ODK ACF-E TOOLKIT

ACF-E FORMS

AUTHOR SURVEYS FOR ODK COLLECT USING XLS FORM SYNTAX

The following guidelines are intended for ACF-E staff that is willing to learn how to ***author surveys for ODK Collect using xlsform syntax***

INTRODUCTION

ODK Collect is a powerful tool to rapidly gather complex data types in the field using Android devices**.** Using ODK Collect with ODK Aggregate and ODK briefcase, requires the creation of an ***Excel file that contains the questions, formatting instructions and data validation conditions that will allow enumerators collect data on Smartphone or Tablet devices running Android***. The following modules will cover all the syntax elements you will need to author basic and advanced surveys.

Here are the steps involved in writing a survey with the XLSform format and getting it quickly deployed on ODK Suite.

1. Write a survey using the XLSform syntax described below and save it as an .xls file.
2. Convert your excel .xls file to .xml format
3. Upload your .xml file into ODK Aggregate or use it directly in ODK Collect.
4. Configure ODK Collect to your ODK Aggregate account, download forms, and start collecting data!

Author: Jorge Durand Zurdo

BASIC SURVEY AUTHORING

Survey authoring in Excel is a simple process once you understand the basic XLSform syntax.

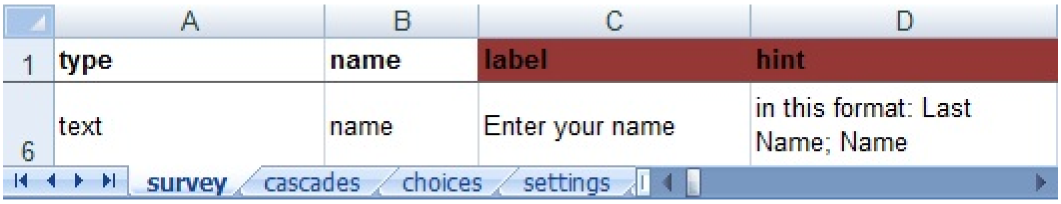
To make your learning process smoother a generic survey example will be used across the manual, this sample survey is an example form that has been designed for training purposes.

In case you want to modify the example form you can find the Excel survey file from your technical advisor in HQ and in the server address every ACF-E mission has available for ODK usage (i.e. http://odk.acf-e.org/ forms), the sample form we are going to use is named "***acf_sample_xlsform".***

> *Once we modify the original file we will save the survey with the file name acf_sample_xlsform_0_1.xls, acf_sample_xlsform_0_2.xls, acf_sample_xlsform_0_3.xls……) this notation will help us on managing the forms and keep a tracking on changes introduced.*
>
> *Note that we name the forms with underscores and no spaces*

FORM STRUCTURE

Within the excel file there is 4 worksheets named "survey", "cascades", "choices" and "settings"



	A	B	C	D
1	**type**	**name**	**label**	**hint**
6	text	name	Enter your name	in this format: Last Name; Name

IMPORTANT RULES TO REMEMBER:

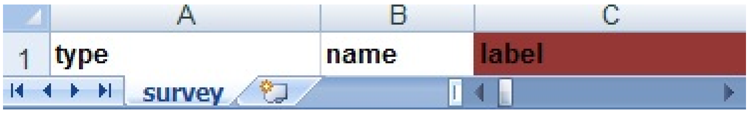
- Make sure your file is saved in the .xls format and contains no spaces or special characters ('-' and '_' are allowed).
- Make sure that your column headers are in lowercase (i.e. "label" or "name", not "Label" or "Name")
- Make sure that your sheet names are appropriately named (i.e. "survey" not "Sheet 1", "Survey" or "surveys")
- Make sure that the question names are unique and do not contain spaces or special characters ('-' and '_' are allowed).

"SURVEY" SHEET

"survey" is the worksheet where all questions should be located (each row in this worksheet is a question in your survey)

Description	**Column Headers**
Establish the type of question and the form structure	**type**
Assigns a unique variable name that will serve as a reference to the survey question	**name**
Establish what text will be displayed to the user when opening the form with ODK Collect	**label**
Establish what hint or secondary text will be displayed to the user when opening the form with ODK Collect	**hint**
Provides limited range of possible answers that can be considered as valid	**constraint**
Specify the message displayed on the screen when the constrain is not true	**constraint_message**
Designate certain questions as "required" questions	**required (yes, no, TRUE, FALSE)**
Designate defaults answers in case the user does not enter anything	**default**
Only present a question to the enumerator if a condition, based on the response to a previous question, is true.	**relevant**
Designate certain questions as "read-only" questions	**read_only (yes, no, TRUE, FALSE)**
Perform dynamic calculations using the values of preceding questions	**calculation**
This three columns can be translated to any language we want , just setting column::Language (i.e. label::English, label::Swahili)	**hint/label/constrain_message::language**
Establish what image will be displayed to the user when arrives to the question with the image(it requires that in the /odk /media folder in the device exist a copy of the image to be displayed	**media::image**
Establish what audio will be played to the user when arrives to the question with the sound(it requires that in the /odk /media folder in the device exist a copy of the audio track to be played	**media::audio**
Establish what video will be displayed to the user when arrives to the question with the video(it requires that in the /odk /media folder in the device exist a copy of the video to be displayed	**media::video**
This column can be configured to show a different image depending on the language we are using	**media::image::language**
This column can be configured to play a different audio file depending on the language we are using	**media::audio::language**
This column can be configured to show a different image depending on the language we are using	**media::video::language**
Allows for certain changes in the formatting of Smartphone screens	**appearance**

As you can see in the "survey" sheet there are several columns with different properties each



A	B	C
type	name	label

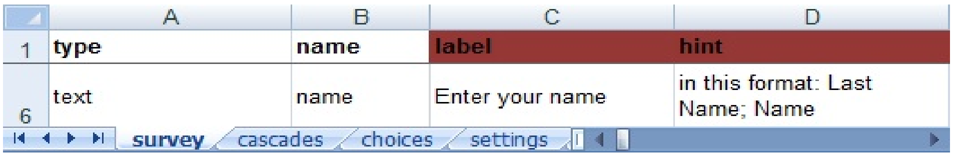
- The **'type'** column describes the question type (text, number, photo, ...), design the form structure by establishing the different groups and repeat groups and also could be used for gather device properties data when the surveys is collected
- The **'name'** column assigns a unique variable name that will serve as a reference to the survey question (the name must be unique, must begin with a letter, and can only contain letters, numbers, dashes (-) and underscores (_)).
- The **'label'** column contains the text that will actually be presented on the Android device.

1. The survey pictured bellow has one basic question which asks "Enter your name?" ('label') and presents a text input box (as specified in 'type' column) to the surveyor. Once data has been collected and sent to the server the data results Excel file that can be downloaded from ODK Aggregate will have a column called 'name' (the identifier for this question as specified in 'name' column)
2. This file will be saved in the 'Excel 97-2004 Workbook (.xls)' format.

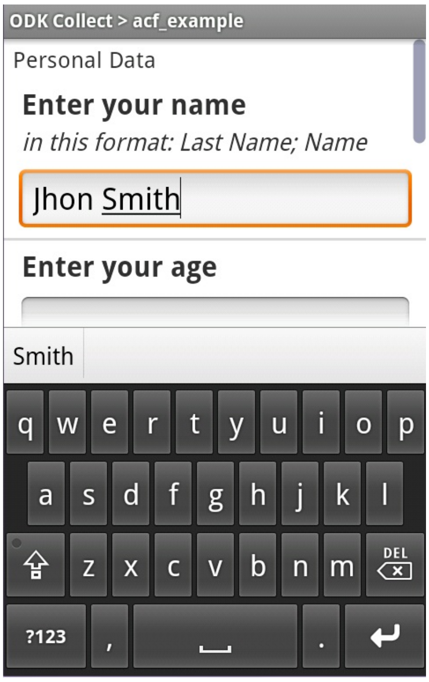
Now let's start populating the survey questionnaire. The first section will likely include basic information such as the respondent's name, age of the respondent, household number, etc. We already know how to collect the name. For age, household number, and other fields, ODK lets us specify the "type" of the information to be entered (numbers, dates, etc). This will help us reduce data entry errors. To do this we can use a different question 'type', ODK Collect has defined a number question types to support all data types that surveys usually collect. Here we present some of the most basic types ODK Collect supports (you'll see the full list later). Let's add the survey respondent's age as numeric integer fields and the respondent's name as a simple alphanumeric field. This will require the use of the 'type' 'text' and the 'type' 'integer'.

EXAMPLES

A **text** field would look like this on the Android device screen ODK Collect application:



	A	B	C	D
1	type	name	label	hint
6	text	name	Enter your name	in this format: Last Name; Name



ODK Collect > acf_example

Personal Data

Enter your name

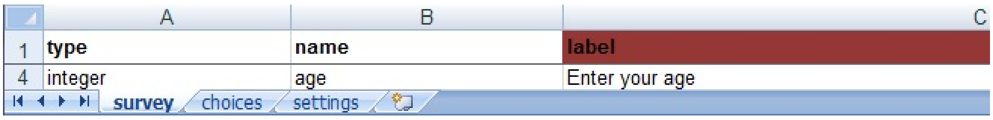
in this format: Last Name; Name

Jhon Smith

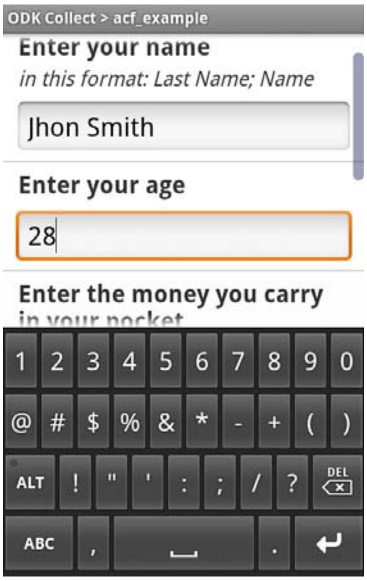
Enter your age

Smith

An **integer** field would look like this on the Android



	A	B	C
1	type	name	label
4	integer	age	Enter your age



ODK Collect > acf_example

Enter your name

in this format: Last Name; Name

Jhon Smith

Enter your age

28

Enter the money you carry in your pocket

Notice how the numeric keypad appears automatically and the alphabetic keyboard is deactivated.

Question Types

Data collecting with an Android device allows us to gather different kind of information within the same survey (photos, audio, video, GPS coordinates and information encoded in barcodes or QR codes).

Description	***Question Types***
Insert an Input text box	**text**
Insert an Input integer (number) box	**integer**
Insert an Input integer (decimal number) box	**decimal**
Insert a dialog box with options of select one of the choices given	**select_one [options] [or_other]**
Insert a dialog box with options of multiple select of the choices given	**select_multiple [options] [or_other]**
Insert a check box	**acknowledge**
Insert a date (dd/mm/yy) widget	**date**
Insert a date (dd/mm/yy) & time (hh:mm) widget	**datetime**
Insert a time (hh:mm) widget	**time**
Insert a GPS launcher	**geopoint**
Insert a Barcode launcher	**barcode**
Insert a Camera launcher	**image**
Insert a Audio recorder launcher	**audio**
Insert a Video recorder launcher	**video**
Insert a Video recorder launcher	**note**
Creates an instance variable with no body element that gets set by an xpath expression (specified in the calculation column)	**calculate**

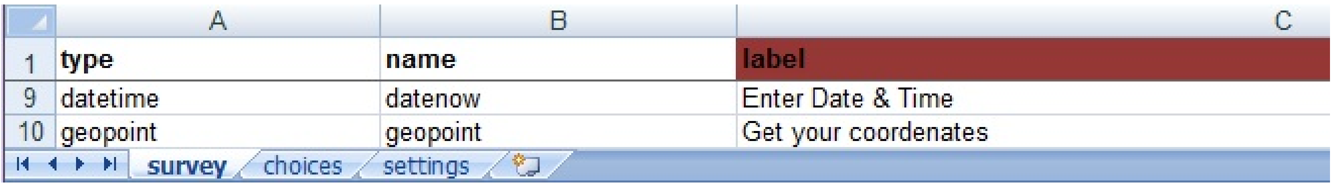
GEOLOCATION AND IMAGE

Now we’ll see how to use two of this Question types (‘geopoint’ and‘image’).

geopoint	Collect GPS coordinates.
image	Take a photograph.

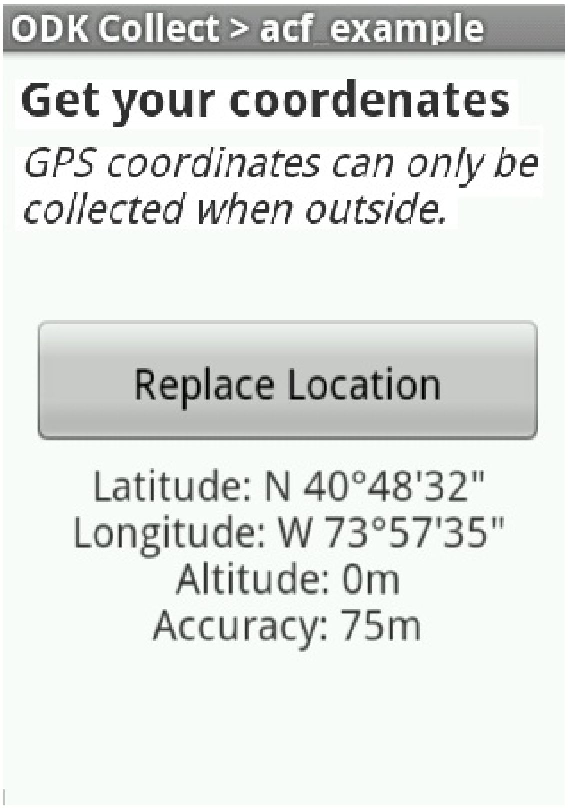
Let’s say we want to collect the gps coordinates and a picture within our survey. To make this tutorial easier to read we will switch from the Excel file screenshots to a tabular notation.

Excel 'survey' worksheet



	A	B	C
1	type	name	label
9	datetime	datenow	Enter Date & Time
10	geopoint	geopoint	Get your coordenates

Notice how the Excel ‘a_worksheet_name’ worksheet preceding the table specifies on which worksheet of the workbook we are working on (we will be working on the Excel ‘survey’ worksheet for now but other worksheets will be added as we develop a more advanced survey).



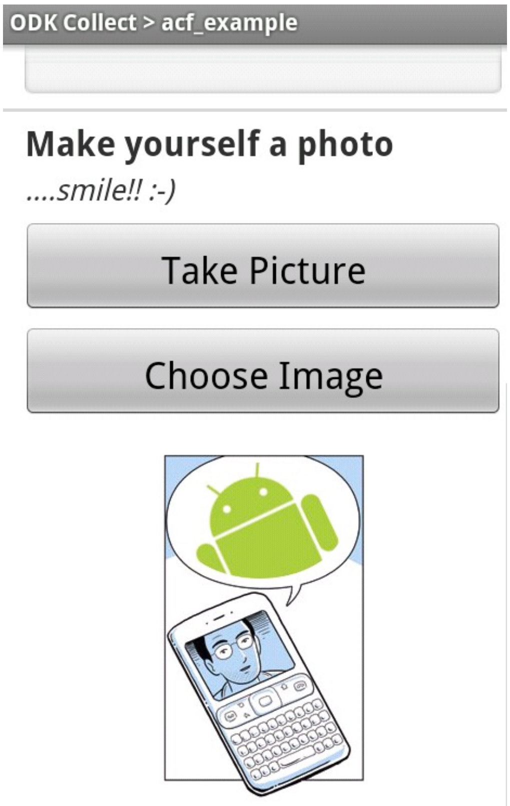
ODK Collect > acf_example

Get your coordenates

GPS coordinates can only be collected when outside.

