Using entities in ODK central

# Introduction to Entities

This guide and its examples are designed to serve as an introductory resource to exploring how to use entities in ODK central. As a concept, “entities” can be thought of as “things” that need to be shared between forms. These could be tangible things such as farms, fields, plots, trees, village etc. or something more abstract such as visits or cases.

Entities will be stored within “Entity Lists” which can be thought of as databases that are shared across multiple forms. These forms can then be used to read, write and update entries on that list. In other words, forms will be able to be used to select directly which entity is being surveyed, read in the data about that entity and update new information about that entity. This is in addition to being able to create new entities.

These are extremely useful options for any kind of longitudinal or multiple form data collection project as there is a more direct tracking of these entities across the forms. In general, these can be used to replace having prebuilt csv files that have been used previously to achieve the same objectives. Therefore, aiming to minimize errors in follow up and reducing strain on data cleaning and validation later down the line.

For a more detailed introduction to the concept of entities, please see the official ODK documentation <https://docs.getodk.org/entities-intro/>

# Example 1

This first example includes a form to register individual farms for the project which would ultimately include a baseline, midline and endline survey form to assess how the wellbeing of the farmers changes over time.

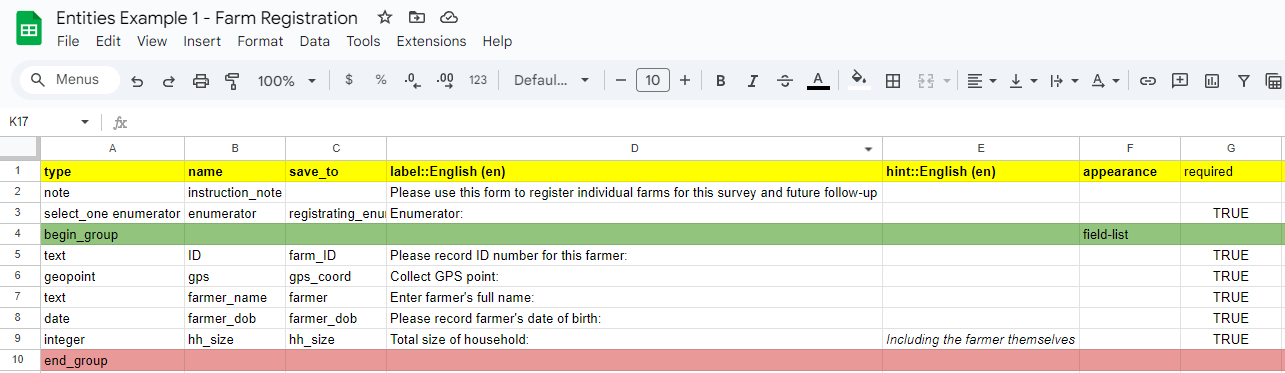
For brevity, this example will only include the baseline form, but the processes required for a midline and endline form would ultimately be virtually the same.

## Setting up the registration form

For this first example, we are creating a registration form that will be used to register our individual farms for later data collection. This form can be fairly short as it is only used to generate our list of farms. There are generally two key pieces of data to collect with such a form, GPS coordinates and an identifier such as an ID number. However, it would also be recommended to use this initial registration visit to collect some other key identifying information that could be additionally helpful in defining the household.

We can set up our XLSform largely as we normally would, using an xlsx spreadsheet in Excel or Google Sheets to write up the questions into the “survey” sheet. As you can see, for this registration form we have included a geopoint type question to collect the GPS coordinates as well as questions to give the farm an ID number, the farmer’s name, their date of birth and the household size.

There is only one change that will need to be made to the “survey” sheet, you will see there is a new column called “save\_to”. This column is used to denote which variables from our ODK from do we want to save into the resulting entities dataset that will be created by ODK Central. For a registration form, these will generally by the variables we think are most helpful in identifying the household for the future data collection forms (ID, gps, farmer\_name etc.) and variables that may be useful for monitoring (enumerator). In this column, we provide the name for the variable as we want it to appear in the entities dataset. This does not necessarily have to be the same as the “name” column.



The other key addition to be made to XLSforms to allow for the use of entities is to crate a new tab in the worksheet called “entities”. On this sheet we only need to provide two columns of information for a registration form. There is other information we can supply depending on the purpose of the form as we can also update entries as we will see later in this example.

