description		
101	Inventory structure	299 tracts, each with 4 plots, each with 3 subplots	☑	⊗
201	Tree allometry	Reality checks for tree dimensions	☑	⊗

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



Title: Validate tree height

Entity:
- [land_use_section] - Land use section
  - [lus_forest_management_structure] - Forest manag
- [tree] - Tree/Stub
- [clump] - Culm
- [subplot] - Subplot

Attribute:
- [tree_location_plot_axis] - Along plot axis
- [tree_distance_from_plot_axis] - From plot axis
- [tree_direction_from_plot_axis] - Direction
- [tree_dbh] - DBH
- [tree_diameter_height] - Diameter height
- [tree_years_since_cut] - Year(s) since cut
- ☑ [stub] - Stub
- [tree_total_height] - Total height
- [tree_commercial_height] - Commercial height
- [tree_stem_quality] - Stem quality

Query Type: 201 - Tree allometry

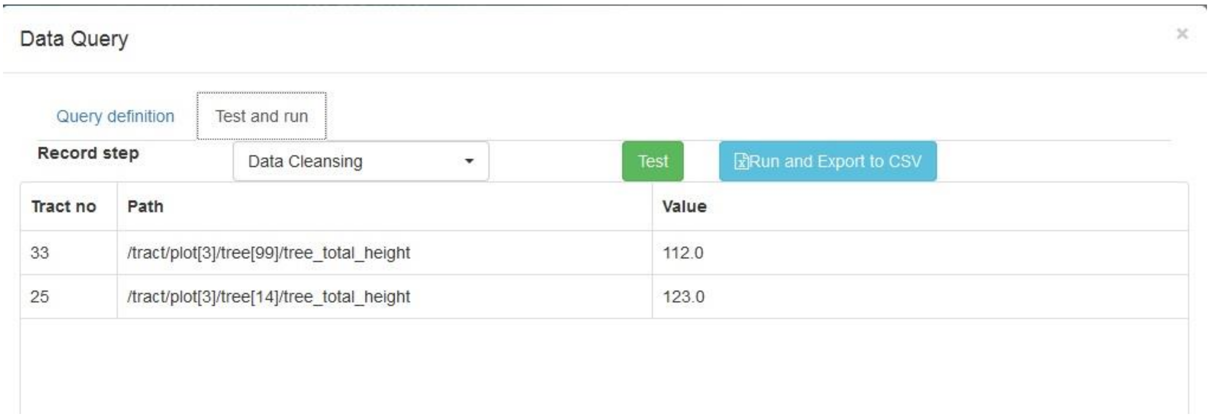
Error Severity: Error

Conditions: $this > 110

Description: Maximum theoretical height for trees must not be exceeded.

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.

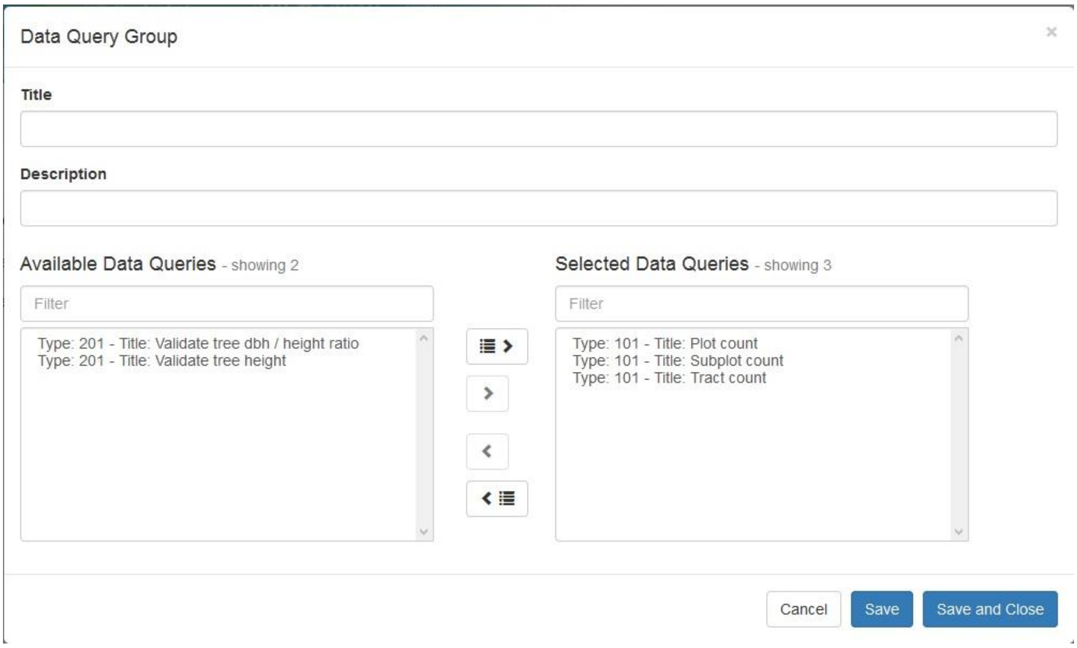
To see if a Query works, go to the “Test and Run” tab. Select the appropriate record step (the phase of data management at which your data is now, usually Data Cleansing now) and click “Run”. A summary of successfully located records appears on the window.



Record step: Data Cleansing

Tract no	Path	Value
33	/tract/plot[3]/tree[99]/tree_total_height	112.0
25	/tract/plot[3]/tree[14]/tree_total_height	123.0

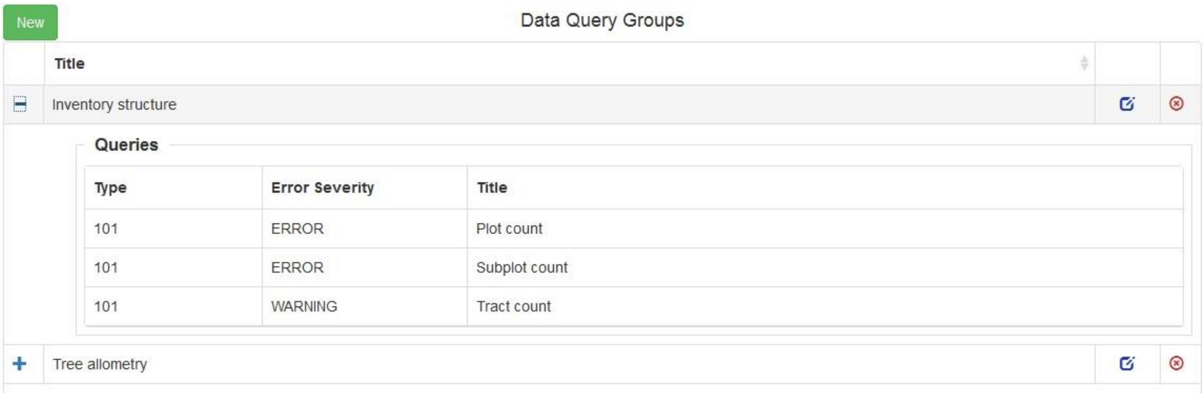
12.2.4GROUP QUERIES INTO DATA QUERY GROUPS



Available Data Queries:
Type: 201 - Title: Validate tree dbh / height ratio
Type: 201 - Title: Validate tree height

Selected Data Queries:
Type: 101 - Title: Plot count
Type: 101 - Title: Subplot count
Type: 101 - Title: Tract count

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.