Latitude: N 40°48'32"
Longitude: W 73°57'35"
Altitude: 0m
Accuracy: 75m

A geopoint type



ODK Collect > acf_example

Make yourself a photo

....smile!! :-)

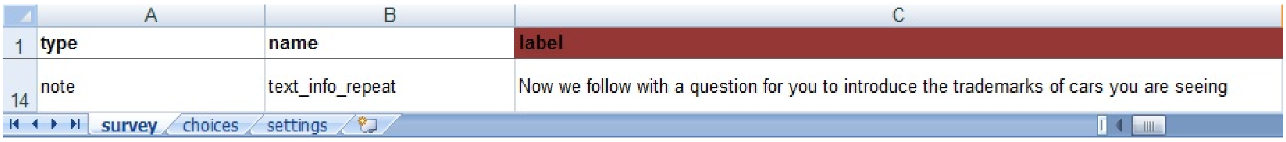
An image type

NOTES AND DATES

Notes

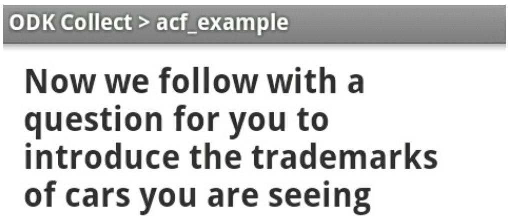
Another question 'type' that can be very useful to provide information to your enumerator while conducting a survey are notes. Notes are no-input text prompts which can give the enumerator information to read to the respondent, remind her that she is now moving into a new section of the survey, or any other hints or messages to respond to.

Excel 'survey' worksheet



	A	B	C
1	type	name	label
14	note	text_info_repeat	Now we follow with a question for you to introduce the trademarks of cars you are seeing

This is how a note would look like this on the Android device screen.



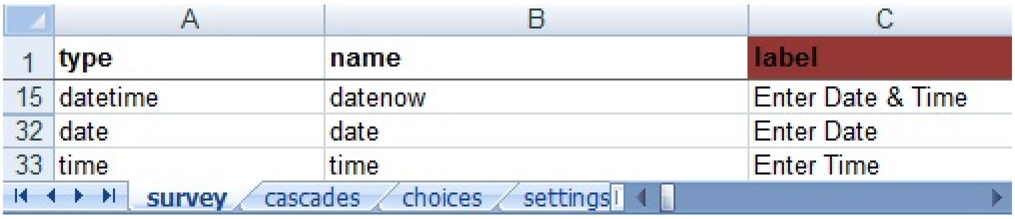
ODK Collect > acf_example

Now we follow with a question for you to introduce the trademarks of cars you are seeing

Dates

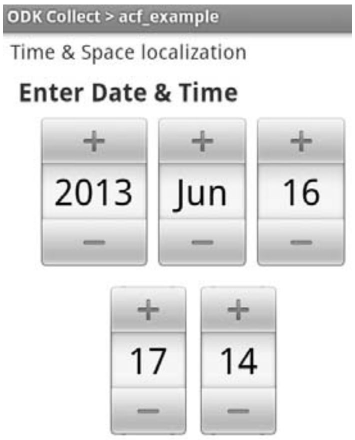
Date and time questions are fairly intuitive; they help collect a specific calendar date or time. You can collect it together using type "datetime" or separately type "date" and "time"

Excel 'survey' worksheet



	A	B	C
1	type	name	label
15	datetime	datenow	Enter Date & Time
32	date	date	Enter Date
33	time	time	Enter Time

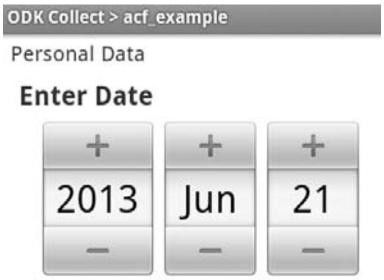
Date and Time as separated questions would look like this on the Android smartphone.



ODK Collect > acf_example

Time & Space localization

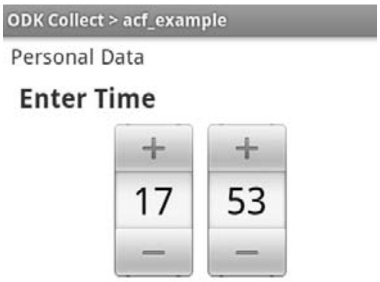
Enter Date & Time



ODK Collect > acf_example

Personal Data

Enter Date



ODK Collect > acf_example

Personal Data

Enter Time

CHOICE QUESTIONS

As we continue with the household survey, we are interested in the gender of the respondent, the number of people living in the household and the different sources of water used in the household. For these questions, we could simply ask text and integer questions, but often it is useful to standardize data by asking enumerators multiple choice questions. By using a list of predefined choices in survey questions, we can increase the value of data for quantitative analysis later. This technique obviates the need for "coding" data in many cases, and reduces errors introduced by typos and the need to interpret data.

Choice Questions

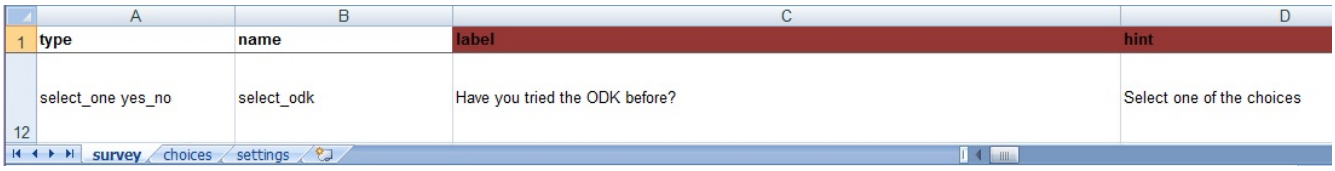
ODK Collect has support for both 'select one option' ('select_one') and 'select several options that apply ('select_multiple') questions. Writing a multiple or single choice questions requires some additional steps to the other types we have used up to now. We need to specify what the predefined list of choices will be!

select_one [choices]	Single choice question; only one answer can be selected (the [] notation indicates that you will need to replace [choices] by a name in the choices sheet).
select_multiple [choices]	Multiple choice question; multiple answers can be selected (the [] notation indicates that you will need to replace [choices] by a name in the choices sheet).

1. Add a second worksheet called 'choices' (the name of this new worksheet has to be exactly 'choices' in lowercase) to our Excel workbook.
2. In this new worksheet create three columns 'list name' (the name of the list of alternative choices for this multiple option question and will need to be referenced from the Excel 'survey' worksheet),'name' (a unique identifier of each choice alternative that contains no spaces or special characters) and 'label' (the text that will be displayed on the phone).
3. In the Excel 'survey' worksheet create a new row with the 'type' set to 'select_one' or 'select_multiple' followed by a space and replace [choices] by the name of the list of alternative choices as defined in the Excel 'choices' worksheet under the column 'list name'.
4. Populate the Excel 'choices' worksheet with all the options that are anticipated.

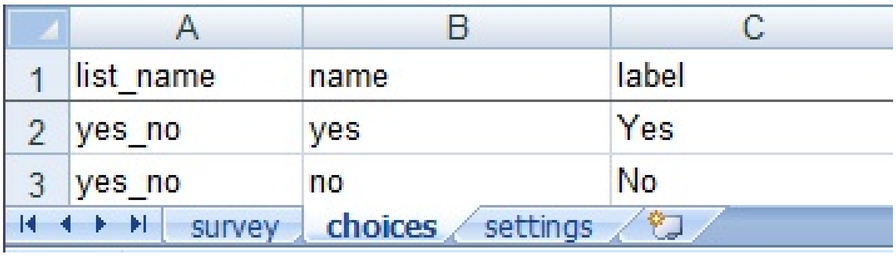
Here is an example of a ‘select one’ question to restrict the choices to ‘Yes’ and ‘No’:

Excel 'survey' worksheet



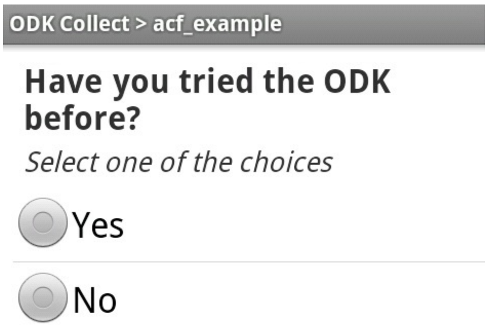
	A	B	C	D
1	type	name	label	hint
12	select_one yes_no	select_odk	Have you tried the ODK before?	Select one of the choices

Excel 'choices' worksheet



	A	B	C
1	list_name	name	label
2	yes_no	yes	Yes
3	yes_no	no	No

This is how it would look like on the Android device´s screen:



ODK Collect > acf_example

Have you tried the ODK before?

Select one of the choices

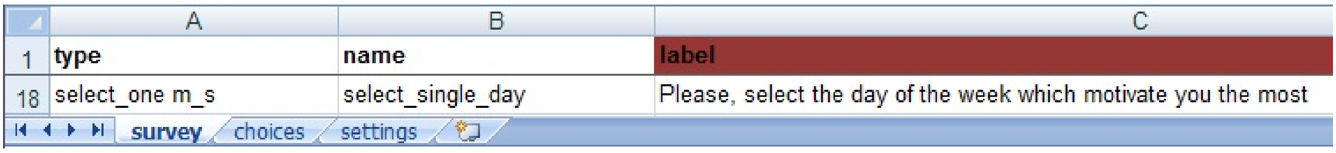
○ Yes

○ No

Note that yes_no appears in both worksheets, this is how the elements are linked between the “choices” list and the “survey” sheet. A choice list_name can be re-used in multiple survey questions.

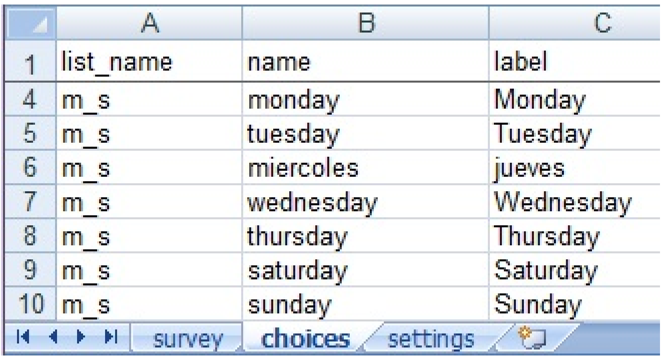
A more elaborate single select questions that only accepts one selection could be as the example that follows

Excel 'survey' worksheet



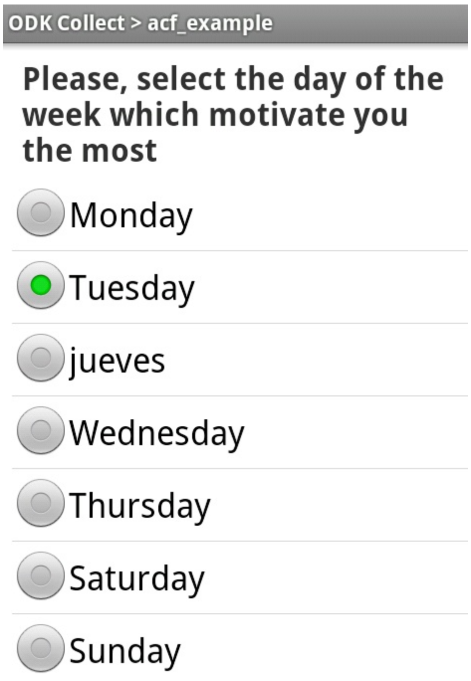
	A	B	C
1	**type**	**name**	**label**
18	select_one m_s	select_single_day	Please, select the day of the week which motivate you the most

Excel 'choices' worksheet



	A	B	C
1	list_name	name	label
4	m_s	monday	Monday
5	m_s	tuesday	Tuesday
6	m_s	miercoles	jueves
7	m_s	wednesday	Wednesday
8	m_s	thursday	Thursday
9	m_s	saturday	Saturday
10	m_s	sunday	Sunday

This is how it would look like on the Android device´s screen:



ODK Collect > acf_example

Please, select the day of the week which motivate you the most

○ Monday

● Tuesday

○ jueves

○ Wednesday

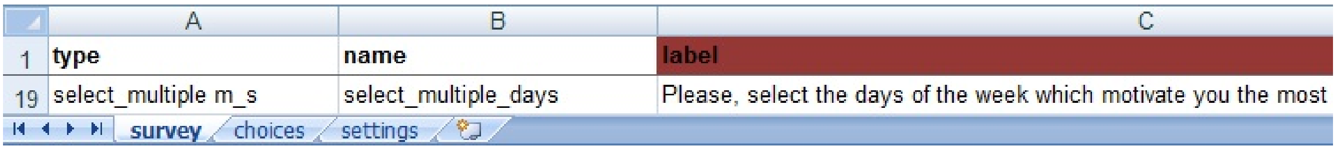
○ Thursday

○ Saturday

○ Sunday

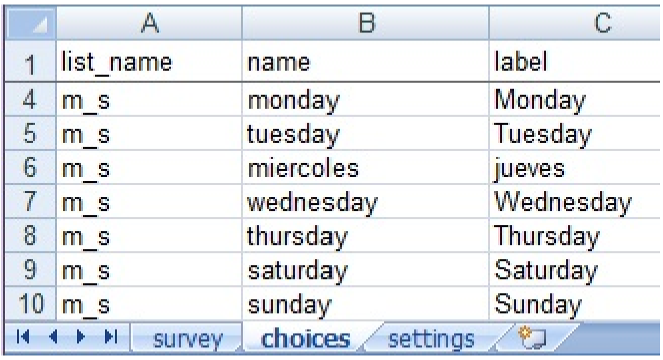
We can also add multiple choices questions that allow multiple answers to be selected. For example in our survey it could be the days of the week which motivates the respondent the most (could be one, two, all of them...).