Frist, we supply a “list\_name” which will be the name of the entities dataset we want to create in ODK Central. Next, we provide someway to identify each entry to this dataset which should be created in some way by using variables that are created within the form. In this case we are simply using the ID number we are inputting in the form (${ID}), but we could combine variables to make clearer identifiers such as using the ODK function concat() to combine the ID number and the farmer’s name.

A screenshot of a computer

Description automatically generated

## Setting up the baseline form

As with the registration form, there are additional changes that must be made to subsequent data collection forms from what would usually be expected. Firstly, the form will need to include someway of identifying which entity (farm) we are now visiting. Perhaps you have previously used prebuilt csv files to track your list of farms, households, fields etc. you may have used lookups using the search() function in the appearance column to populate options of a select\_one question. Here we are essentially trying to accomplish something similar except with fewer steps as we do not need to manually upload a csv or make any entries into the “choices sheet”. Instead, rather than using “select\_one list\_name”, we use “select\_one\_from\_file [Entities].csv” where Entities is the name of our entities dataset we specified in the registration form. In this case “select\_one\_from\_file Farms.csv”

A screenshot of a computer

Description automatically generated

For many data collection forms, this may be the only change that would need to be made. However, as previously mentioned this is a baseline survey and we may want to reference the values entered here at a later time in the midline form or the endline form. Therefore, we want to enable this form to be able to update the entities dataset. You may have noticed therefore that this form also has the new “save\_to” column. This behaves just as it does in the registration form, with us providing variables names for the questions. Note that for many of these questions, we have supplied the suffix “\_BL”, this is not mandatory in anyway but a useful option for specifying that these values will be coming from the baseline (BL) implementation of the survey. This will help separate it later down the line from any midline (ML) or endline (EL) enumeration. Not every variable has to be saved into the dataset, if you do not want a variable to become part of the Farms dataset then you simply can leave the column blank.

In order to bring in and reference variables from the registration form that were saved to the entities dataset, we need to add some calculations to pull them in. These are akin to the pulldata() calculations that have otherwise been used to bring in data from attached media files. The code is a bit more complicated however but basically you just need to use the structure:

instance(`[name of entity list]`)/root/item[name=${variable name for selecting the entity being surveyed}]/variable name in the dataset

A screenshot of a computer

Description automatically generated

Finally, as we wish to make updates to the entities dataset, we must also include the entities sheet into the survey form. Though we make one change compared to the registration form, rather than specifying the label, we use the column “entity\_id” instead. In this column, we write the name of the variable we used to select which entity is being surveyed from the survey sheet.

A screenshot of a computer

Description automatically generated

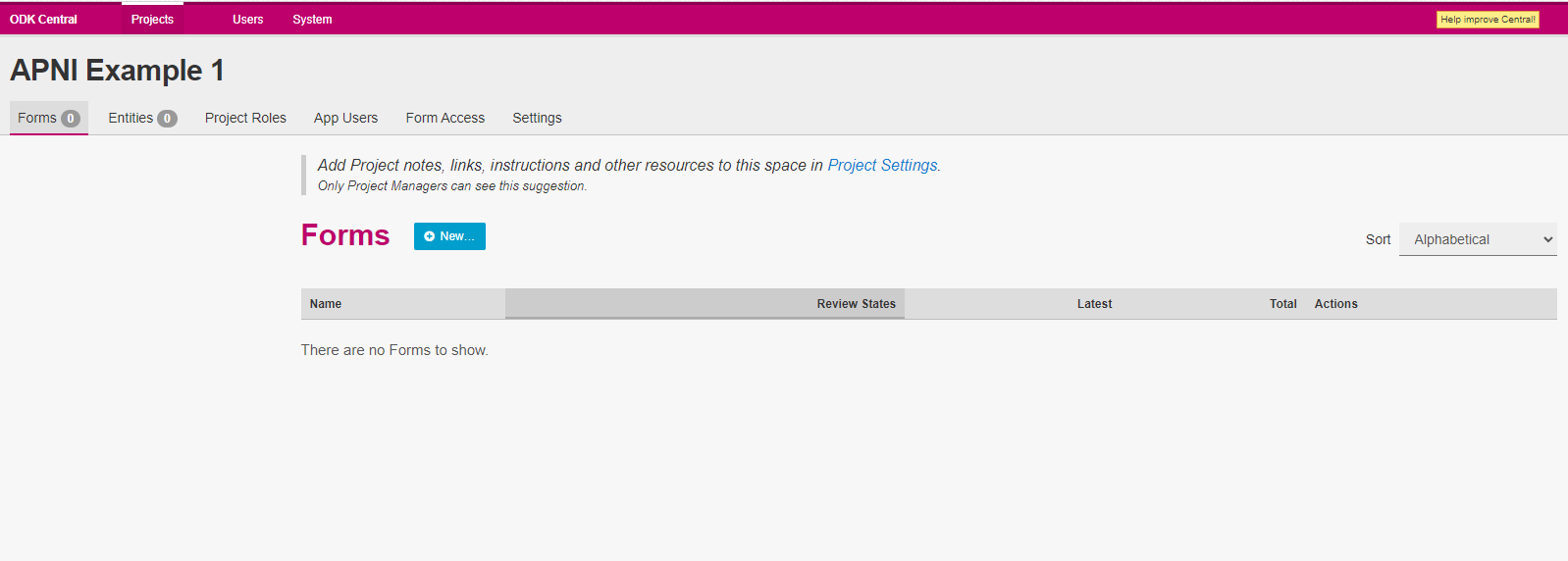
## Creating a project and uploading your forms