Data Query Groups

	Title
−	Inventory structure
+	Tree allometry

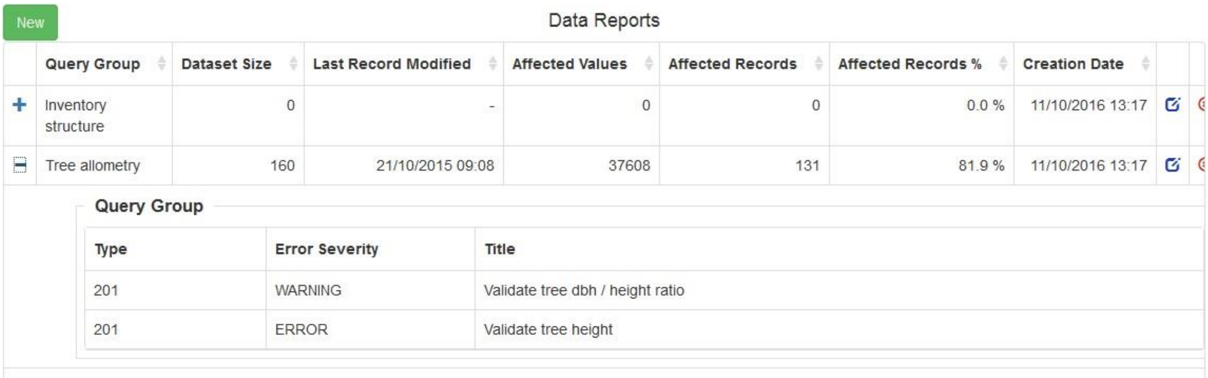
Queries

Type	Error Severity	Title
101	ERROR	Plot count
101	ERROR	Subplot count
101	WARNING	Tract count

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.



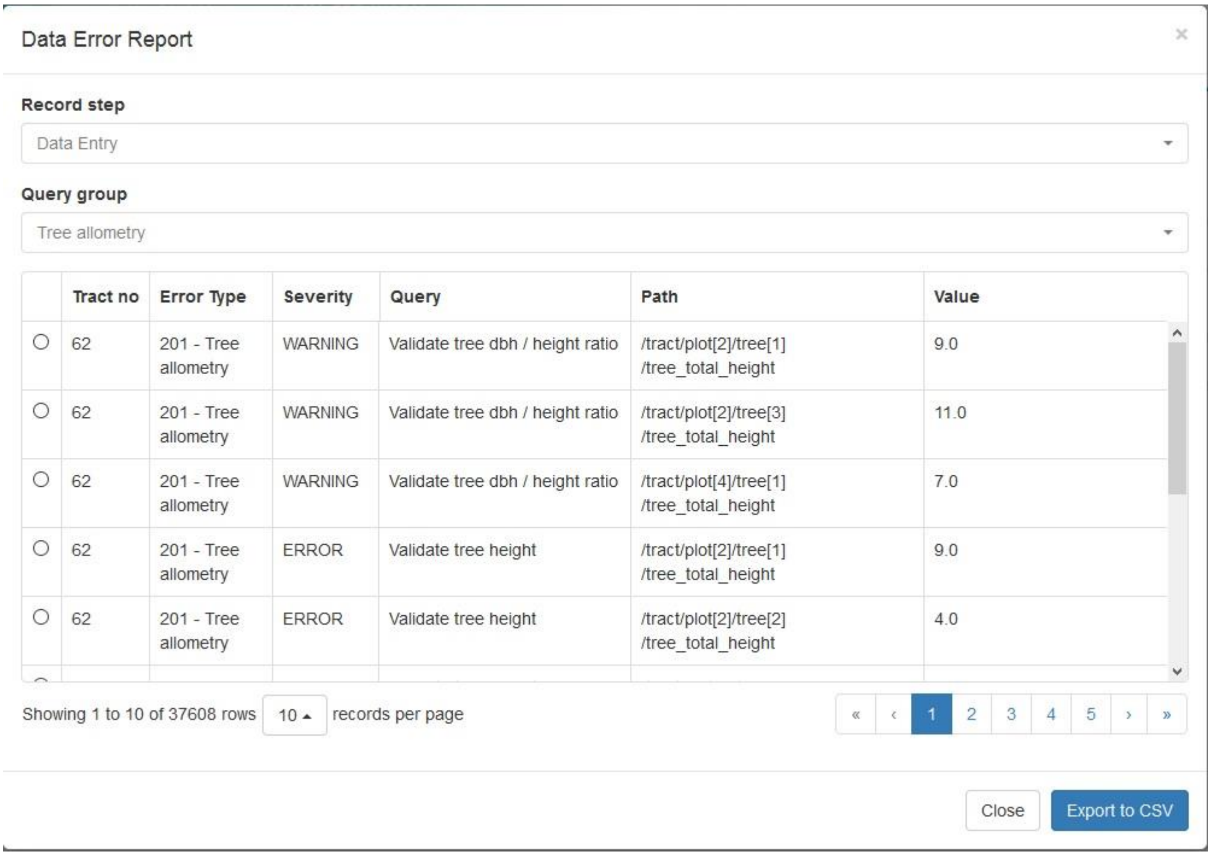
Data Reports

	Query Group	Dataset Size	Last Record Modified	Affected Values	Affected Records	Affected Records %	Creation Date
+	Inventory structure	0	-	0	0	0.0 %	11/10/2016 13:17
−	Tree allometry	160	21/10/2015 09:08	37608	131	81.9 %	11/10/2016 13:17

Query Group

Type	Error Severity	Title
201	WARNING	Validate tree dbh / height ratio
201	ERROR	Validate tree height

To view the contents of the Data Report, click on the blue edit sign.



Data Error Report ×

Record step

Data Entry

Query group

Tree allometry

	Tract no	Error Type	Severity	Query	Path	Value
○	62	201 - Tree allometry	WARNING	Validate tree dbh / height ratio	/tract/plot[2]/tree[1]/tree_total_height	9.0
○	62	201 - Tree allometry	WARNING	Validate tree dbh / height ratio	/tract/plot[2]/tree[3]/tree_total_height	11.0
○	62	201 - Tree allometry	WARNING	Validate tree dbh / height ratio	/tract/plot[4]/tree[1]/tree_total_height	7.0
○	62	201 - Tree allometry	ERROR	Validate tree height	/tract/plot[2]/tree[1]/tree_total_height	9.0
○	62	201 - Tree allometry	ERROR	Validate tree height	/tract/plot[2]/tree[2]/tree_total_height	4.0

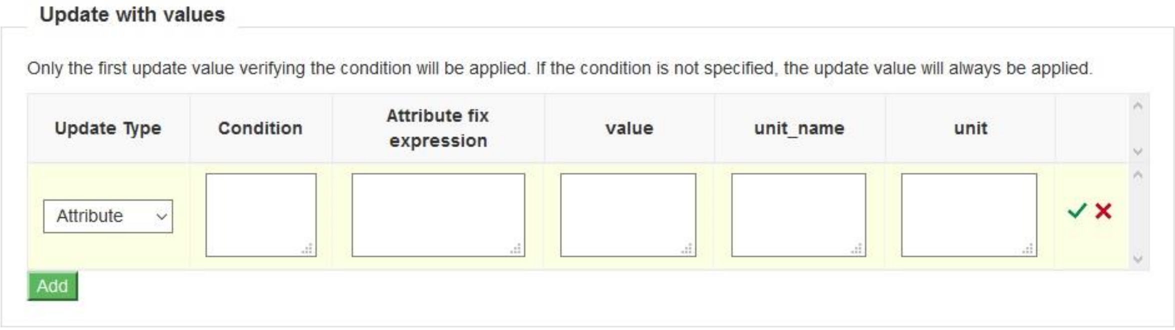
Showing 1 to 10 of 37608 rows 10 ▴ records per page