Excel 'survey' worksheet



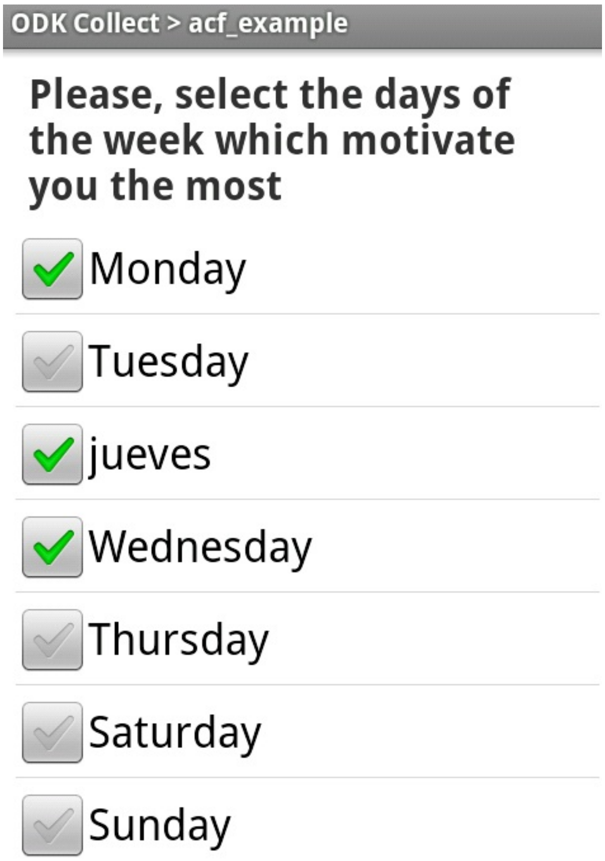
	A	B	C
1	**type**	**name**	**label**
19	select_multiple m_s	select_multiple_days	Please, select the days of the week which motivate you the most

Excel 'choices' worksheet



	A	B	C
1	list_name	name	label
4	m_s	monday	Monday
5	m_s	tuesday	Tuesday
6	m_s	miercoles	jueves
7	m_s	wednesday	Wednesday
8	m_s	thursday	Thursday
9	m_s	saturday	Saturday
10	m_s	sunday	Sunday

This is how it would look like on the Android device´s screen:



Please, select the days of the week which motivate you the most

☑ Monday

☐ Tuesday

☑ jueves

☑ Wednesday

☐ Thursday

☐ Saturday

☐ Sunday

THINGS TO REMEMBER FOR CHOICE QUESTIONS:

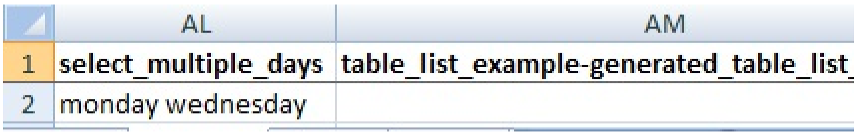
- The second worksheet should be called 'choices' and not 'Choices' 'choicelist' or any other alternative.

- Make sure you don't forget the first row and the headings 'list name', 'name' and 'label'.

- The name of your choices list in the Excel 'choices' worksheet (under 'list name' in the choices survey) has to be used again right after the 'select_one' or 'select_multiple' question type in under the column 'type' in the Excel 'survey' worksheet.

- The name of each choice, the content of the 'name' column of the Excel 'choices' worksheet cannot contain spaces or special characters and will appear in the survey data results file.

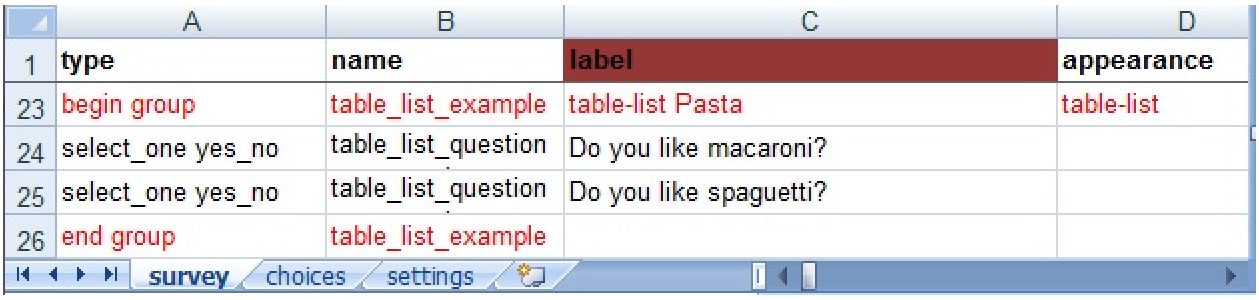
This is how the survey exported data for multiple questions will look like. All the selected answers appear on the same cell.



	AL	AM
1	select_multiple_days	table_list_example-generated_table_list_
2	monday wednesday	

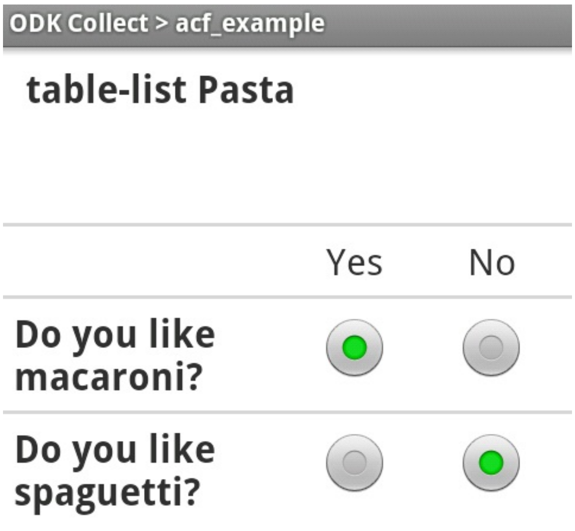
This could be a problem for data processing purposes ; in order to overcome this issue we propose an alternative option to do with the multiple choice questions. By creating a group with "table-list" in the appearance column will show the choices as a horizontal, more compact, list. If we make the different options into "single select" questions with "yes" or "no" answers we have an easy way process the exported results.

Excel 'survey' worksheet



	A	B	C	D
1	type	name	label	appearance
23	begin group	table_list_example	table-list Pasta	table-list
24	select_one yes_no	table_list_question	Do you like macaroni?	
25	select_one yes_no	table_list_question	Do you like spaguetti?	
26	end group	table_list_example		

This is how it would look like on the Android device´s screen:

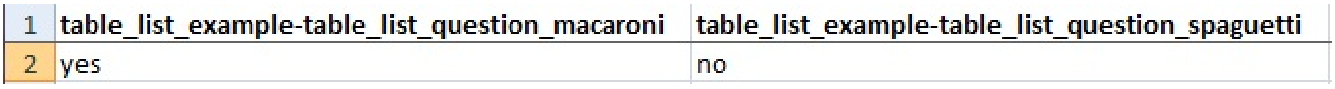


ODK Collect > acf_example

table-list Pasta

	Yes	No
Do you like macaroni?	●	○
Do you like spaguetti?	○	●

This is how it would appear once the data is exported.



1	table_list_example-table_list_question_macaroni	table_list_example-table_list_question_spaguetti
2	yes	no

*Notice that for each “**select one**” type of questions the responses are exported in one column named as defined under the ‘name’ column in the Excel 'survey' worksheet. The content of the cells is the name of each alternative as defined in the Excel 'choices' worksheet under the ‘name’ column.*

For “select multiple” questions one column is created for each option defined in the Excel 'choices' worksheet and the name is the name of the question as defined under the ‘name’ column in the Excel 'survey' worksheet followed by ‘/’ and the name of each choice as defined in the ‘name’ column of the Excel 'choices' worksheet. The content of the cells is TRUE or FALSE depending on if that particular option was selected or not.

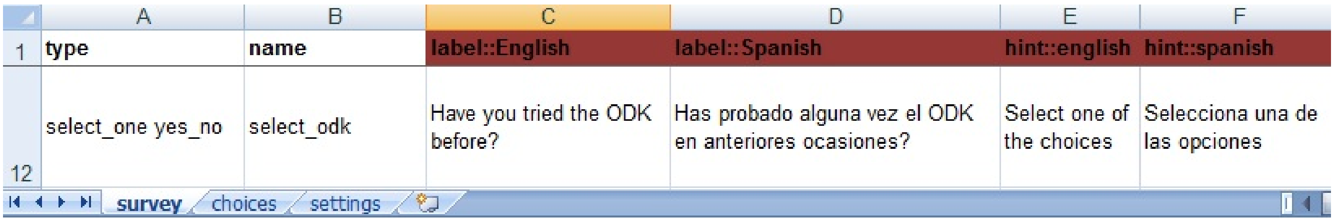
INTERMEDIATE SURVEY AUTHORING

We have reviewed most data types and you should be comfortable with the creation of basic surveys. We will now focus on strengthening the logic flow of the survey questions and we will introduce conditional elements to increase data quality and validation.

Multiple Language Support

It is possible to add multiple languages to a survey using in xlsform. To do this, you have to essentially create a difference label column for each of your languages. For example a form with the columns label::English and label::French it will allow the surveyors to choose between English and French on ODK Collect, in order to select a different language press the Menu key and the "Change Language" option. For the form below, "English" and "Spanish" will show up as the possible options.

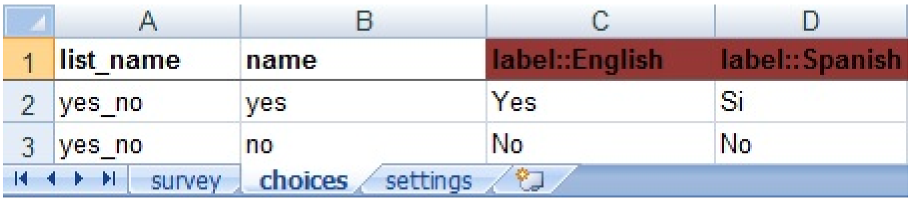
Excel 'survey' worksheet



	A	B	C	D	E	F
1	type	name	label::English	label::Spanish	hint::english	hint::spanish
12	select_one yes_no	select_odk	Have you tried the ODK before?	Has probado alguna vez el ODK en anteriores ocasiones?	Select one of the choices	Selecciona una de las opciones

Make sure that your column headers in the "choices" sheet match those in the "survey" sheet (i.e. there will be errors if you columns are "label::English" on the survey and "label" on the choices worksheet)

Excel 'choices' worksheet



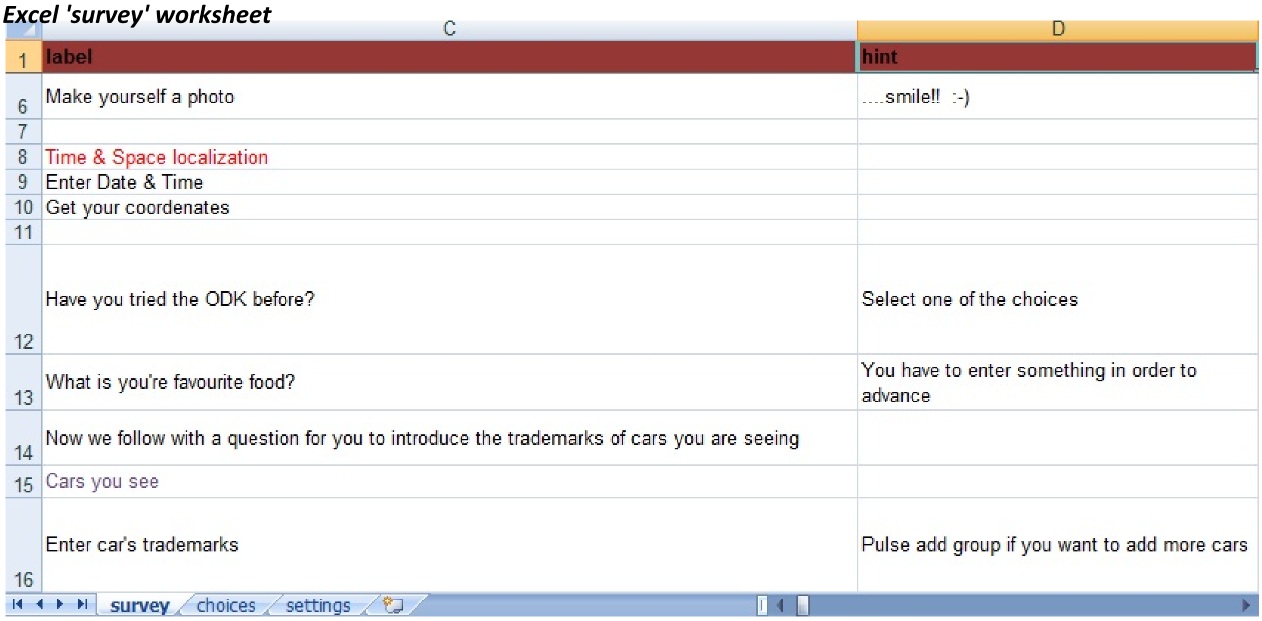
	A	B	C	D
1	list_name	name	label::English	label::Spanish
2	yes_no	yes	Yes	Si
3	yes_no	no	No	No

Refer to the "ODK ACF-E How to adapt a form into local language" document(annexed to this toolbox) to learn what columns headers can accept a language modification.

Hints

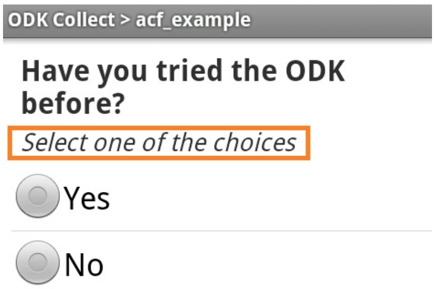
When asking questions, it often makes sense to give enumerators some extra guidance. You can already do this with notes, but what about something simpler that doesn't require an extra screen? The answer is "hints," which are listed along-side the question in italics font (main questions appears in bold font). To use them, just create new column in your survey worksheet named 'hint', and add your hints.

Excel 'survey' worksheet



	C	D
1	label	hint
6	Make yourself a photo	....smile!! :-)
7		
8	Time & Space localization	
9	Enter Date & Time	
10	Get your coordenates	
11		
12	Have you tried the ODK before?	Select one of the choices
13	What is you're favourite food?	You have to enter something in order to advance
14	Now we follow with a question for you to introduce the trademarks of cars you are seeing	
15	Cars you see	
16	Enter car's trademarks	Pulse add group if you want to add more cars

Question "Have you tried the ODK before? Look like this on the Android device´s screen:



ODK Collect > acf_example

Have you tried the ODK before?

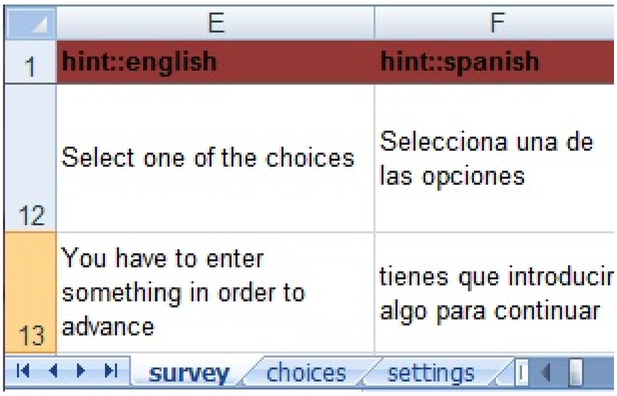
Select one of the choices

○ Yes

○ No

You can also have your hints in different languages. For example, the above form could use two additional columns "hint::English" and "hint::Spanish" to provide hints in both of those languages.