Upon logging into the ODK Central site you will be met with the homepage where you will be able to find all projects which are made through your server. If this is your first project, then this page will of course be blank. To create a project for your forms, click on the blue “+ New” button next to the “Projects” title. All forms that you intend to be connected as part of the overall data collection process should be under the same project.

A blue rectangle with white text

Description automatically generated

This will open a pop-up window to provide a name for the project, once written click on “create”. This will then take you straight to the homepage for this new project as seen below.



To upload the first form, starting with the registration form, again click on the “+ New” button next to the “Forms” title. Then drop in the XLSform or chose the file from the file explorer and press “upload”. If there are any errors in your form, you will be informed and prevented from uploading the file. You can use the following site to validate your forms before uploading <https://getodk.org/xlsform/>

A screenshot of a computer

Description automatically generated

Once successfully uploaded, the homepage for that form will appear. Notice that under “Your Current Draft” there is a section listing that there is 1 entity list updated by this form, named “Farms” which has 6 properties (the 6 variables we save in the registration form).

A screenshot of a computer

Description automatically generated

Testing works slightly differently currently when using entities. ODK Central currently does not allow for full testing of entities as only published forms can create entities. Therefore, any submissions to the registration form while it is still in “testing” will not be saved into the entities dataset and thus cannot be used in the testing for the later forms. It is recommended that you create a project purely for testing the forms separately from your real data collection project in order to test the full flow of creating entities and then collecting data for those entities.

A screenshot of a computer

Description automatically generated

To publish the form, return to the “Status” tab and click “Publish Draft…”. A pop-up window will appear which will show you the details of the creation of the entity list and the variables which will be uploaded.

A screenshot of a computer

Description automatically generated

You can repeat the same process for the baseline form by adding it to the same project as the registration form. You do not need to create a project for each form. The main difference being that there will be an additional entry on the form page for a “Dataset” which is referring to the entities list created by the registration form. Additionally, as we are updating the entities list using the baseline form the number of properties attached to the entity list has increased.

A screenshot of a computer

Description automatically generated

## Collecting data through Enketo or ODK Collect

Collecting data for these forms is largely the same as it would be for any other project. Either Enketo or ODK Collect will both be able to update and receive the entities list for both forms.

### Enketo

To us Enketo on your computer to complete data entry, go to the submissions tab and again click on “+ New”.

A screenshot of a computer

Description automatically generated

The form will then open in your browser to be completed as normal.

A screenshot of a computer

Description automatically generated

### ODK Collect

There are a few additional steps you first must complete before being able to collect data through the app on your mobile device. Return to the project home page and navigate to the “App Users” tab. Click on “+ Create App User” and provide a display name for these users such as “enumerators”. This will generate a QR code that can be scanned in the app to set up the project.