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



Update with values

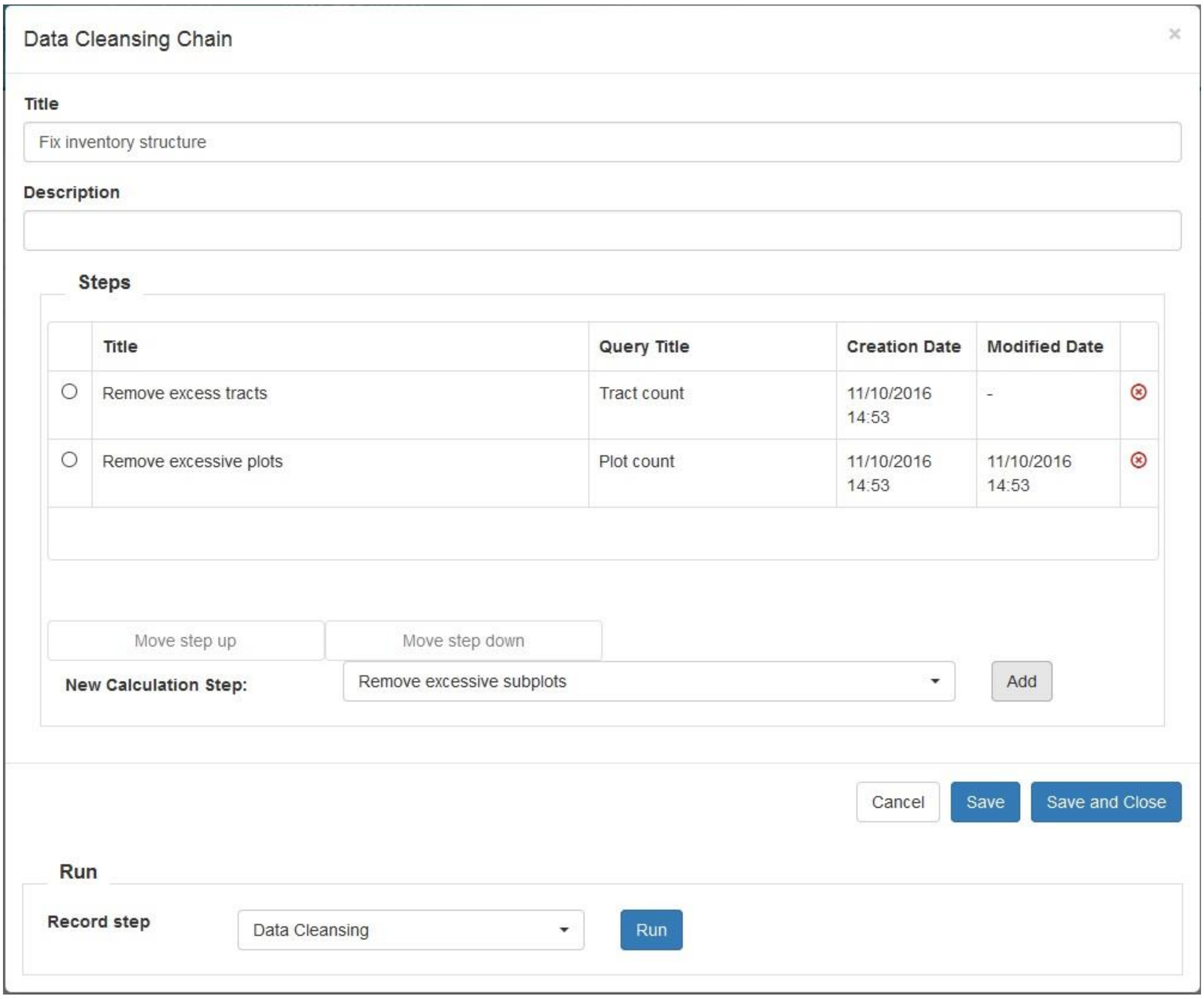
Only the first update value verifying the condition will be applied. If the condition is not specified, the update value will always be applied.

Update Type	Condition	Attribute fix expression	value	unit_name	unit	
Attribute						✓ ✕

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



Data Cleansing Chain

Title: Fix inventory structure

Description:

Steps

	Title	Query Title	Creation Date	Modified Date	
○	Remove excess tracts	Tract count	11/10/2016 14:53	-	⊗
○	Remove excessive plots	Plot count	11/10/2016 14:53	11/10/2016 14:53	⊗

New Calculation Step: Remove excessive subplots

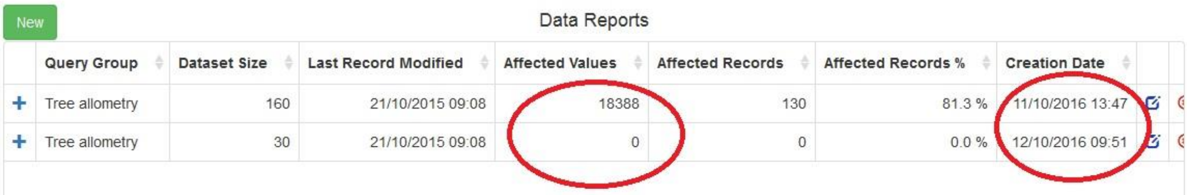
Run

Record step: Data Cleansing

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.



New

Data Reports

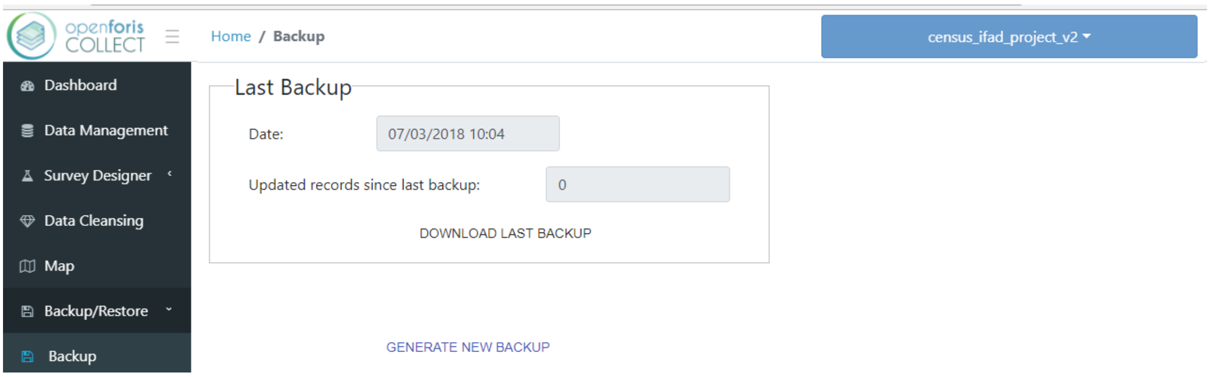
	Query Group	Dataset Size	Last Record Modified	Affected Values	Affected Records	Affected Records %	Creation Date		
+	Tree allometry	160	21/10/2015 09:08	18388	130	81.3 %	11/10/2016 13:47		
+	Tree allometry	30	21/10/2015 09:08	0	0	0.0 %	12/10/2016 09:51		

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.