Excel 'survey' worksheet



	E	F
1	hint::english	hint::spanish
12	Select one of the choices	Selecciona una de las opciones
13	You have to enter something in order to advance	tienes que introducir algo para continuar

Metadata

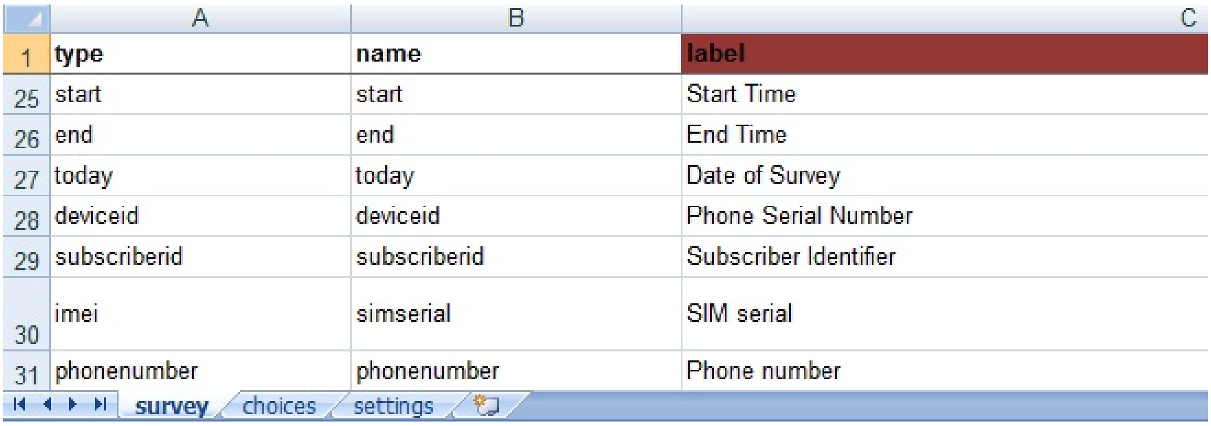
In prevision for extensive data analysis, it is useful to collect additional data for administrative purposes. For example, you might want to know the total and average time it takes enumerators to conduct the full survey. During the pilot stage, identifying which phone was used to collect certain GPS coordinates might help the detection of faulty GPS systems (important at scale). Other similar information can be used for other administrative and review processes; the “start” and “end” date-time fields are particularly helpful when checking for duplicate information entry.

ODK Collect makes a number of metadata fields available for collection:

start	*Start date and time of the survey.*
end	*End date and time of the survey.*
today	*Day of the survey.*
deviceid	*IMEI (International Mobile Equipment Identity)*
subscriberid	*IMSI (International Mobile Subscriber Identity)*
imei	*SIM serial number (if available).*
phonenumber	*Phone number (if available).*

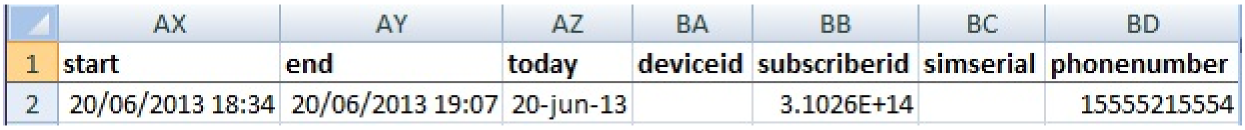
If we wanted our survey to collect all of this metadata we would add the following rows at the top of the survey (they can be anywhere but it is a best practice to put metadata fields at the beginning of the file):

Excel 'survey' worksheet



	A	B	C
1	**type**	**name**	**label**
25	start	start	Start Time
26	end	end	End Time
27	today	today	Date of Survey
28	deviceid	deviceid	Phone Serial Number
29	subscriberid	subscriberid	Subscriber Identifier
30	imei	simserial	SIM serial
31	phonenumber	phonenumber	Phone number

These questions do not appear on the phone at all, but will be automatically included in the data ODK Collect writes to store and transmit the survey data. To see these fields at work, fill out a new blank form on your Android Smartphone, save it and send it to the server. When you download the data you will see that new columns have been added to the exported survey data.



	AX	AY	AZ	BA	BB	BC	BD
1	**start**	**end**	**today**	**deviceid**	**subscriberid**	**simserial**	**phonenumber**
2	20/06/2013 18:34	20/06/2013 19:07	20-jun-13		3.1026E+14		15555215554

Conditional survey constructs

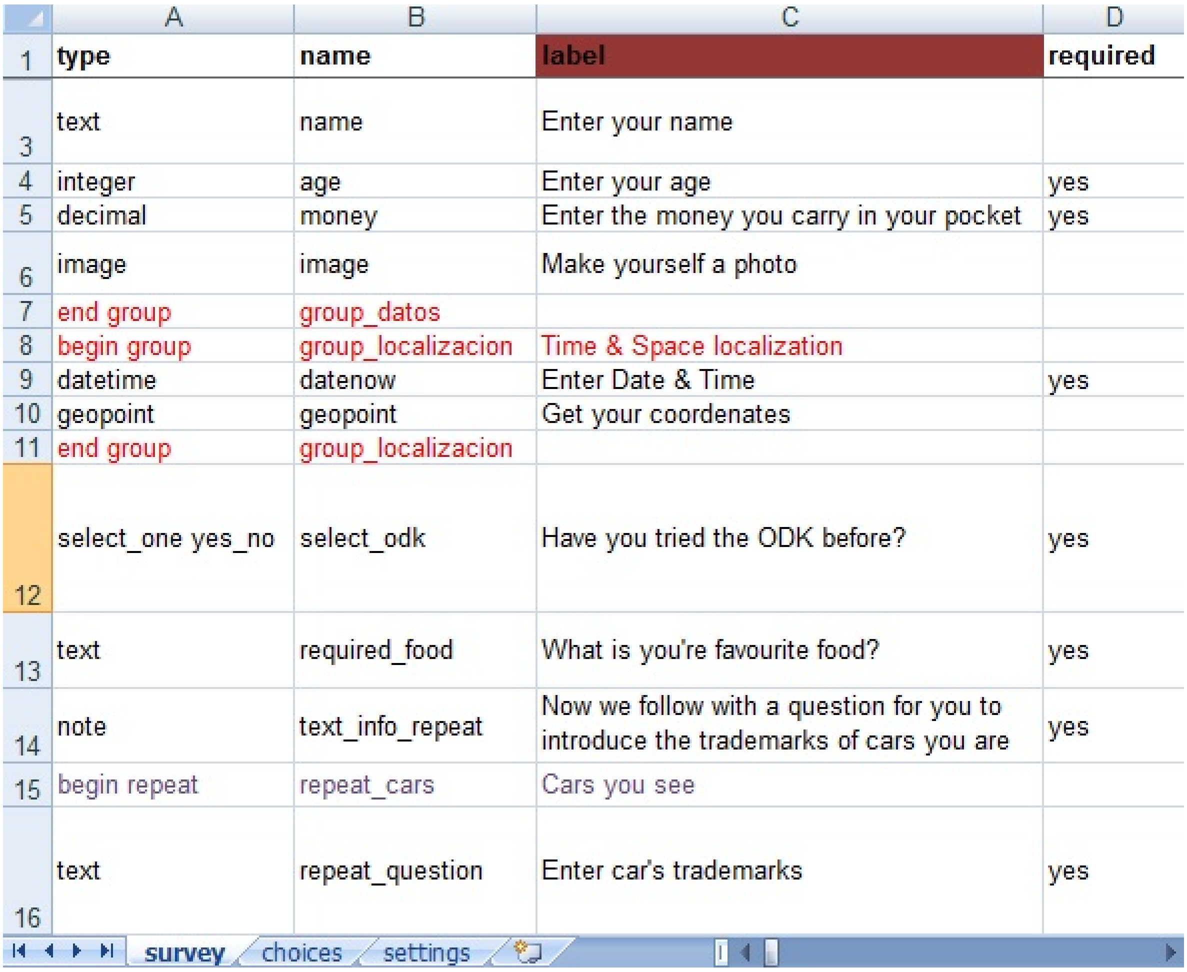
In order to reduce input errors ODK Collect has some options to control and restrict the way data is entered by enumerators. Three types of constructs are available:

1. *Requiring that certain questions are filled with data before moving forward.*
2. *Constraining the values of data entered in certain questions.*
3. *Referencing to previous questions*
4. *Skipping non-relevant questions depending on the answers to previous questions (avoid asking what "other" if the respondent did not answered "Other , please specify").*

1. Required questions

ODK allows the survey author to designate certain questions as "required" questions. Normally, enumerators are able to "skip" answering any question by simply swiping forward on a page. However, many surveyors prefer that enumerators enter a specific value (such as "-999" for "Not available") rather than simply skip question. ODK allows you to enforce this; simply column labeled 'required' and filling the cells with the reserved word 'yes' to the questions that you consider critical to have in your data.

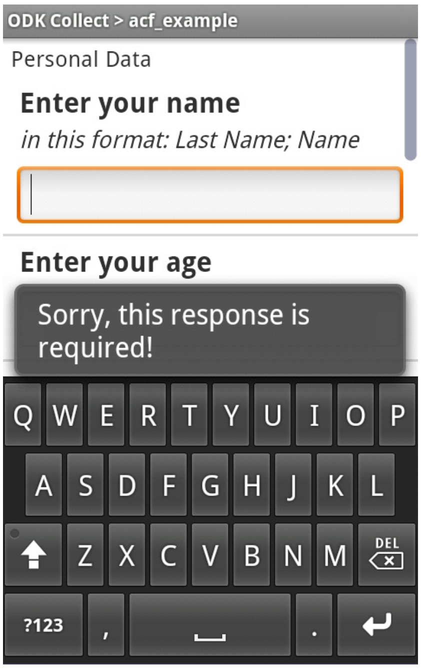
Excel 'survey' worksheet



	A	B	C	D
1	**type**	**name**	**label**	**required**
3	text	name	Enter your name	
4	integer	age	Enter your age	yes
5	decimal	money	Enter the money you carry in your pocket	yes
6	image	image	Make yourself a photo	
7	end group	group_datos		
8	begin group	group_localizacion	Time & Space localization	
9	datetime	datenow	Enter Date & Time	yes
10	geopoint	geopoint	Get your coordenates	
11	end group	group_localizacion		
12	select_one yes_no	select_odk	Have you tried the ODK before?	yes
13	text	required_food	What is you're favourite food?	yes
14	note	text_info_repeat	Now we follow with a question for you to introduce the trademarks of cars you are	yes
15	begin repeat	repeat_cars	Cars you see	
16	text	repeat_question	Enter car's trademarks	yes

If you test the form, and try to swipe ahead on (for example) question "Enter your name", you will not be able to move forward, and instead, get a prompt saying "Sorry, this response is required!"

on the Android device´s screen:



ODK Collect > acf_example

Personal Data

Enter your name

in this format: Last Name; Name

Enter your age

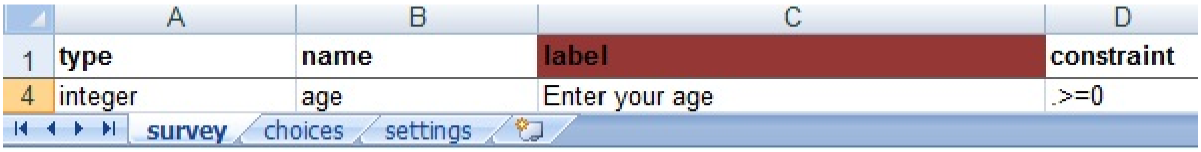
Sorry, this response is required!

Be careful when adding conditional constructs to a survey or the enumerator might find herself incapable of finishing the survey (if the GPS on the phone is not working and you have required the collection of geographic coordinates, enumerators might not be able to move forward to the next sections of the survey). It is also important to test every survey with conditional constructs as much as possible before deployment. For text and integer fields, be sure to specify what should be entered in case the respondent is not able to answer the enumerator.

2. Constraining Responses

Certain questions only have a limited range of possible answers that can be considered as valid. If, for example, you ask a household member his/her age , the response can only be a number higher than 0. ODK Collect handles this type of data validation with the use of the 'constraint' column.

Excel 'survey' worksheet



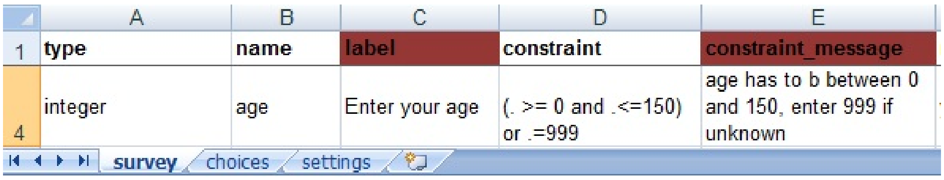
	A	B	C	D
1	type	name	label	constraint
4	integer	age	Enter your age	.>=0

Here we've put a constraint on age so that negative values are not accepted. ODK Collect will not allow the enumerator to swipe to the next question if she enters -5. These constraints can help protect from typos and other unintentional errors.

The constraint formulas can be fairly complex, accommodating the use of and, or and not operators.

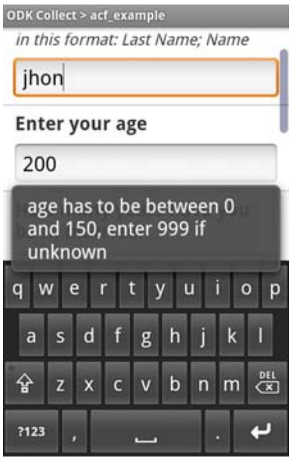
Let's say that our age limit is 150 (much lower than the oldest person every known to exist), and that we want to allow for the enumerator to enter 999 in case the respondent cannot answer the question. The constraint can be written as follows.

Excel 'survey' worksheet



	A	B	C	D	E
1	type	name	label	constraint	constraint_message
4	integer	age	Enter your age	(. >= 0 and .<=150) or .=999	age has to b between 0 and 150, enter 999 if unknown

On the Android device´s screen:



ODK Collect > acf_example

in this format: Last Name; Name

jhon

Enter your age

200

age has to be between 0 and 150, enter 999 if unknown

In fact, we have even added a new column labeled 'constraint_message' that specified the message displayed on the screen (by default the popup message will be "Sorry, this response is invalid")

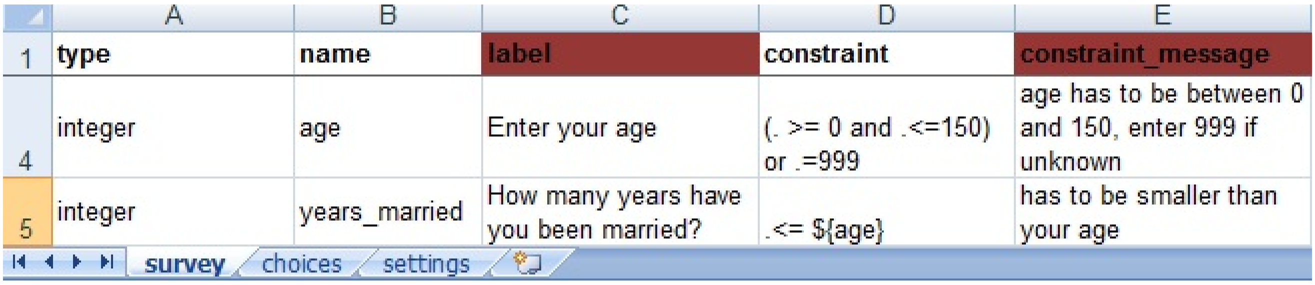
3. Referencing a previous question

We can refer to previous answers to questions in order to set new constrains depending on the answer inserted by the respondent.