A screenshot of a computer

Description automatically generated

Before scanning the code and collecting however, navigate to the “Form Access” tab and make sure that the appropriate app users have been given access to the correct forms by checking the tick boxes and pressing “Save”

A screenshot of a computer

Description automatically generated

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generatedThen go back to “App Users” and press the “See code” link. This will open the QR code you can then scan on ODK Collect to configure your device for data collection. Then you can complete submissions for the first form as normal.

A screenshot of a phone

Description automatically generated

## Entities in ODK Central

By default, every submission will automatically create an entry into the entity list. To see all the information and entries to the list, go to the “Entities” tab from the project homepage. Here there will be a list of all entity datasets which are created and updated from all forms in the project. In this case we see just one listed, Farms. From this page you can also export this dataset in a csv file.

A screenshot of a computer

Description automatically generated

Clicking onto the name of the dataset (Farms) willl then open an overview page which details the properties of the dataset and its connections to the forms in the projects.

A screenshot of a computer

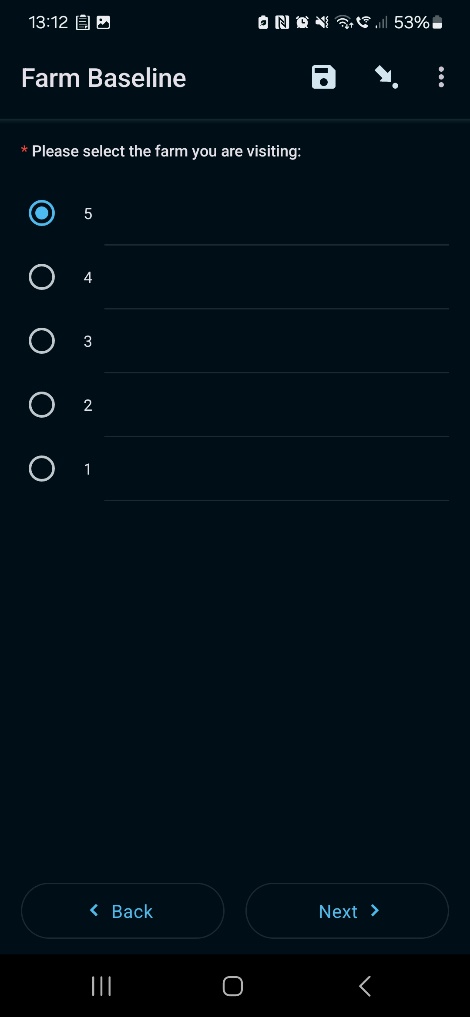
Description automatically generated

Opening the “Data” tab opens a table view of the dataset including metadata as to who created the entry, when it was created and when it was last updated. Below is the view created after filling out 5 entries in the farm registration form.

A screenshot of a computer

Description automatically generated

## Completing the Baseline form and viewing the updated entities

Now that we have 5 registered farms in the entities dataset, we can make our visit for the baseline survey. Currently, a stable internet connection is required to retrieve the entities dataset following each registration. So you may not be able to complete registration and data collection in the same visit if you cannot guarantee you will be able to upload completed submissions and download updates while in the field. Working offline with entities is on the roadmap for future updates to ODK.

Collecting data is virtually unchaged from how you may have used ODK in the past. Simply open the form on Enketo or ODK Collect and select which farm you wish to survey when prompted. In this case, we are presented with the ID numbers we created when registering farms. Once the farm is selected, the data collection can then proceed as normal.

In this form, after selecting the farm you will also be presented with a screen stating the name of the farmer and their date of birth, these being variables form the entity dataset we were able to bring across from the registration form.

Once we haved entered a few submissions for the baseline form, we can go back the entities dataset on the ODK Central site and see that the dataset has now been updated with the baseline results.

A screenshot of a computer

Description automatically generated

# Example 2

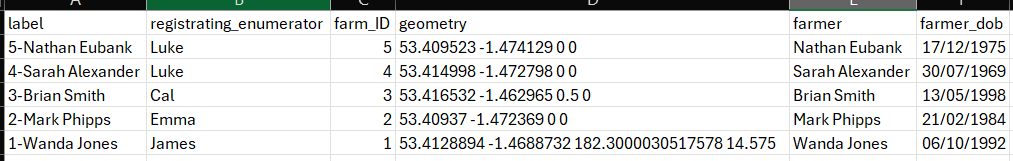
In this second example, we will be testing 4 treatment options on different fields for each farm in our survey. There will be 4 fields per farm, with each using a different treatment option.