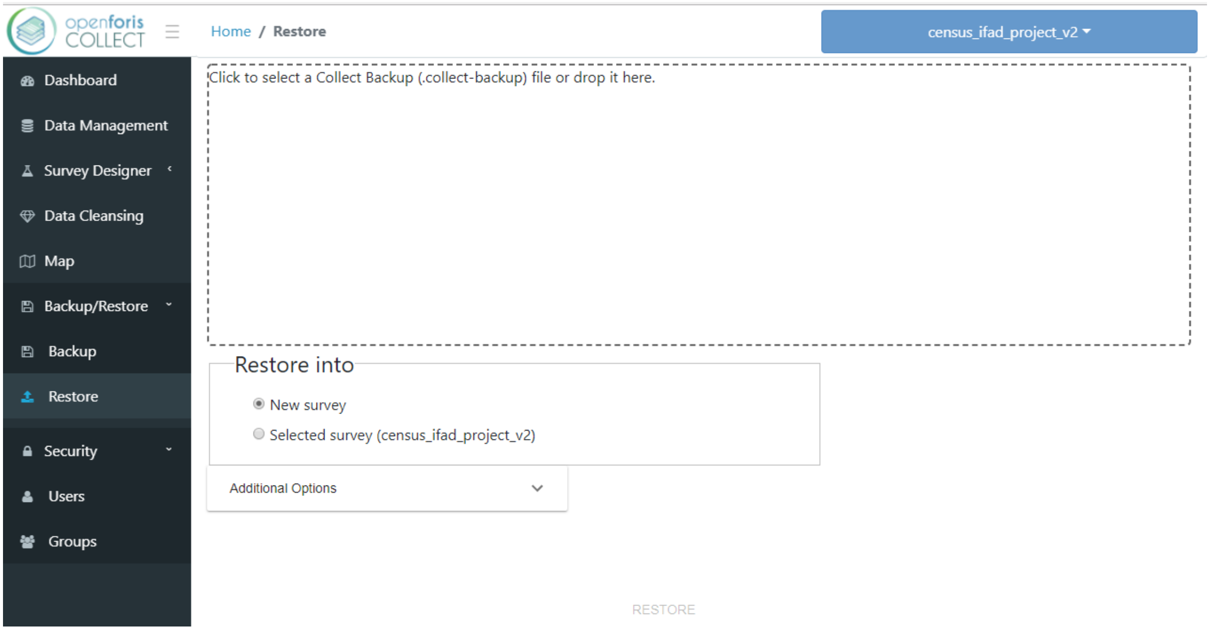


Last Backup

Date: 07/03/2018 10:04

Updated records since last backup: 0

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.



Click to select a Collect Backup (.collect-backup) file or drop it here.

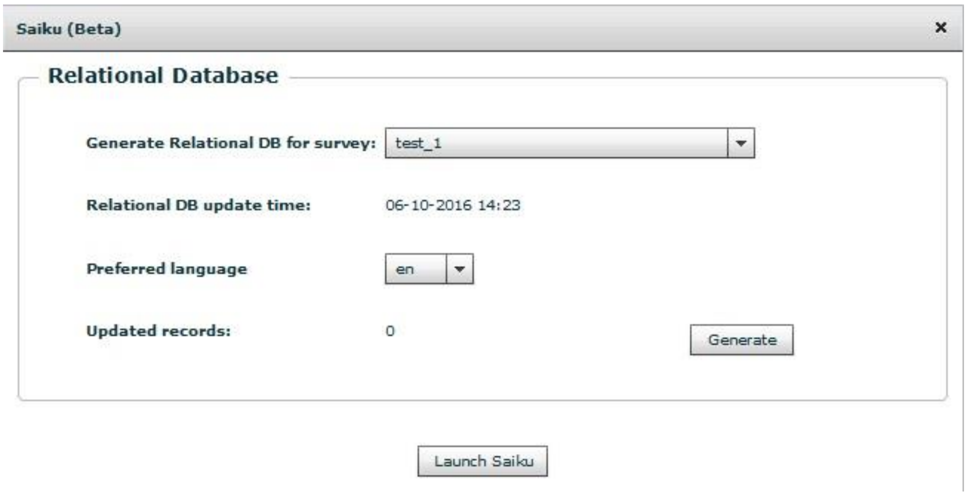
Restore into

● New survey

○ Selected survey (census_ifad_project_v2)

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.



Generate Relational DB for survey: test_1

Relational DB update time: 06-10-2016 14:23

Preferred language: en

Updated records: 0

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.

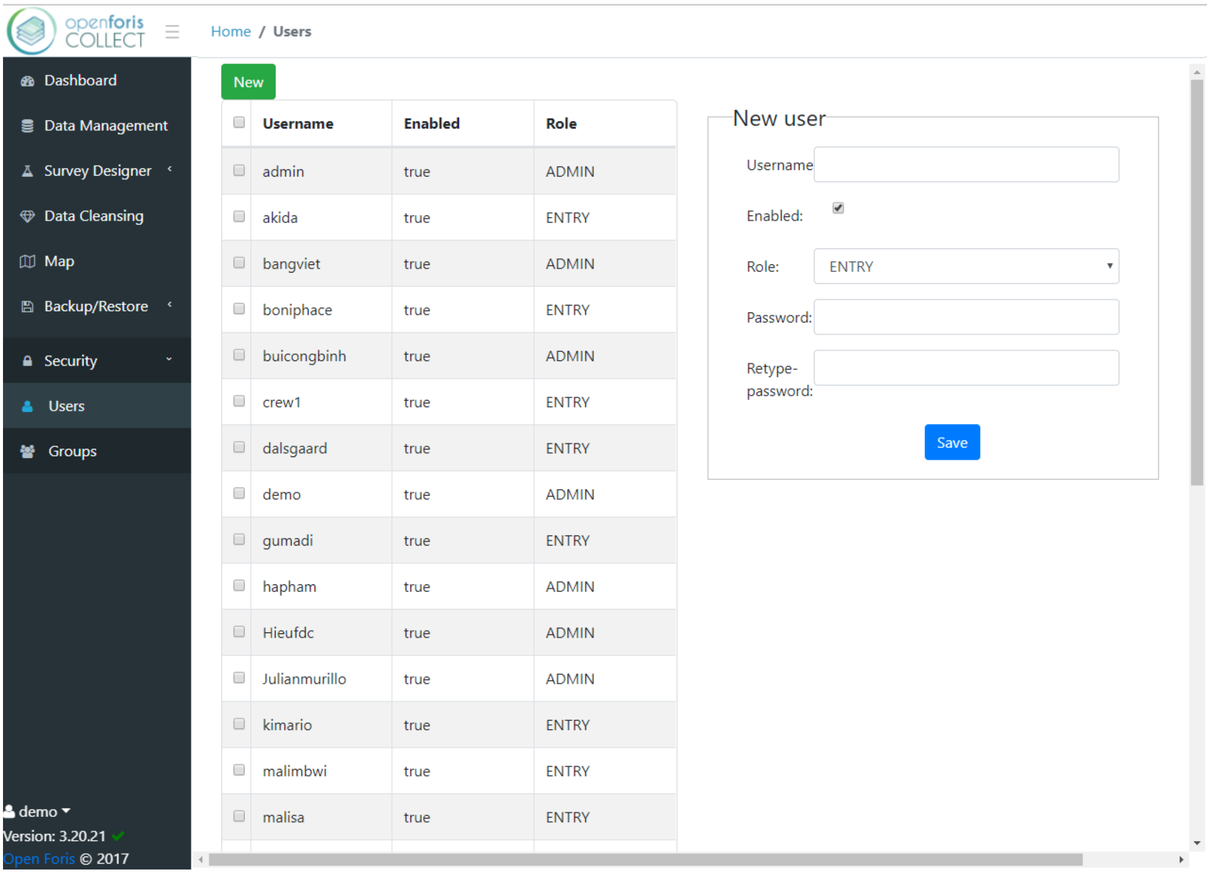
http://www.openforis.org/tools/collect-earth/tutorials/saiku.html

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.