In the following example we will add a question" How many years have you been married?" the number of years on to this question has to be smaller than or equal (quite unlikely) to the total number of living years.

XLSform can let you reference the results of a previous question by using a very specific syntax: ${name_of_previous_question} will actually fetch the value entered by the enumerator in the question that has a name (as defined in the 'name' column of the Excel 'survey' worksheet). Notice that the special character '**.**' will be interpreted by ODK Collect as a reference to the current question (as if we had typed${name_of_current_question}).

Excel 'survey' worksheet

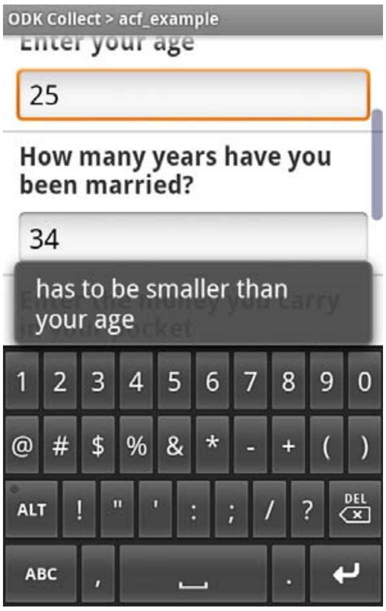


	A	B	C	D	E
1	type	name	label	constraint	constraint_message
4	integer	age	Enter your age	(. >= 0 and .<=150) or .=999	age has to be between 0 and 150, enter 999 if unknown
5	integer	years_married	How many years have you been married?	.<= ${age}	has to be smaller than your age

survey
choices
settings

In this example the answer to "How many years have you been married?" is constrained to be less than or equal to the answer given to question "Enter your age".

on the Android device´s screen:



ODK Collect > acf_example

Enter your age

25

How many years have you been married?

34

has to be smaller than your age

Advanced: The column 'constraint' takes an XPath expression. The ${QUESTION_NAME} notation generates a valid XPath formula. A question is accepted if the 'constraint' column evaluates to true.

4. Asking Relevant Questions

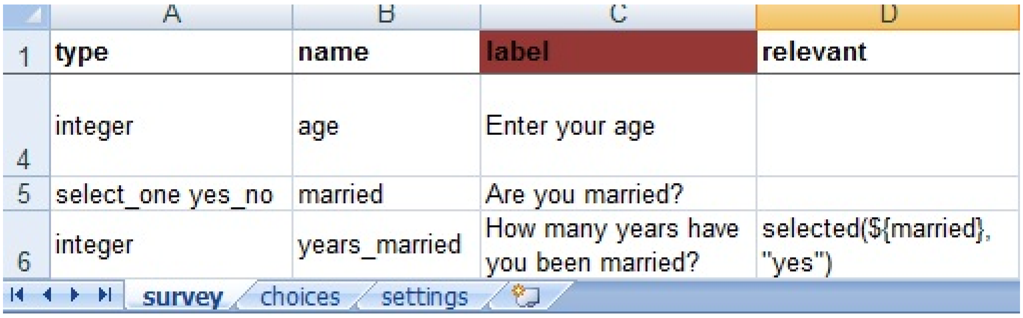
Most of surveys have questions that should be asked depending on the answer given to previous questions. One great feature of ODK Collect is the ability to only present a question to the enumerator if a condition, based on the response to a previous question, is true.

For example, if you presented a multiple choice question and the answer by the interviewee does not correspond to any of the pre-programmed choices, you might want to offer ‘Other’ as an alternative. If this is the case you would need to ask the interviewee to specify what other means. To do this we add the ‘relevant’ column and enter a condition that specifies when the questions is relevant and should be presented to the enumerator.

You can interpret this conditions as: “Only ask this question if the answer to the previous question: name_of_question (as defined in the ‘name’ column of the Excel 'survey' worksheet) was equal to ‘option_selected’ (as defined in the ‘name’ column of the Excel 'choices' worksheet)”.

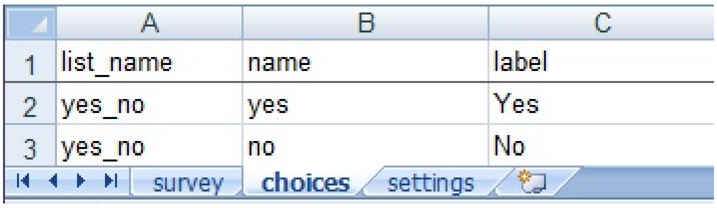
Notice that in the case of ‘select_multiple’ questions the syntax is a bit different and you can actually specify relevant conditions for each one of the options selected. Below is an example of how to do this using XLSform.

Excel 'survey' worksheet



	A	B	C	D
1	**type**	**name**	**label**	**relevant**
4	integer	age	Enter your age	
5	select_one yes_no	married	Are you married?	
6	integer	years_married	How many years have you been married?	selected(${married}, "yes")

Excel 'choices' worksheet

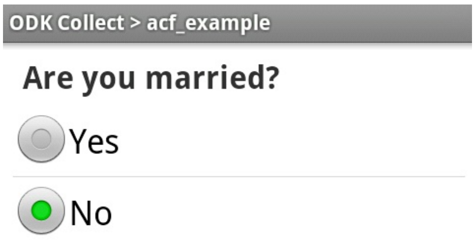


	A	B	C
1	list_name	name	label
2	yes_no	yes	Yes
3	yes_no	no	No

If in the question “Are you married?” the interviewee answers ‘Yes’ then the ODK Collect will move to question “How many years have you been married” The content in the ‘relevant’ column: selected(${married}, 'yes') actually means: “only ask this question if the answer to the question with ‘name’ equal “married” was the choice alternative with a ‘name’ equal to “yes”.

If in question “Are you married” the interviewee selects the option “No” then question “How many years have you been married?” should be skipped and “Enter the money you carry in your pocket” be presented on the phone.

On the Android device´s screen:

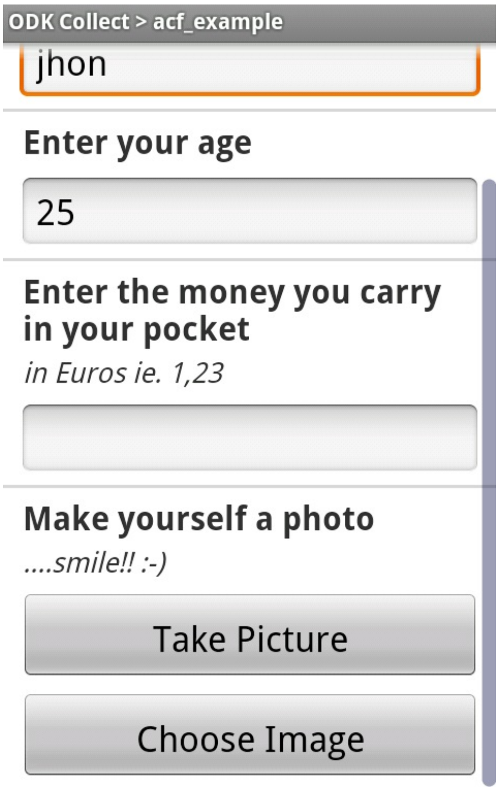


ODK Collect > acf_example

Are you married?

○ Yes

● No



ODK Collect > acf_example

jhon

Enter your age

25

Enter the money you carry in your pocket

in Euros ie. 1,23

Make yourself a photo

....smile!! :-)

You will learn how to skip more than one question at a time later on.

Important to remember:

For skip logic based on ‘select_one’ questions, you can also use the = (equals) construct: ${married}= yes. For more on proper XPath expressions, see this page: https://bitbucket.org/javarosa/javarosa/wiki/buildxforms

Advanced Survey Authoring

You should be comfortable now with the use of required, constraint and relevant. By now you should be able to author an intermediate survey that validates the data entered by the enumerator improving data quality.

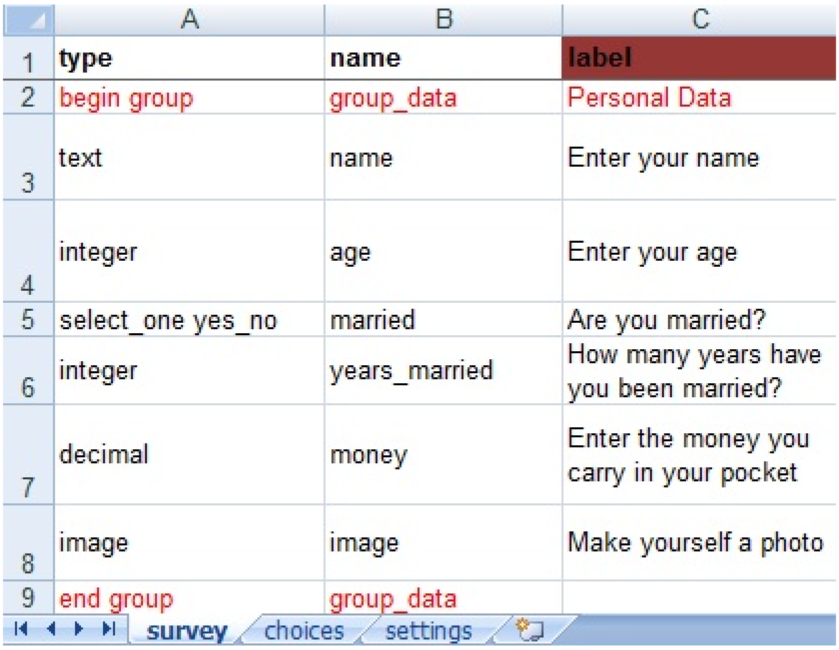
The use of relevant conditions is key to a successful survey so make sure you understand it is use before moving forward. Advanced survey authoring presents some miscellaneous syntax elements that will improve your survey.

We will cover the use of groups for skipping several questions and present groups of questions on the same page. We will also improve the formatting of our survey with the use of the 'appearance' column and finally we will present the cascading select functionality that will allow presenting cascading multiple choice selections with dependencies between them.

Grouping Questions

Grouping questions has several advantages; the most immediate one is that the group name will be presented to the enumerator as small text before each question. This can be helpful to remember in which section of the survey we are. Other advantages of groups will be explored in the next sections. To create a group of questions try the following:

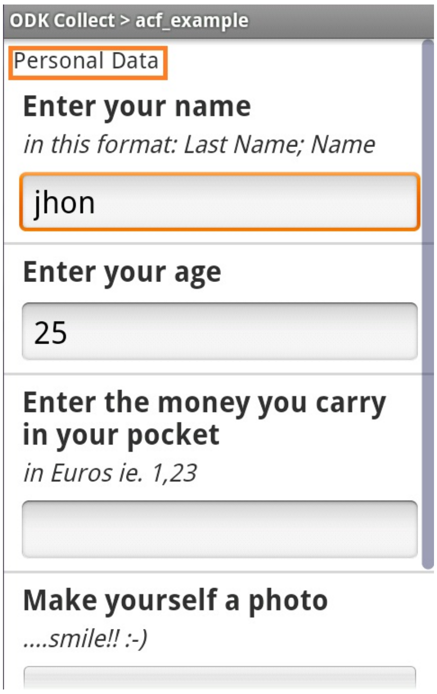
Excel 'survey' worksheet



	A	B	C
1	type	name	label
2	begin group	group_data	Personal Data
3	text	name	Enter your name
4	integer	age	Enter your age
5	select_one yes_no	married	Are you married?
6	integer	years_married	How many years have you been married?
7	decimal	money	Enter the money you carry in your pocket
8	image	image	Make yourself a photo
9	end group	group_data	

Using groups adds the 'label' content of the group construct to all questions pertaining to that group.

On the Android device´s screen:



ODK Collect > acf_example

Personal Data

Enter your name

in this format: Last Name; Name

jhon

Enter your age

25

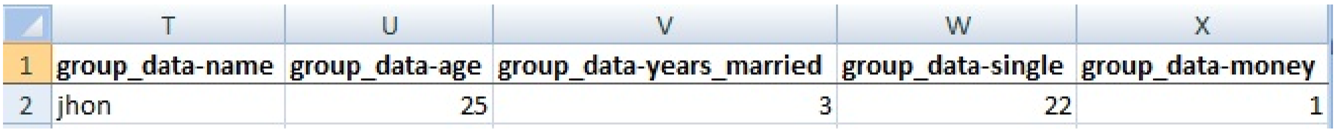
Enter the money you carry in your pocket

in Euros ie. 1,23

Make yourself a photo

....smile!! :-)

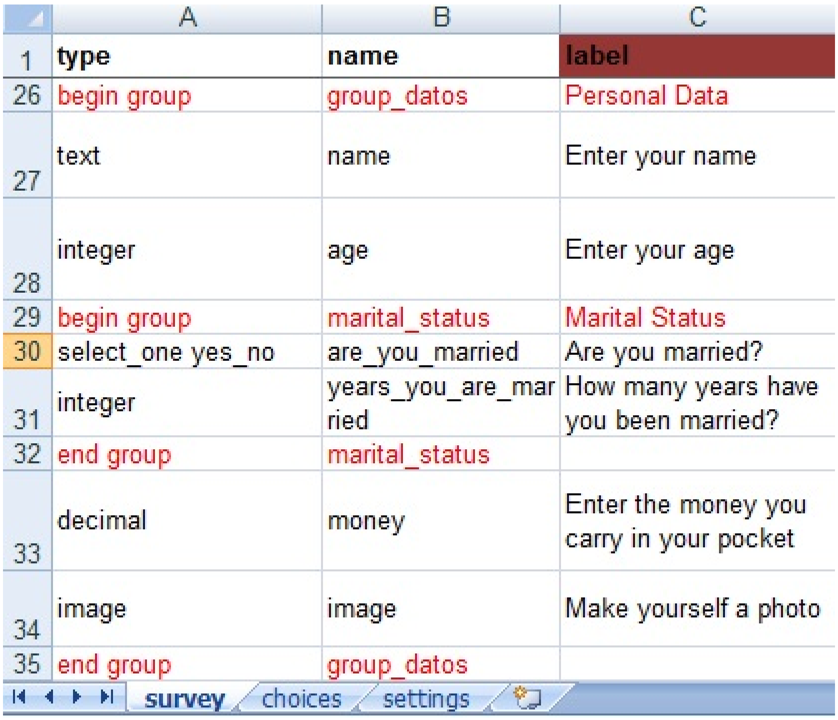
Using groups also changes how the survey data file columns are named when exported to .csv The group ‘name’ now precedes the question ‘name’ with a ‘-’ in between. Like “group_data-name”.



	T	U	V	W	X
1	**group_data-name**	**group_data-age**	**group_data-years_married**	**group_data-single**	**group_data-money**
2	jhon	25	3	22	1

If you have subsections within your sections, groups can be nested very easily. Here is an example, where we have grouped all questions asked to the respondent within section A:

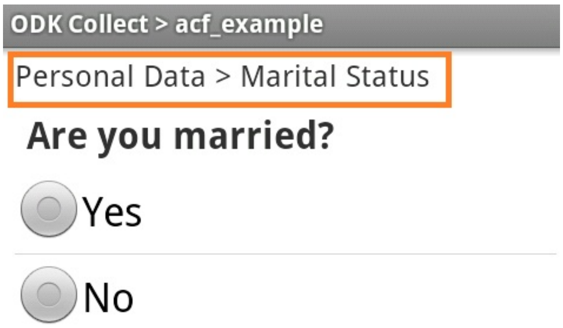
Excel 'survey' worksheet



	A	B	C
1	**type**	**name**	**label**
26	begin group	group_datos	Personal Data
27	text	name	Enter your name
28	integer	age	Enter your age
29	begin group	marital_status	Marital Status
30	select_one yes_no	are_you_married	Are you married?
31	integer	years_you_are_married	How many years have you been married?
32	end group	marital_status	
33	decimal	money	Enter the money you carry in your pocket
34	image	image	Make yourself a photo
35	end group	group_datos	

Notice how the nested group is shown on the phone with the parent group 'label' and the '>' character in the group text area now. This can be useful for subsections, just remember not to put too many nested groups or your phone screen will become illegible.