Rather than registering our farms we will instead be starting with a prebuilt list of farms and then using these farms to help populate a registration form for each field. We could start with a farm registration form, and then register our fields in the same visit using a second form but as mentioned previously this would require a stable internet connection which may not always be feasible.

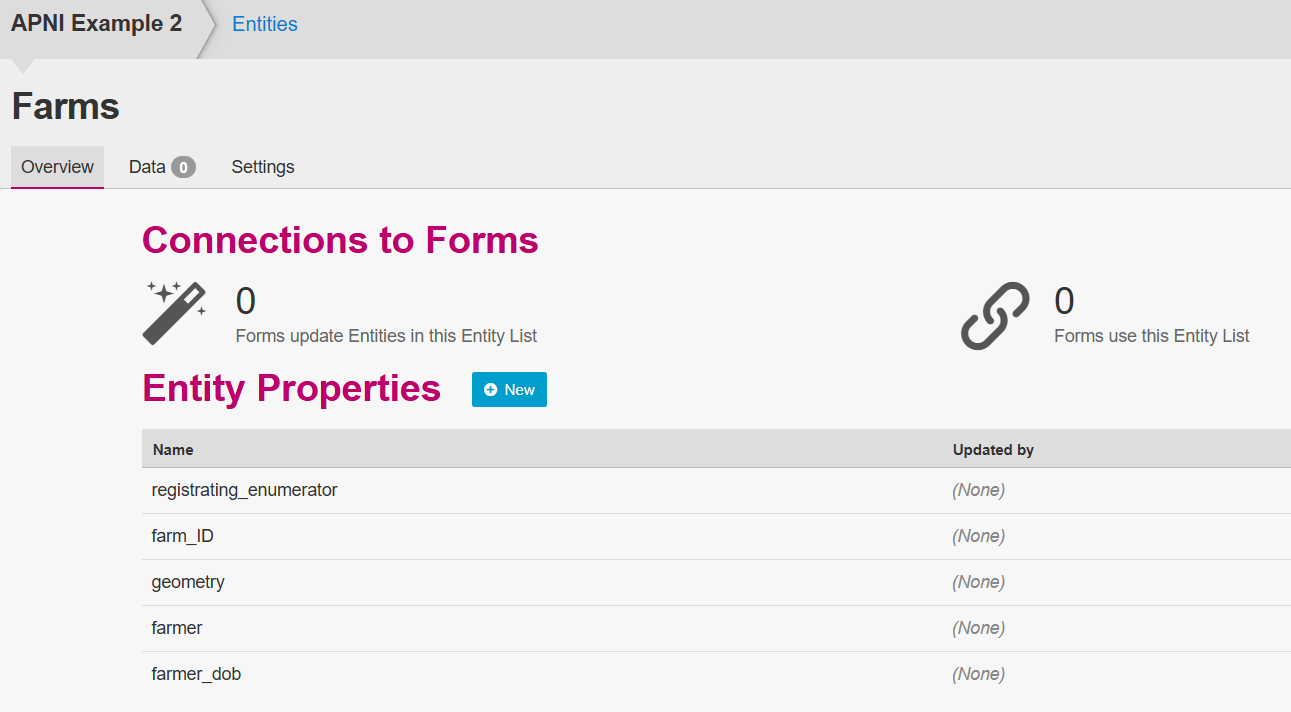
These fields will then subsequently populate a form for the first visit after applying the treatment. In theory this would then be supplemented with forms with for second, third, fourth etc. visits to monitor changes over time.