☐	Username	Enabled	Role
☐	admin	true	ADMIN
☐	akida	true	ENTRY
☐	bangviet	true	ADMIN
☐	boniphace	true	ENTRY
☐	buicongbinh	true	ADMIN
☐	crew1	true	ENTRY
☐	dalsgaard	true	ENTRY
☐	demo	true	ADMIN
☐	gumadi	true	ENTRY
☐	hapham	true	ADMIN
☐	Hieufdc	true	ADMIN
☐	Julianmurillo	true	ADMIN
☐	kimario	true	ENTRY
☐	malimbwi	true	ENTRY
☐	malisa	true	ENTRY

New user

Username

Enabled: ☑

Role: ENTRY

Password:

Retype-password:

demo

Version: 3.20.21 ✓

Open Foris © 2017

You will be then able to fill the new user form with details as appropriate. Then click **Save**.

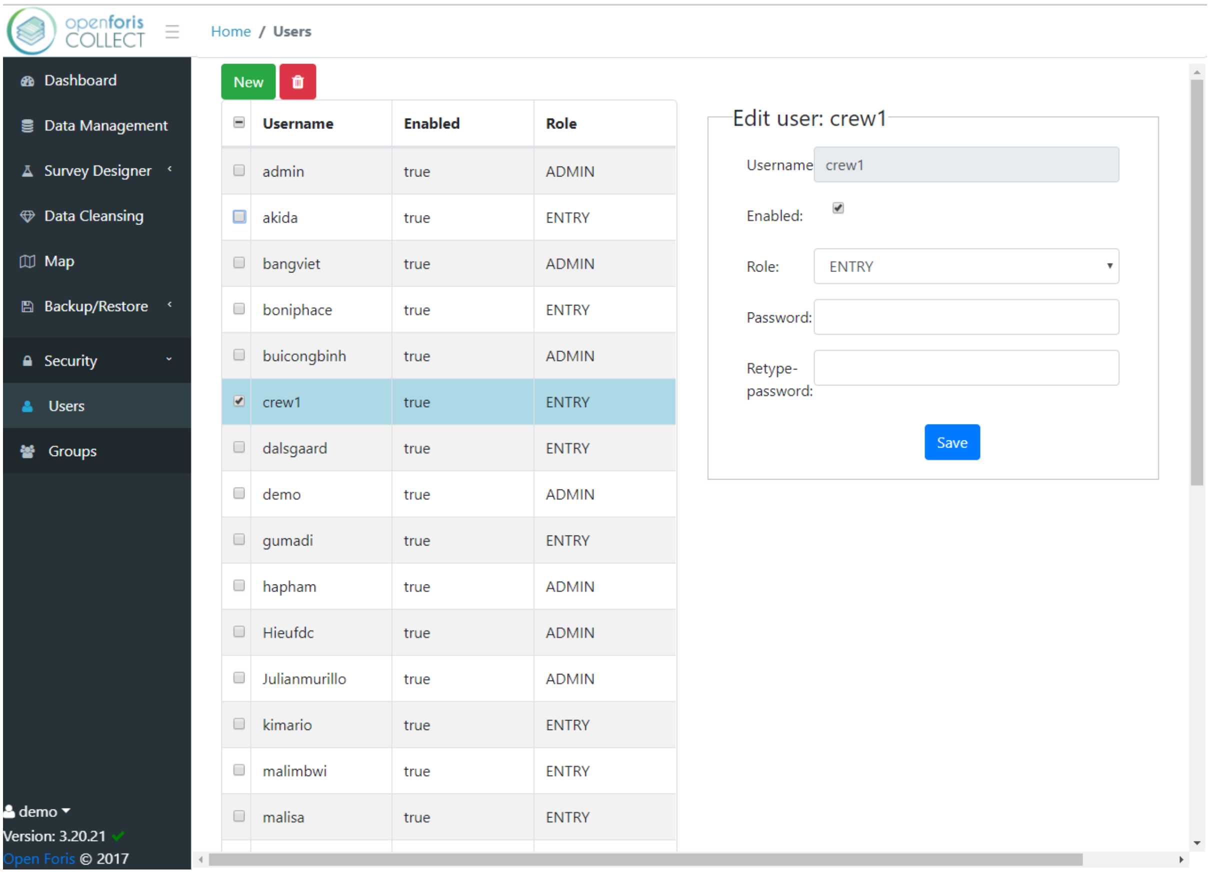
User's details include Name and password and a check box for enabling access. Most importantly, each user will have a specified **Role** which will determine the level of access to different components of Collect and the ability to work on specific phases of data workflow.

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.



☐	Username	Enabled	Role
☐	admin	true	ADMIN
☐	akida	true	ENTRY
☐	bangviet	true	ADMIN
☐	boniphace	true	ENTRY
☐	buicongbinh	true	ADMIN
☑	crew1	true	ENTRY
☐	dalsgaard	true	ENTRY
☐	demo	true	ADMIN
☐	gumadi	true	ENTRY
☐	hapham	true	ADMIN
☐	Hieufdc	true	ADMIN
☐	Julianmurillo	true	ADMIN
☐	kimario	true	ENTRY
☐	malimbwi	true	ENTRY
☐	malisa	true	ENTRY

Edit user: crew1

Username: crew1

Enabled: ☑

Role: ENTRY

Password:

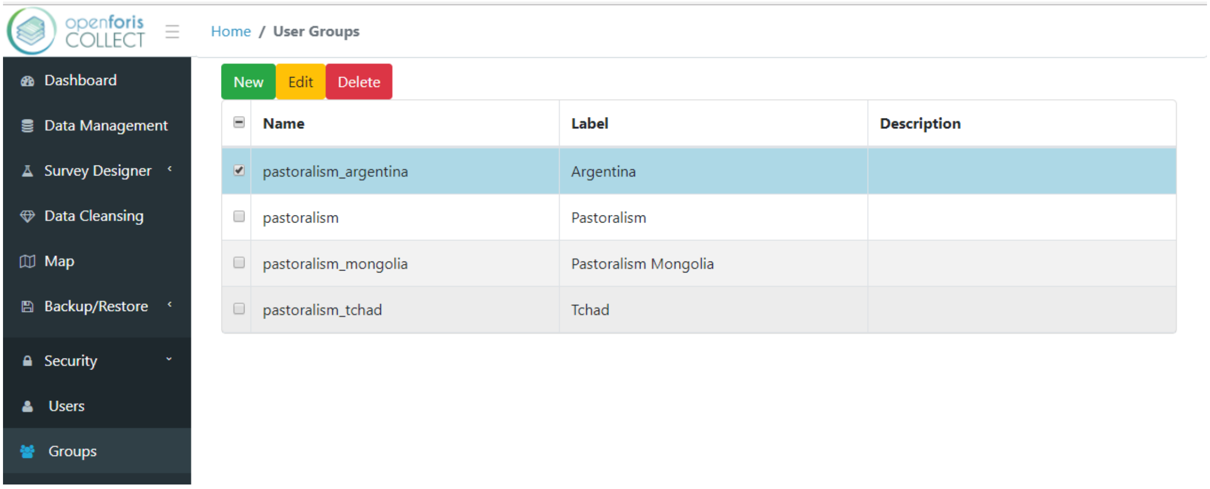
Retype-password:

15.2 User role permissions table

	View	Entry Limited	Entry	Cleansing	Analysis	Design	Admin
View records	✓	✓	✓	✓	✓	✓	✓
Create new records			✓	✓	✓	✓	✓
Edit records in Entry phase		✓ (only owned records)	✓	✓	✓	✓	✓
Edit owned and not owned records from Entry to Analysis phase				✓	✓	✓	✓
Submit records from Entry to Cleansing phase		✓ (only owned records)	✓	✓	✓	✓	✓
Reject records from Cleansing to Entry phase				✓	✓	✓	✓
Reject records from Analysis to Cleansing phase				✓	✓	✓	✓
Delete records (only in Entry phase)			✓	✓	✓	✓	✓
Saiku analysis				✓	✓	✓	✓
Data Cleansing				✓	✓	✓	✓
Survey Designer						✓	✓
Backup/Restore							✓
Users/Roles management							✓