On the Android device´s screen:



ODK Collect > acf_example

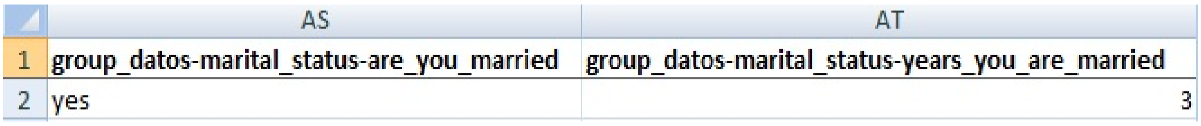
Personal Data > Marital Status

Are you married?

○ Yes

○ No

In the survey exported data result file notice how nested groups add the 'label' of the parent group to the column name for that variable. The answers for question "are you married" are now under "group_datos-marital_status-are_you_married".



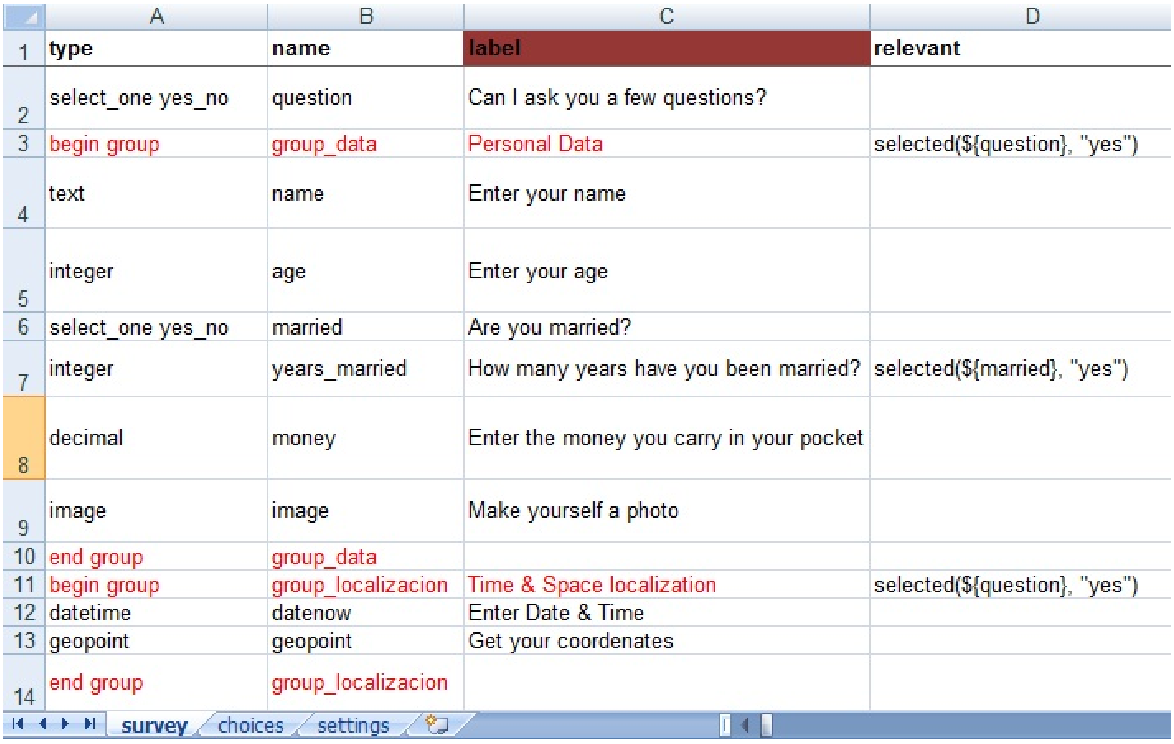
	AS	AT
1	group_datos-marital_status-are_you_married	group_datos-marital_status-years_you_are_married
2	yes	3

Skipping many questions at once

One of the most useful uses of groups is to skip a whole bunch of questions at once. In our survey, we only want to ask questions in section B if the respondent is over 18 years of age.

We will need to add a relevant condition telling ODK the following: "Only present this question if the answer to the question "Can I ask you a few questions?" is "Yes".

Excel 'survey' worksheet



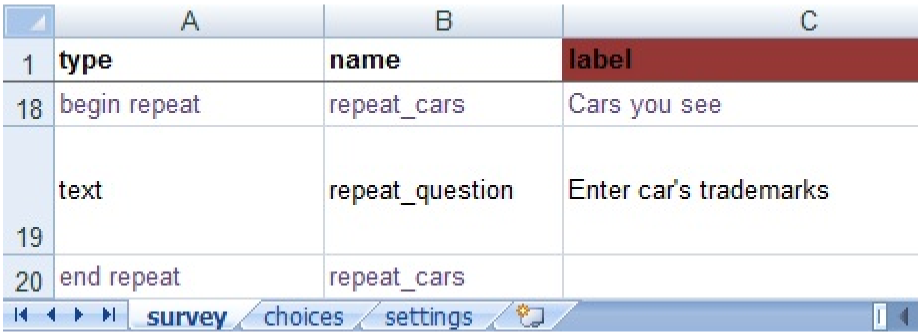
	A	B	C	D
1	type	name	label	relevant
2	select_one yes_no	question	Can I ask you a few questions?	
3	begin group	group_data	Personal Data	selected(${question}, "yes")
4	text	name	Enter your name	
5	integer	age	Enter your age	
6	select_one yes_no	married	Are you married?	
7	integer	years_married	How many years have you been married?	selected(${married}, "yes")
8	decimal	money	Enter the money you carry in your pocket	
9	image	image	Make yourself a photo	
10	end group	group_data		
11	begin group	group_localizacion	Time & Space localization	selected(${question}, "yes")
12	datetime	datenow	Enter Date & Time	
13	geopoint	geopoint	Get your coordenates	
14	end group	group_localizacion		

Without the use of groups you would need to add the relevant condition in each row. This would be cumbersome and difficult to maintain for the survey author. Using groups is a much more efficient way of performing this task.

Repeating a set of questions

You might also want to create a repeating set of questions. In our case we might be interested in the measurement of every child under 5 years old of the household. We cannot do this with existing tools, because we don't know the number of household members in advance. To do this, we use the following construct:

Excel 'survey' worksheet

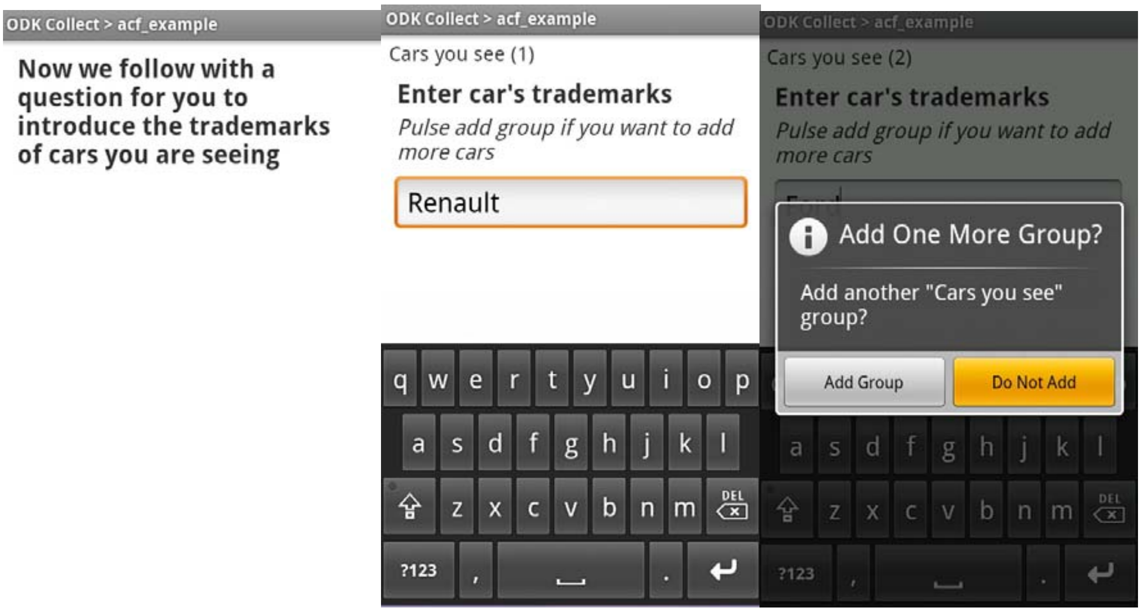


	A	B	C
1	type	name	label
18	begin repeat	repeat_cars	Cars you see
19	text	repeat_question	Enter car's trademarks
20	end repeat	repeat_cars	

This is an example using cars trademarks, with a text input field.

The phone will ask to enter a car´s trademark, and then ask if the enumerator wants to Add a new 'Cars you see" group.

On the Android device´s screen:



ODK Collect > acf_example

Now we follow with a question for you to introduce the trademarks of cars you are seeing

ODK Collect > acf_example

Cars you see (1)

Enter car's trademarks

Pulse add group if you want to add more cars

Renault

ODK Collect > acf_example

Cars you see (2)

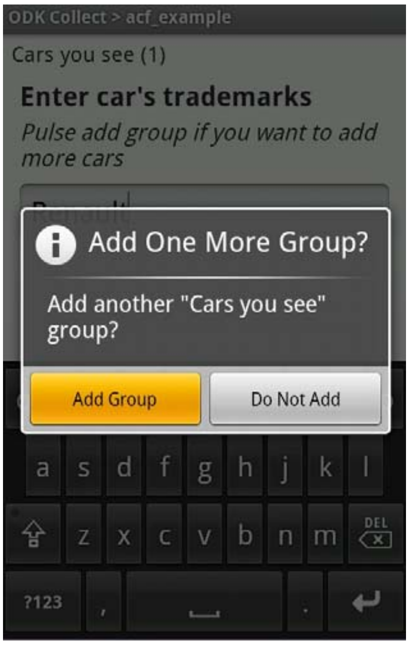
Enter car's trademarks

Pulse add group if you want to add more cars

Add One More Group?

Add another "Cars you see" group?

If we keep adding groups notice that in the group text area there is a counter (a number between parentheses) that tells us how many car´s trademarks we have added.



ODK Collect > acf_example

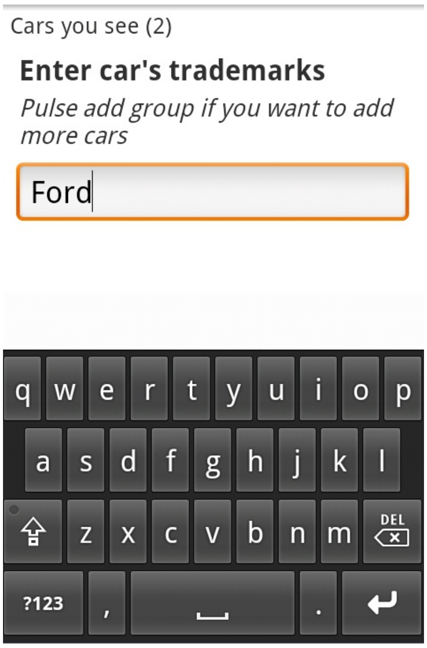
Cars you see (1)

Enter car's trademarks

Pulse add group if you want to add more cars

Add One More Group?

Add another "Cars you see" group?



Cars you see (2)

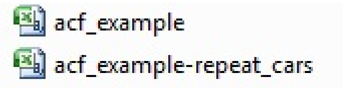
Enter car's trademarks

Pulse add group if you want to add more cars

Ford

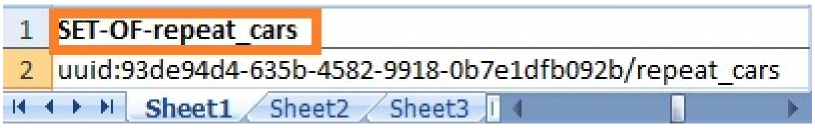
Note that the exported data for repeats comes in a variety of formats depending on the export you perform, and data within repeats can be difficult to perform analysis with. We recommend using repeats with care, and constructing a methodology around the use of that repeat data before doing large scale data collection with repeats.

When using ODK briefcase to export the data collected, the results appears in two different .csv files, one for the form and the other for the repeat groups



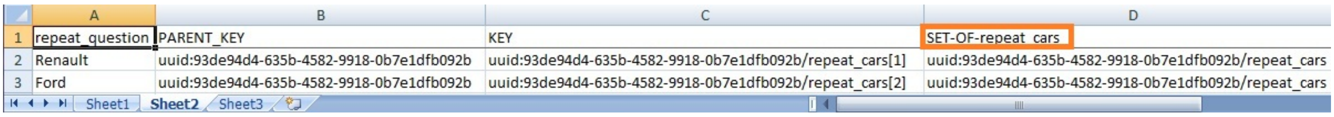
acf_example
acf_example-repeat_cars

Both are related trough the column SET-OF-"name of the repeat group" in this case "SET-OF-repeat_cars"



1	SET-OF-repeat_cars
2	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b/repeat_cars

(acf_example)



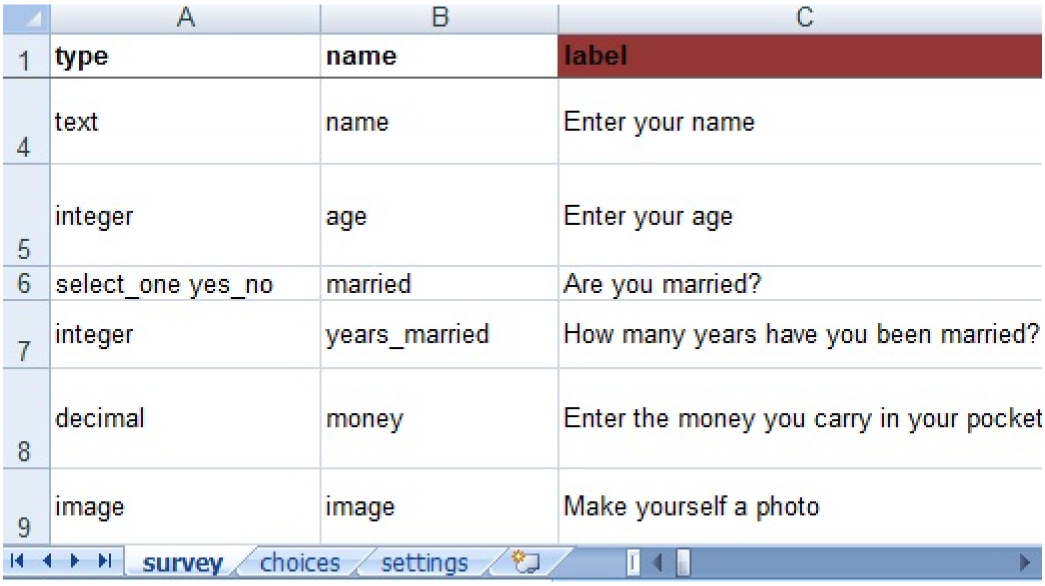
	A	B	C	D
1	repeat_question	PARENT_KEY	KEY	SET-OF-repeat_cars
2	Renault	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b/repeat_cars[1]	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b/repeat_cars
3	Ford	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b/repeat_cars[2]	uuid:93de94d4-635b-4582-9918-0b7e1dfb092b/repeat_cars

(acf_example-repeat_cars)

Special Formatting

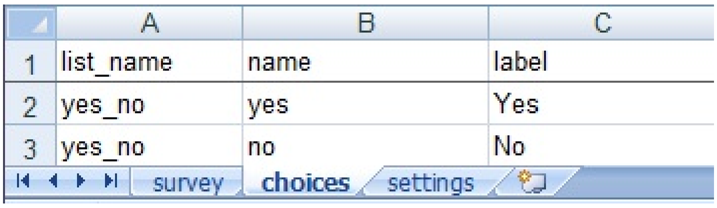
ODK Collect allows for certain changes in the formatting of device´s screens, for example you might want to show multiple choice “select_one” question as a list box instead of a long list of checkboxes. You might also want to present two questions on the same screen on the phone.