## Uploading prebuilt entity lists

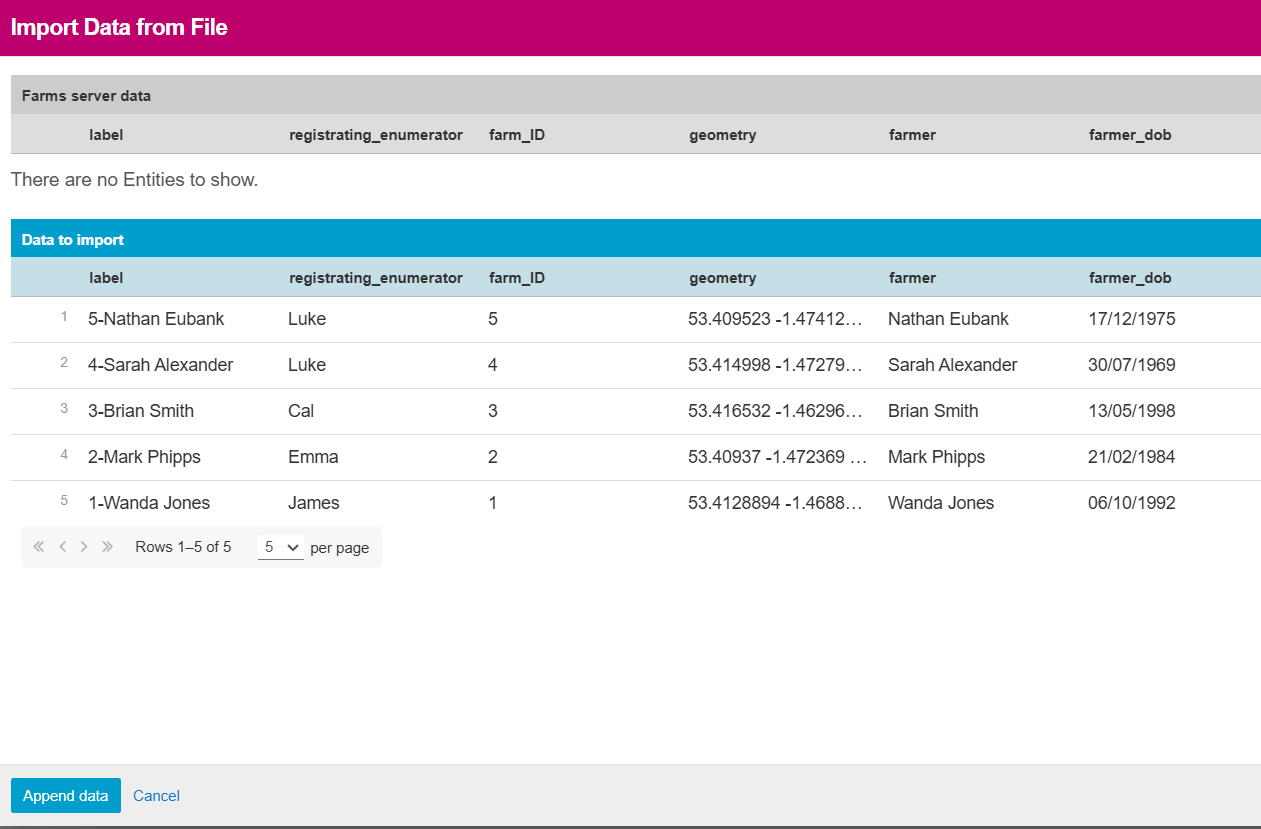
For this example, we are starting with a pre-existing list of farms from which we will collect data across 4 fields each. Below is a csv with 5 farms listing a label for the farm, the enumerator, an ID number, the gps coordinates, the farmers name and their date of birth. This csv was created using the data collected from example 1, when trying out this example for yourself you can either use the data we have attached to this guide or create a csv file of your own. If you chose to do the latter, make sure that the structure of the data remains the same with the same column names.



After creating the project, go straight to the entities tab and click on “+ New” to create an entity list. Before we can upload the csv file, we first need to define the properties of the entity list, i.e. the variables in our csv. This must be done for all columns in the data file except for the label column. This is done by clicking “+ New” next to “Entity Properties” and typing in the name of each column from the dataset. As this is done manually it is recommended that an initial sample frame of entities only contains the variables that are absolutely needed for the forms you will be using.



Once you have every variable listed, move to the “Data” tab and press “Upload”. In the pop-up window, then drag in the csv file or upload it through the file explorer to import the data. The window will then show a table of the entries that are to be imported. Press “Append data” to bring the data into the server.