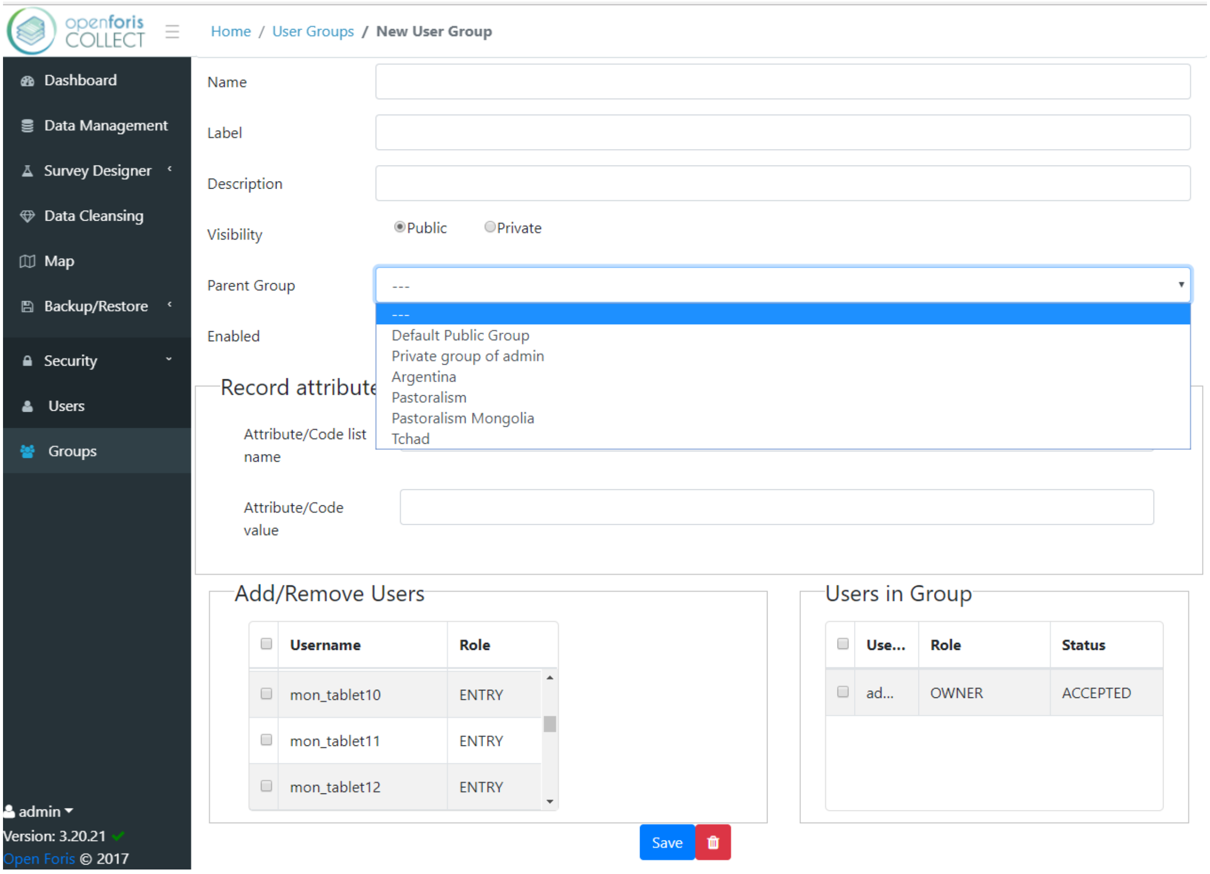
15.3 Groups

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



	Name	Label	Description
☑	pastoralism_argentina	Argentina	
□	pastoralism	Pastoralism	
□	pastoralism_mongolia	Pastoralism Mongolia	
□	pastoralism_tchad	Tchad	

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.



Name

Label

Description

Visibility ● Public ○ Private

Parent Group ---

- ---
- Default Public Group
- Private group of admin
- Argentina
- Pastoralism
- Pastoralism Mongolia
- Tchad

Enabled

Record attribute

Attribute/Code list name

Attribute/Code value

Add/Remove Users

□	Username	Role
□	mon_tablet10	ENTRY
□	mon_tablet11	ENTRY
□	mon_tablet12	ENTRY

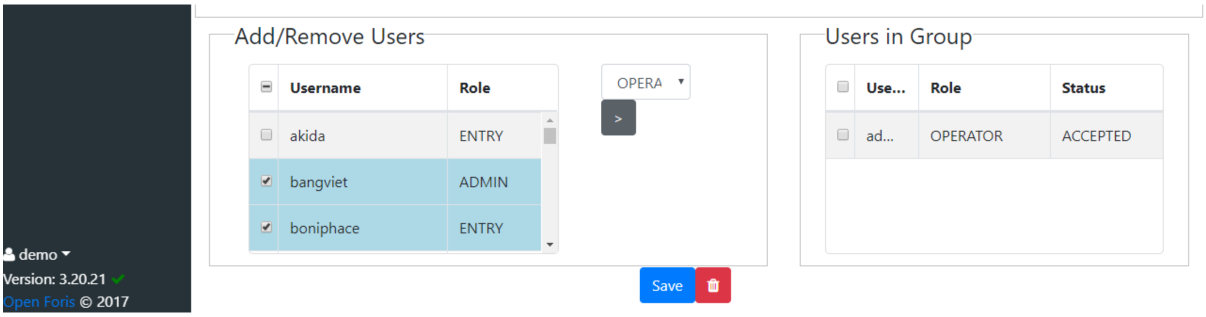
Users in Group

□	Use...	Role	Status
□	ad...	OWNER	ACCEPTED

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.



Add/Remove Users

☐	Username	Role
☐	akida	ENTRY
☑	bangviet	ADMIN
☑	boniphace	ENTRY

Users in Group

☐	Use...	Role	Status
☐	ad...	OPERATOR	ACCEPTED

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

	Viewer	Operator	Data Analyser	Supervisor	Owner	Administrator
View records (read only mode)	✓	✓	✓	✓	✓	✓
Create new records		✓	✓	✓	✓	✓
Edit records		✓	✓	✓	✓	✓
Delete records (in entry phase)		✓	✓	✓	✓	✓
Import records			✓	✓	✓	✓
Change record owner			✓	✓	✓	✓
Promote/Demote records in bulk			✓	✓	✓	✓

ANNEX 1. OPEN FORIS COLLECT EXPRESSION LANGUAGE

The IDM (Inventory Data Meta-model) expression language is used to define custom rules and other logic of a survey model. It is a subset of the W3C-standard [XML Path Language (XPath).
Expressions may contain paths and formulas which evaluate to values (e.g. total_height * 8) or other nodes in the schema (parent()/tree).
Standard XPath axes are not supported (".", "..", "ancestor", "child", "descendant", etc.). Instead, the following custom functions and special variables are provided to consistently resolve relative paths:

- *parent()*
  the parent of the entity or attribute where the expression is defined

- *$this*
  the value associated with the attribute where the expression is defined

- *$context*
  The context node of an expression, that is equivalent to the node returned by the expression *$this/parent()*

IDM expressions are always relative to the parent of the node where they are defined. For example, if you have a biophysical survey with the main entities organized in a structure like *cluster* -> *plot* -> *tree*, if you are defining an expression for the attribute *dbh* in the *tree* entity and you want to refer the attribute *height* of the same *tree* entity, then you can write simply *height > 1.3* (that is equivalent to: *$context/height > 1.3*); instead if you want to refer the attribute *accessibility* in the *plot* entity which the current tree belongs to, you will have to write something like *parent()/accessibility = 'A'.*

SUPPORTED XPATH CORE FUNCTIONS

Almost all the XPath functions are supported.