## Using GPS and map appearances

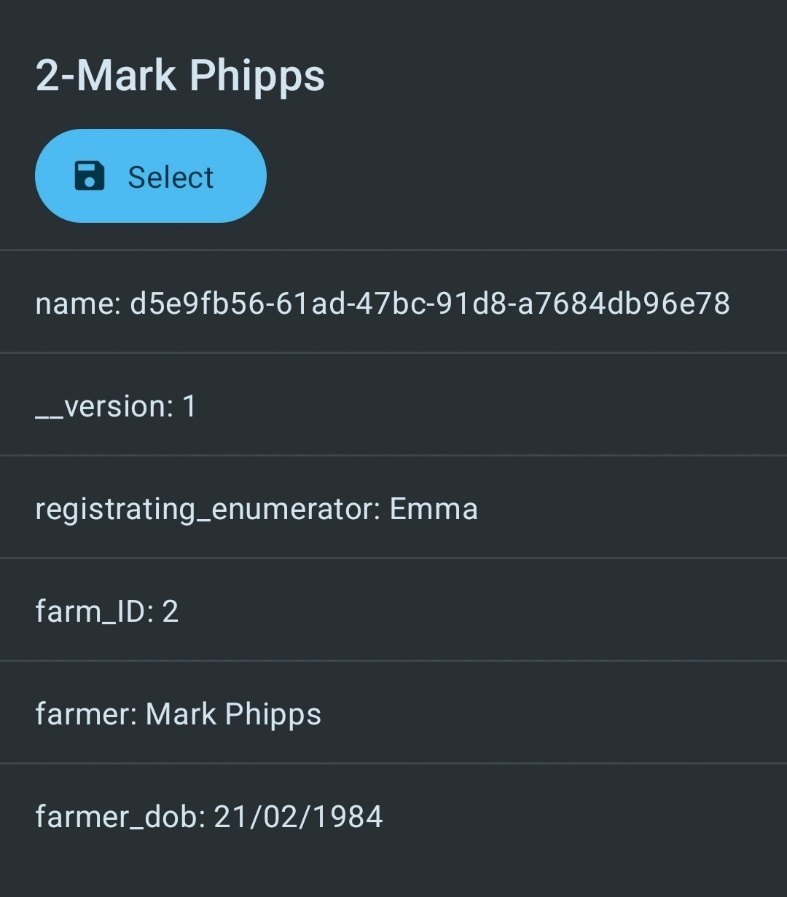
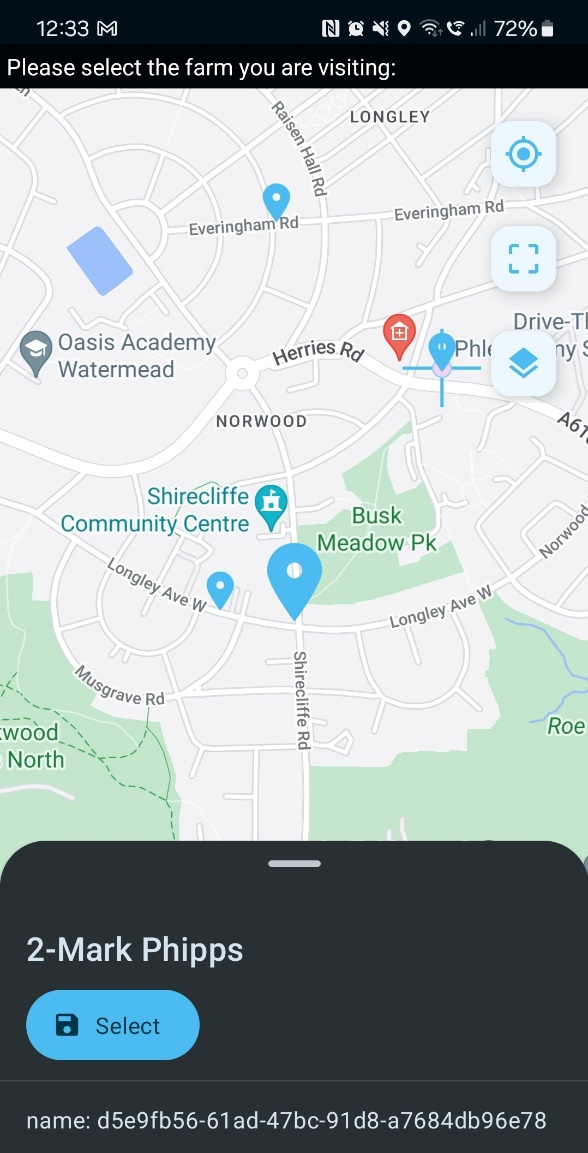
A very useful feature to use when dealing with entities is to enable the map appearance for selecting the entity being surveyed. This allows the data collector to view the farms on a map to aid in selection, rather than relying on knowing ID numbers, addresses or farmers names.

In order to do this however, the gps coordinates attached to the entity must be saved into a column called “geometry”. ODK will not recognize coordinates saved to a column by any other name. Within the XLSform, the “map” appearance should be specified for the selection question as seen in the field registration form below.

A screenshot of a computer

Description automatically generated

When collecting data in ODK Collect, you will be able to “Select place” when reaching the farm selection question. This will open a google maps view which you can zoom into your location by pressing the compass button in the top right corner. When selecting a point on the map a tab will then open at the bottom of the screen with the information available from the dataset which the enumerator can use to be sure they are selecting the correct entity. Press “Select” to then move through the rest of the form. Note that map appearances for select questions will not work in Enketo currently, they are only available for mobile devices.



## Choosing and creating entities in the same form

Within this example, we first have our farm entities which have been pre-defined from which we are first registering four fields using ODK. The survey sheet for this form can be seen at the top of the page. In terms of selecting the farm, the set up is the same as seen in the first example with using a “select\_one\_from\_file Farms.csv” and using the instance calculations to bring in the farm ID and farmers name.

This is followed by a “select\_one” to define the field number as well as gps coordinates for the field, the crop grown on the field, the treatment used and the date of application.

To create the “Fields” entity list, its details must be filled out in the “entities” sheet. Here we have defined the label a little more directly by using a combination of the farm ID, the farmers name and then the field number.

A screenshot of a computer

Description automatically generated

Also note that we filled out the “save\_to” column for our calculations to bring in data from the Farms data, this allows us to additionally save that data into the Fields dataset. This will make connecting farms and their fields much simpler both in forms and during data cleaning and analysis.

## Manually uploading new entities

Just as we used a full pre-built list of farms to populate the Farms entities, we can also manually upload additional entries to entity lists. Below is the first set of entries to our “Fields” entity list, the four fields for farm 1. Let’s imagine that the fields for farm 2 for whatever reason are in a separate csv, perhaps there was an issue with the mobile device at the time of field registration so details were collected by hand. Clicking on “Upload” will open a pop up window with the entities shown in a table at the top of the page.

A screenshot of a computer

Description automatically generated

In order to be able to upload new entries to the list, the data file must be in the same format with all the same columns and headers. This can be made simpler as this pop up window allows for the download of a template csv with the column headers preloaded.

A screenshot of a computer

Description automatically generated

Using this template we can then fill in all the details for the fields on farm 2.

A screenshot of a computer

Description automatically generated

After dropping this file into the pop up window, a 2nd table will appear showing the new entries that will be uploaded to the dataset.

A screenshot of a computer

Description automatically generated

After clicking on “Append data”, you will see that the new entries to the Fields dataset have been added to the data table.

A screenshot of a computer

Description automatically generated

## Collecting data on the first visit

The second form in this project collects information about the growth and damage to the plants during a first visit to each field. We can actually connect to both the Farms and Fields datasets to help ease the selection process. So rather than having 1 selection that will list all fields, we can first select the farm, then select the field. This can be done as we can treat the “select\_one\_from\_file” with choice filters just like we can from a standard “select\_one”. As such we first select the farm, then bring in the farm ID and farmers name. In the choice filter for the field selection we can then use farm\_ID = ${farm\_ID} to filter the options down to just the fields for the selected farm.

We can then bring in data from the fields dataset in the same way as we did for farms, feeding this information back in a note to the enumerator. As this form is not updating the entity lists in any way, there is no need for the “save\_to” column or the “entities” sheet.

A screenshot of a computer

Description automatically generated

## Additional resources

This guide is not an exhaustive guide on all topics related to entities, please review the additional documentation from the ODK team to find more information including resolving “parallel updates”, manually editing entities in ODK central, FAQs and some of the limitations of using entities.

<https://docs.getodk.org/entities-intro/>

<https://docs.getodk.org/tutorial-community-reporting/>

<https://docs.getodk.org/central-entities/>