Functions on Boolean Values:

- boolean(arg)
- not(arg)
- true()
- false()

Functions on Numeric Values:

- number(arg)
- round(num)
- floor(num)
- ceiling(num)
- format-number(number, format, [decimalformat])

Functions on Strings:

- string(arg)
- concat(string, string)
- substring(string,start,len)
- string-length(string)
- normalize-space(string)
- contains(string1, string2)
- starts-with(string1, string2)
- ends-with(string1,string2)

Aggregate Functions:

- sum(arg,arg,...) or sum(nodeset)
- count((item,item,...))

Context functions:

- last()

Here you can find the complete documentation on these functions: XPath Functions Documentation

CUSTOM FUNCTIONS

idm:array(value1, value2, value3, ..., valueN)

returns a list composed by the specified values. E.g. idm:array('A', 'B', 'C', attribute1, attribute2)

idm:index()

returns the index of the current node relatively to its parent (starts from 0)

idm:position()

returns the position of the current node relatively to its parent (starts from 1)

idm:blank(node)

returns true if the attribute at the specified node has no value (symbols, units and other properties are ignored)

idm:not-blank(node)

returns true if the attribute at the specified node is filled (symbols, units and other properties are ignored)

idm:contains(multiple_attribute, value)

returns true if the specified value is among the ones selected for the specified multiple attribute

idm:samplingPointCoordinate(key1expr, key2expr, ...)

retrieves a sampling point coordinate stored in the sampling points list. The sampling point selected will be the one having the key identifiers equal to the values calculated according to the specified expressions.
E.g. *idm:samplingPointCoordinate(cluster_id, plot_no)* evaluates the expressions *cluster_id* and *plot_no* and use the values as parameters to look for a coordinate in the sampling piont list having *level1_code* equal to the evaluated value of *cluster_id* and *level2_code* equal to the evaluated value of *plot_no*.

idm:samplingPointData(dataColumnName, level1KeyAttribute, level2KeyAttribute, ...)

retrieves an extra info column associated to a sampling point in the sampling point data. The sampling point will be selected based on the values of the provided attributes.

e.g. retrieve the "province" associated to the current plot:

```
idm:samplingPointData('province', cluster_no, plot_no)
```

idm:speciesListData(speciesListName, attribute, speciesCode)

retrieves an info column associated to a species list. The species will be selected by the species code. Possible attributes that can be selected are: code, family, genus, scientific_name and all the extra info attributes defined in the species list.

There is the possibility to specify custom attributes for every species.

e.g. retrieve a custom attribute "max_dbh" associated to the species with code "AFZ/QUA" in a species list named "trees":

```
idm:speciesListData('trees', 'max_dbh', 'AFZ/QUA')
```

e.g. retrieve the "family" of a species in a species list named "trees" inserted using a Taxon Attribute named "tree_species":

```
idm:speciesListData('trees', 'family', tree_species)
```

(in this case the quotes on the "tree_species" parameter have not to be specified since it's referring an attribute and not a constant value).

idm:distance(coordinateFrom, coordinateTo)

deprecated: see Geospatial Functions / geo:distance

idm:currentDate()

returns the current date as a Date attribute value.

idm:currentTime()

returns the current time as a Time attribute value.

idm:datetime-diff(date1, time1, date2, time2)

calculates the difference in minutes between the 2 specified datetimes.

idm:datetime-diff(date1, time1, date2, time2, timeUnit)

calculates the difference between the 2 specified datetimes. Time unit can be "minute", "hour", "day".

idm:lookup(table, column, idcol1, idexpr1, idcol2, idexpr2, ...)

retrieves the *column* value from specified *table*, matching value of each *idexpr* with value in each *idcol*. This is implementation dependent and assumes the presence of table-based data which may be referenced. E.g. *idm:lookup(my_custom_table, my_column, keycol1, 1, keycol2, 3)* retrieves the value in *my_column* in table *my_custom_table* where *keycol1*=1 and *keycol2*=3.

MATHEMATICAL FUNCTIONS

Simple binary mathematical operators are supported:

Operator	Description	Example
+	Addition	6 + 4
-	Subtraction	6 - 4
*	Multiplication	6 * 4
div	Division	8 div 4
=	Equal	price = 9.80
!=	Not equal	price != 9.80
<	Less than	price < 9.80
<=	Less than or equal to	price <= 9.80
>	Greater than	price>9.80
>=	Greater than or equal to	price >= 9.80
or	or	price = 9.80 or price = 9.70
and	and	price > 9.00 and price < 9.90
mod	Modulus (division remainder)	5 mod 2

math:abs(value)

returns the absolute value of the specified *value*.

math:pow(base, exponent)

returns the value of the *base* raised to the power of the *exponent*.

math:sqrt(value)

returns the correctly rounded positive square root of *value*.

math:PI()

Returns a value that is closer than any other to pi, the ratio of the circumference of a circle to its diameter.

math:rad(angle)

Converts an angle measured in degrees to an approximately equivalent angle measured in radians. The conversion from degrees to radians is generally inexact. The conversion from radians to degrees is generally inexact; users should not expect *math:cosrad(math:rad(90.0))* to exactly equal 0.0.

math:deg(angleInRadians)

converts an angle measured in radians to an approximately equivalent angle measured in degrees. The conversion from radians to degrees is generally inexact; users should not expect *math:cos(math:deg(math:PI() div 2))* to exactly equal 0.0.

math:sin(angle)

returns the trigonometric sine of an angle.

math:sinrad(angleInRadians)

returns the trigonometric sine of an angle (in radians).

math:cos(angle)

returns the trigonometric cosine of an angle.

math:cosrad(angleInRadians)

returns the trigonometric cosine of an angle (in radians).

math:tan(angle)

returns the trigonometric tangent of an angle.

math:tanrad(angleInRadians)

returns the trigonometric tangent of an angle (in radians).

math:asin(value)

returns the arc sine of a value. Value must be in the range -1 through 1. The arc sine of a value x means "what angle y would have a sine value of x". The returned angle is in radians and it's in the range -pi/2 through pi/2. E.g. math:deg(math:asin(1)) = 90

math:acos(value)

returns the arc cosine of a value. Value must be in the range -1 through 1. The arc cosine of a value x means "what angle y would have a cosine value of x". The returned angle is in radians and it's in the range 0 through pi. E.g. math:deg(math:acos(0)) = 90 ; math:acos(1) = 0 ; math:deg(math:acos(-1)) = 180

math:atan(value)

returns the arc tangent of a value. Value must be in the range -1 through 1. The arc tangent of a value x means "what angle y would have a tangent value of x". The returned angle is in radians and it's in the range -pi/2 through pi/2. E.g. math:atan(0) = 0 ; math:deg(math:atan(1)) = 45 ; math:deg(math:atan(-1)) = -45

math:min(nodeset)

returns the the minimum not null value in a set of node values.

math:max(nodeset)

returns the the maximum not null value in a set of node values.

math:log(value)

returns the natural logarithm (base e) of *value*.

math:log10(value)

returns the base 10 logarithm of *value*.

math:random()

returns a value with a positive sign, greater than or equal to 0.0 and less than 1.0.

GEOSPATIAL FUNCTIONS

geo:polygon(vertices)

generates a KML polygon using the specified coordinates as vertices. The coordinates parameter must be a an expression returning a list of Coordinate Attribute values.

You can use this expression in a Calculated attribute to generate a polygon that can be shown in the Collect Map toolkit.

e.g. if you have a survey with a structure like *cluster* -> *plot* and you are defining this expression in an attribute inside *cluster* and you have a *location* Coordinate attribute inside *plot*, you can write:

```
geo:polygon(plot/location)
```

(mark the attribute *Geometry* in the Survey Designer in order to make it visible in the Map tool)

geo:distance(coordinateFrom, coordinateTo)

calculates the distance (in meters) between the 2 specified coordinates. Coordinate values can be constant text values (e.g. 'SRID=EPSG:21035;POINT(805750 9333820)') or expressions related to coordinate attributes

geo:latlon(coordinate)

transforms the specified coordinate into a coordinate in lat-long (EPSG:4326 Spatial Reference System). The coordinate can be a constant text value or an expression related to a coordinate attribute.

UTILITY FUNCTIONS

util:uuid()

generated a random UUID (Universal Unique Identifier) as a Text attribute value.

ENVIRONMENT FUNCTIONS

env:desktop()

returns true if Collect is running on a desktop computer, false otherwise.

env:mobile()

returns true if Collect is running on a mobile device (Android), false otherwise.

EXAMPLES

Suppose that you have a basic biophysical survey with a structure like cluster -> plot -> tree and you want to:

- make an attribute inside the "plot" entity visible only when cluster is in a specific region

  you will have to write in the "relevant when" field:

  ```
  parent()/region = "A"
  ```
  where A is the code associated to the specific region we are referring to.

- make an attribute inside the "plot" entity visible only when plot is accessible

  you will have to write in the "relevant when" field of the "erosion" attribute just this:

  ```
  accessible
  ```
  where "accessible" is the name of a boolean attribute in the "plot" entity. When referencing boolean attributes, you don't need to compare the attribute with any value in the expression.

- create a calculated attribute (read only) containing the count of all the trees in every plot

  mark the attribute as *Calculated* and define a new *Attribute default* that will use this expression:

  ```
  count(plot/tree)
  ```

ANNEX 2. 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 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 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: /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 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 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: **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 for 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’**

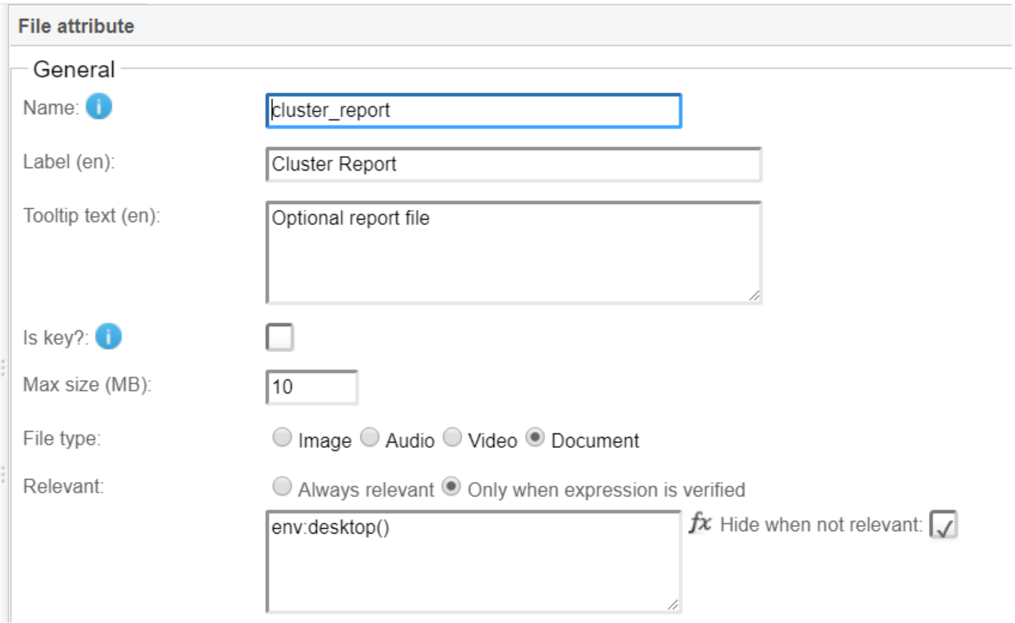
Example attribute (coordinates type): **Path: /cluster/plot/details/position**
Relevant only when expression is verified: **accessibility=’0’**
(here we don’t need a hierarchical path since position and accessibility belong to the same hierarchy, i.e: parent()/plot/details/position and parent()/plot/details/accessibility)

Example entity (multiple table layout): **Path: /cluster/plot/tree**
Relevant only when expression is verified: **details/accessibility=’0’**

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.



File attribute

General

Name: cluster_report

Label (en): Cluster Report

Tooltip text (en): Optional report file

Is key?: □

Max size (MB): 10

File type: ○ Image ○ Audio ○ Video ● Document

Relevant: ○ Always relevant ● Only when expression is verified

env:desktop()

Hide when not relevant: ☑

ANNEX 3. 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**

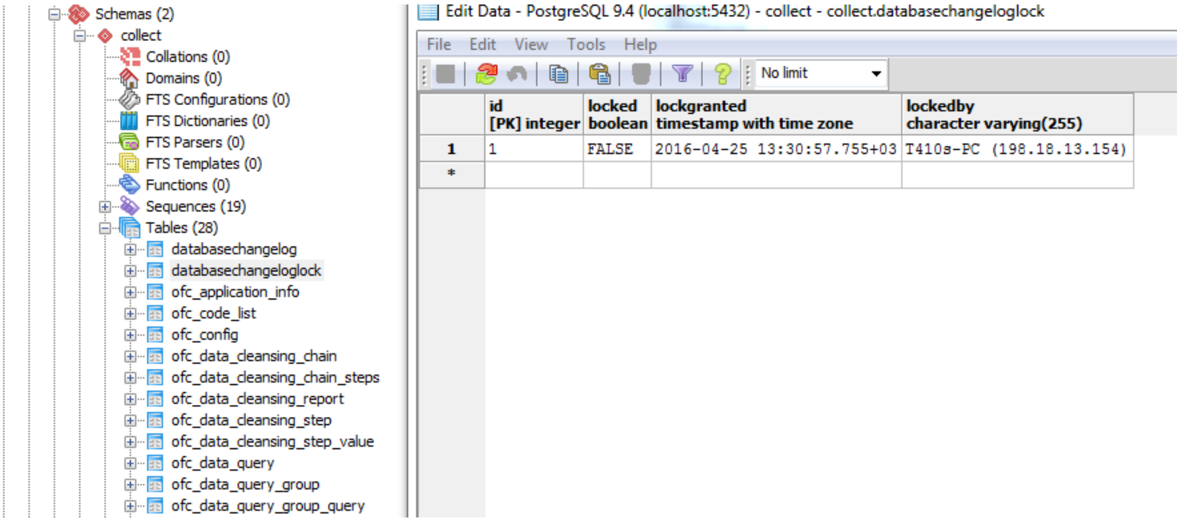
Change the settings to be as follows:

```
#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.



Schemas (2)
collect
Collations (0)
Domains (0)
FTS Configurations (0)
FTS Dictionaries (0)
FTS Parsers (0)
FTS Templates (0)
Functions (0)
Sequences (19)
Tables (28)
databasechangelog
databasechangeloglock
ofc_application_info
ofc_code_list
ofc_config
ofc_data_cleansing_chain
ofc_data_cleansing_chain_steps
ofc_data_cleansing_report
ofc_data_cleansing_step
ofc_data_cleansing_step_value
ofc_data_query
ofc_data_query_group
ofc_data_query_group_query

Edit Data - PostgreSQL 9.4 (localhost:5432) - collect - collect.databasechangeloglock

	id [PK] integer	locked boolean	lockgranted timestamp with time zone	lockedby character varying(255)
1	1	FALSE	2016-04-25 13:30:57.755+03	T410s-PC (198.18.13.154)
*				