Excel 'survey' worksheet



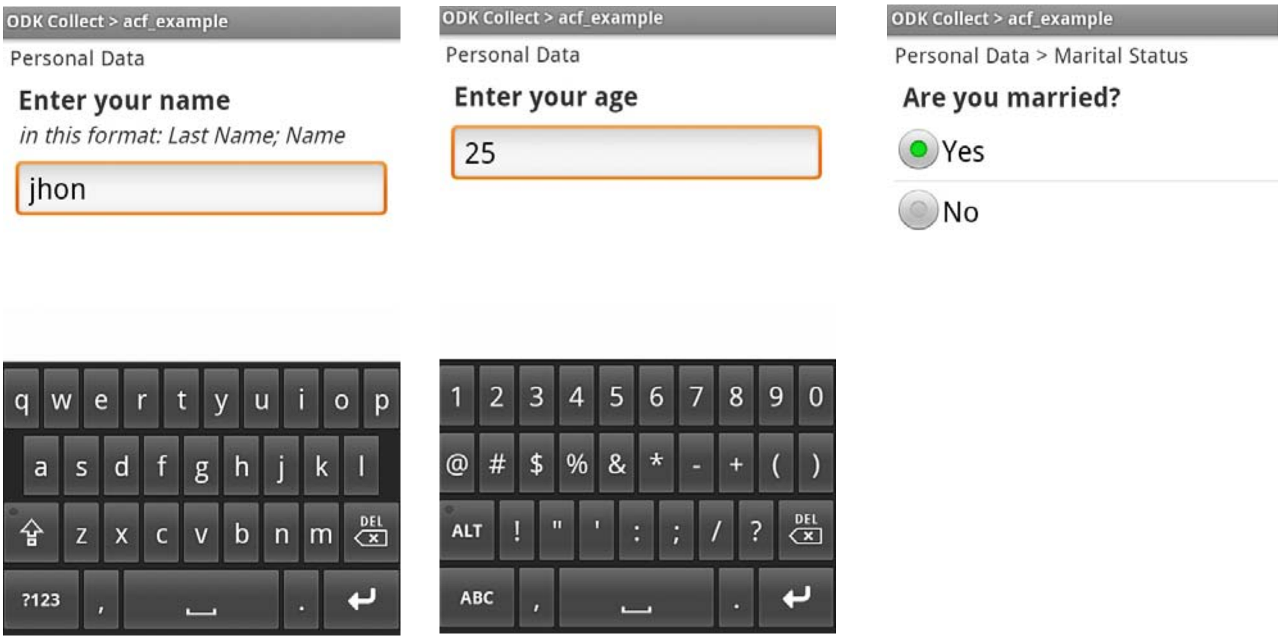
	A	B	C
1	**type**	**name**	**label**
4	text	name	Enter your name
5	integer	age	Enter your age
6	select_one yes_no	married	Are you married?
7	integer	years_married	How many years have you been married?
8	decimal	money	Enter the money you carry in your pocket
9	image	image	Make yourself a photo

Excel 'choices' worksheet



	A	B	C
1	list_name	name	label
2	yes_no	yes	Yes
3	yes_no	no	No

Witch looks like this on the device’s screen:



ODK Collect > acf_example

Personal Data

Enter your name

in this format: Last Name; Name

jhon

ODK Collect > acf_example

Personal Data

Enter your age

25

ODK Collect > acf_example

Personal Data > Marital Status

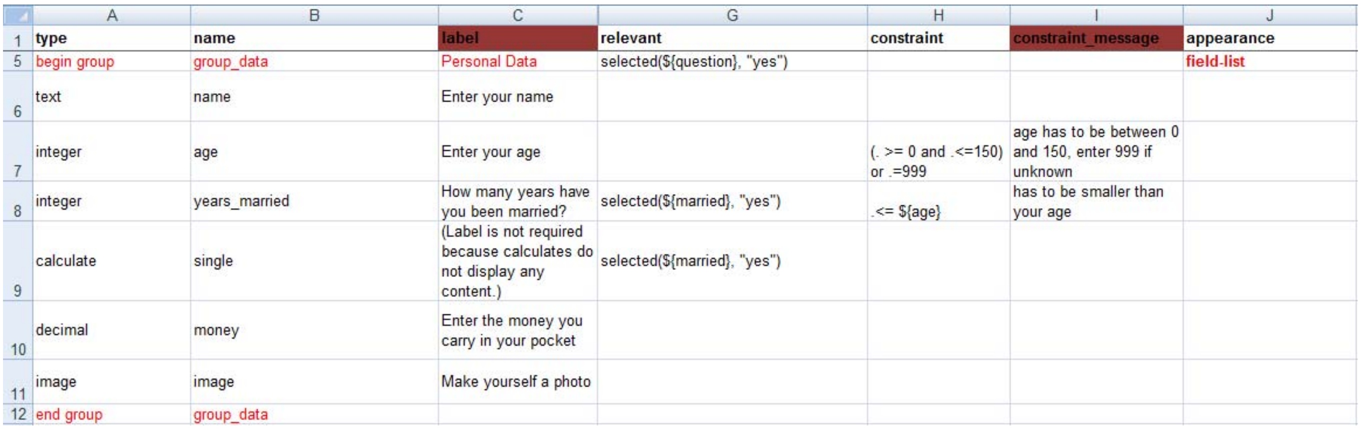
Are you married?

● Yes

○ No

Using “**field-list**” appearance you would get them to show up on the same screen:

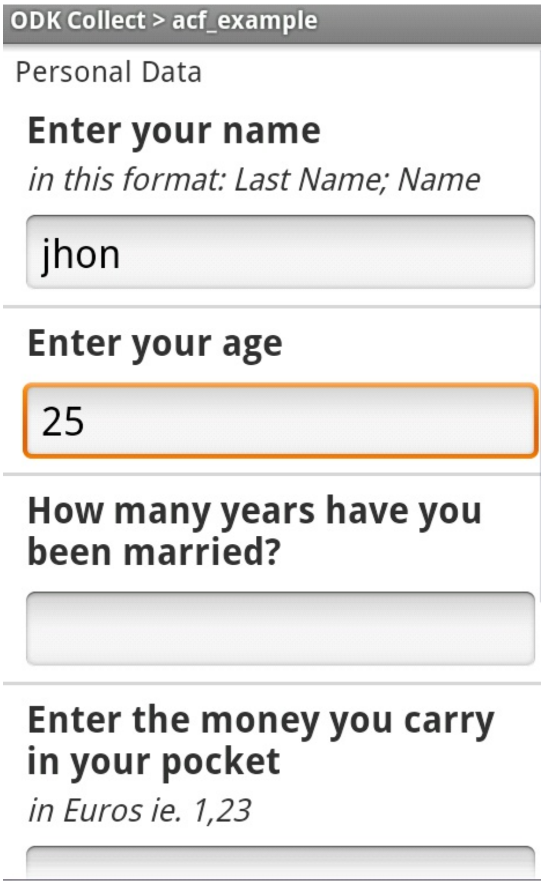
Excel 'survey' worksheet



	A	B	C	G	H	I	J
1	type	name	label	relevant	constraint	constraint_message	appearance
5	begin group	group_data	Personal Data	selected(${question}, "yes")			field-list
6	text	name	Enter your name				
7	integer	age	Enter your age		(. >= 0 and .<=150) or .=999	age has to be between 0 and 150, enter 999 if unknown	
8	integer	years_married	How many years have you been married?	selected(${married}, "yes")	.<= ${age}	has to be smaller than your age	
9	calculate	single	(Label is not required because calculates do not display any content.)	selected(${married}, "yes")			
10	decimal	money	Enter the money you carry in your pocket				
11	image	image	Make yourself a photo				
12	end group	group_data					

Witch looks like this on the device’s screen:

Note that questions are shown one on top of the other separated by a gray line:



ODK Collect > acf_example

Personal Data

Enter your name

in this format: Last Name; Name

jhon

Enter your age

25

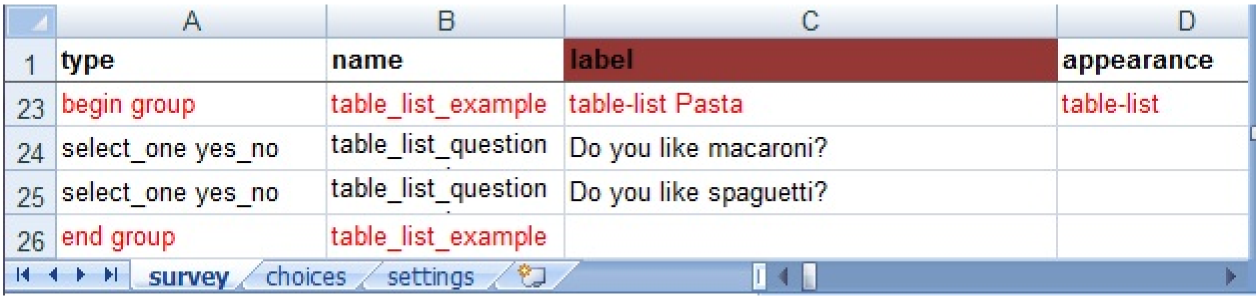
How many years have you been married?

Enter the money you carry in your pocket

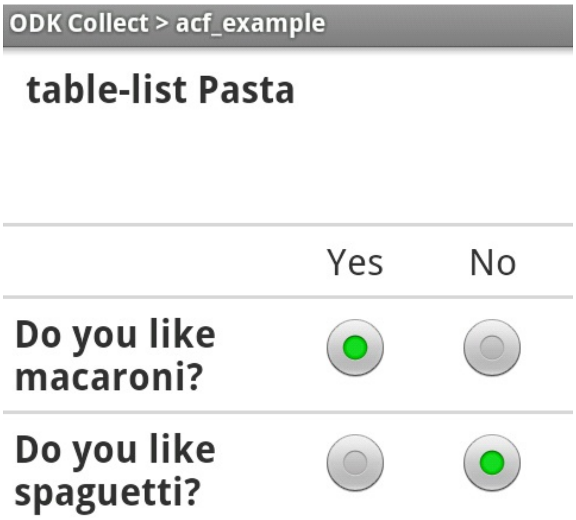
in Euros ie. 1,23

Now let’s use the **“table-list”** appearance widget to present the current choices as a horizontal, more compact, list.

Excel 'survey' worksheet



	A	B	C	D
1	type	name	label	appearance
23	begin group	table_list_example	table-list Pasta	table-list
24	select_one yes_no	table_list_question	Do you like macaroni?	
25	select_one yes_no	table_list_question	Do you like spaguetti?	
26	end group	table_list_example		



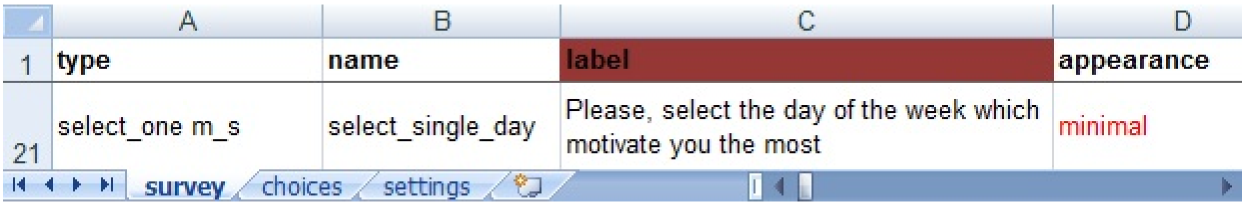
ODK Collect > acf_example

table-list Pasta

	Yes	No
Do you like macaroni?	●	○
Do you like spaguetti?	○	●

Formatting can also be done at the question level. For example, the applying the **"minimal"** "widget" to the "select_one m_s" currency question will make it shows up as a dropdown list:

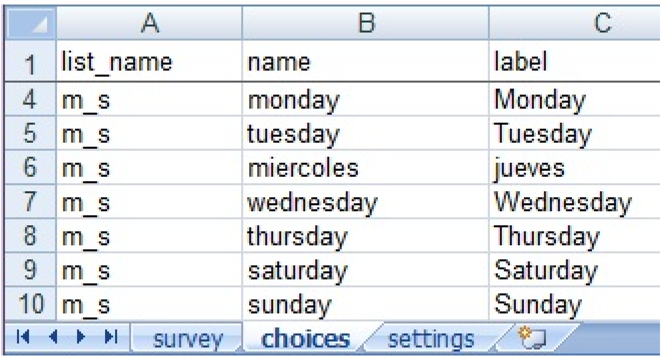
Excel 'survey' worksheet



	A	B	C	D
1	type	name	label	appearance
21	select_one m_s	select_single_day	Please, select the day of the week which motivate you the most	minimal

survey
choices
settings

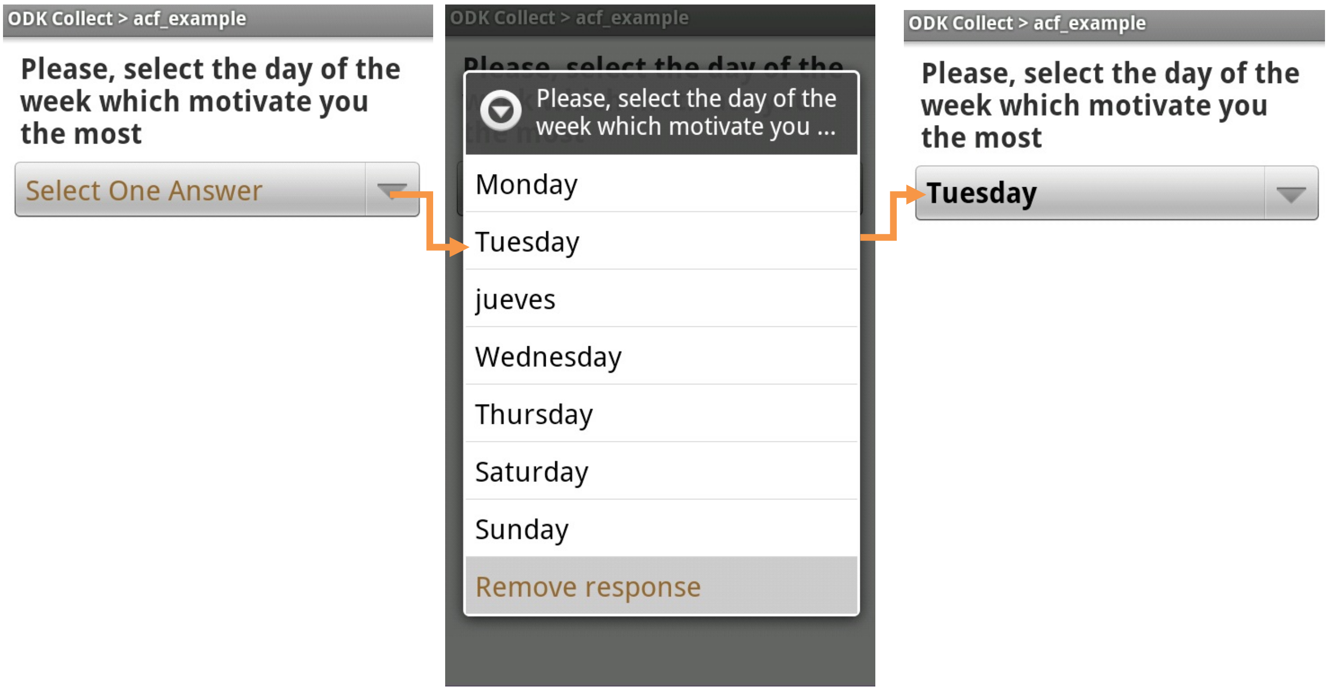
Excel 'choices' worksheet



	A	B	C
1	list_name	name	label
4	m_s	monday	Monday
5	m_s	tuesday	Tuesday
6	m_s	miercoles	jueves
7	m_s	wednesday	Wednesday
8	m_s	thursday	Thursday
9	m_s	saturday	Saturday
10	m_s	sunday	Sunday

survey
choices
settings

Appearance on the device's screen



ODK Collect > acf_example

Please, select the day of the week which motivate you the most

Select One Answer

ODK Collect > acf_example

Please, select the day of the week which motivate you ...

Monday

Tuesday

jueves

Wednesday

Thursday

Saturday

Sunday

Remove response

ODK Collect > acf_example

Please, select the day of the week which motivate you the most

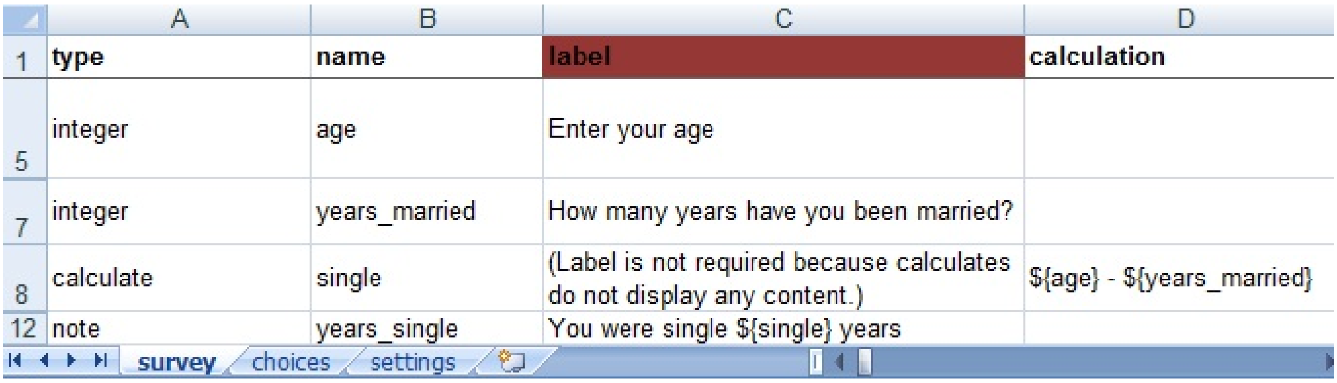
Tuesday

Making Dynamic Calculations

Your survey can perform dynamic calculations using the values of preceding questions. ODK Collect allows the use of a ‘calculate’ type, similar to any other question type but acts as a hidden field that will not show on the Android device´s screen. The calculation can be referenced like any other question using the ${name_of_calculated_field} expression.

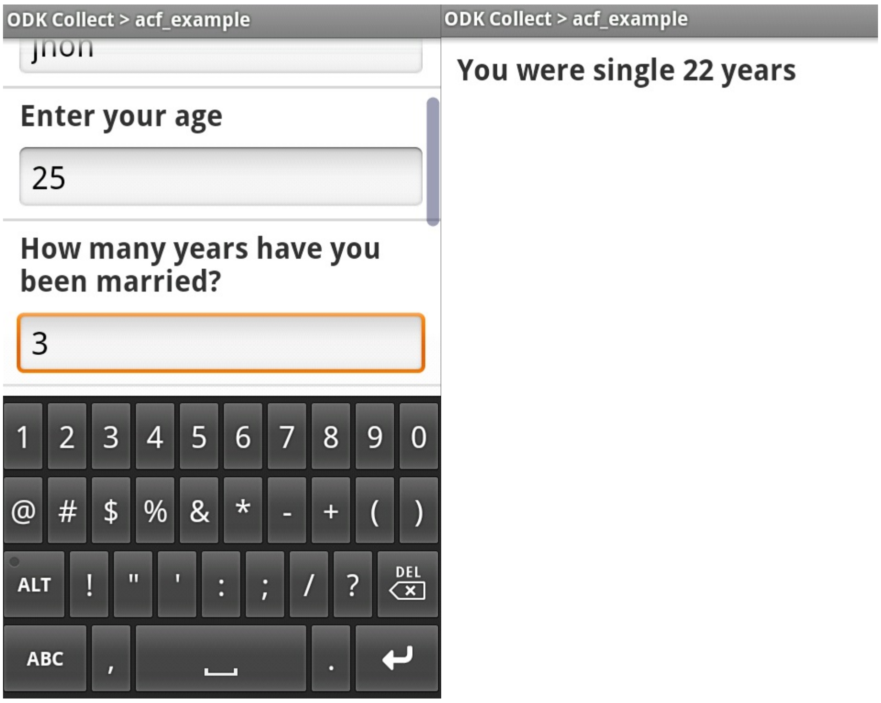
For example, in our survey, we want to calculate the years the respondent have been single and show it to the surveyor. We will require to use the ’calculation’ column. Once a value has been calculated it is not presented to the enumerator directly but can be presented in a note, as part of the label for next questions or as a conditional construct (constraint or relevant).

Excel 'survey' worksheet



	A	B	C	D
1	type	name	label	calculation
5	integer	age	Enter your age	
7	integer	years_married	How many years have you been married?	
8	calculate	single	(Label is not required because calculates do not display any content.)	${age} - ${years_married}
12	note	years_single	You were single ${single} years	

It will look like this on the device’s screen:



ODK Collect > acf_example

jhon

Enter your age

25

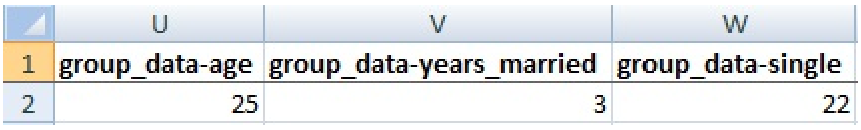
How many years have you been married?

3

ODK Collect > acf_example

You were single 22 years

The survey goes directly from group “personal data” to the note. The calculation was performed in the background and saved in the variable ‘group_data-single’. Notice how the note is presenting dynamic content. Notice that the enumerator never asked or entered any data about the years the respondent was single, the calculate field determined the respondent have been single for 22 years.



	U	V	W
1	group_data-age	group_data-years_married	group_data-single
2	25	3	22

The survey result data does not make any distinction between the data entered manually and a calculated variable.

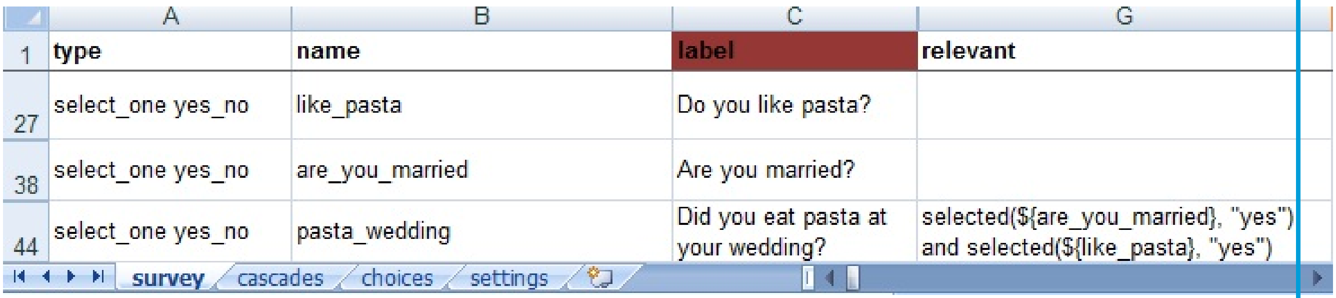
+ - * correspond to add, subtract, and multiply. Division however, is special, and you need to use the word “div” to do division

.

Multiple conditions

Binding conditions and formulas can be as complex as you want. You can combine and, or and not logical operators with arithmetic operators. This example is fairly simple, to appear the “Did you eat pasta at your wedding?” question, it has to been previously answered yes to “Do you like pasta?” and “are you married?”

Excel 'choices' worksheet



	A	B	C	G
1	type	name	label	relevant
27	select_one yes_no	like_pasta	Do you like pasta?	
38	select_one yes_no	are_you_married	Are you married?	
44	select_one yes_no	pasta_wedding	Did you eat pasta at your wedding?	selected(${are_you_married}, "yes") and selected(${like_pasta}, "yes")

Notice how the constraint condition becomes quite long. This constraint is telling ODK Collect to allow the surveyor to move forward only if the combination of selected items is “yes”

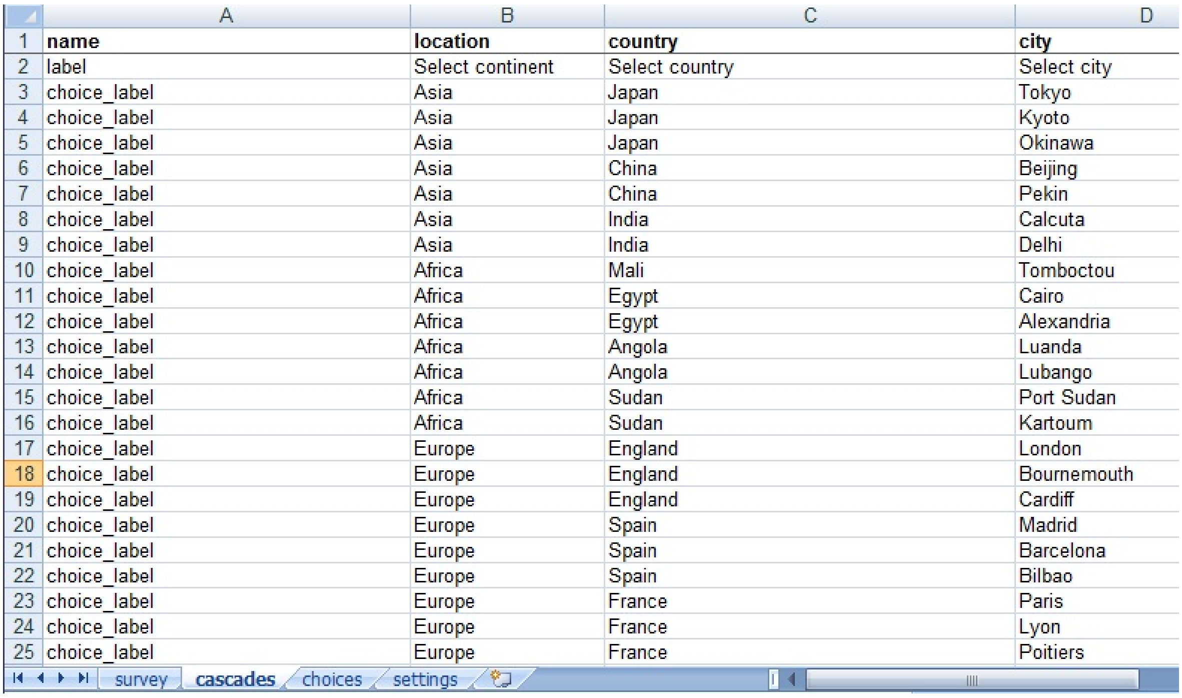
Cascading selects

Now we want to present the interviewee with a series of selection questions, where the answer depends on a previous selection. The classic example is asking someone where he lives. But you don’t want the user to deal with too much options at once, so you present enumerators with a question asking them about the continent, the country depending on the continent, and then, finally, the city.

We now present a syntax addition to overcome this issue and collect cascading-select information.

Since xlsforms are computer processed, we introduced a feature called cascading-select, where the user is able to simply list the cascading relationships. In a new worksheet called ‘cascades’ you create a new column ‘name’ and then an additional column for each level of the cascading hierarchy. In our case we have three levels so we would need a total of four columns ‘name’ ‘location’ ‘country’ and ‘city’. The first non-header row is still a special row and has to start with the word ‘label’ under column ‘name’ then under each additional column you can enter some free-text with the text you want to display to the enumerator for each hierarchy level. The next rows will contain the relationships between the different elements in your hierarchy. Japan for example has three cities (Tokyo, Kyoto and Okinawa) and belongs to Asia. Notice how every element contains ‘choice_label’ under the ‘name’ column. The lower level in the hierarchy (“city” in this example) stipulates the size of the sheet.

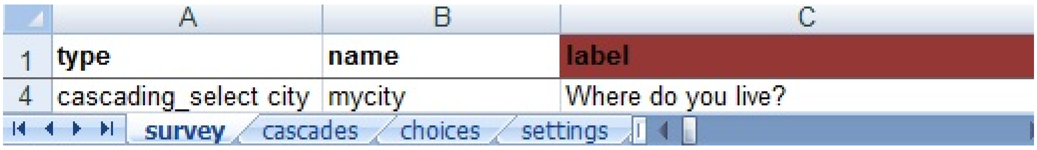
Excel 'cascades' worksheet



	A	B	C	D
1	**name**	**location**	**country**	**city**
2	label	Select continent	Select country	Select city
3	choice_label	Asia	Japan	Tokyo
4	choice_label	Asia	Japan	Kyoto
5	choice_label	Asia	Japan	Okinawa
6	choice_label	Asia	China	Beijing
7	choice_label	Asia	China	Pekin
8	choice_label	Asia	India	Calcuta
9	choice_label	Asia	India	Delhi
10	choice_label	Africa	Mali	Tomboctou
11	choice_label	Africa	Egypt	Cairo
12	choice_label	Africa	Egypt	Alexandria
13	choice_label	Africa	Angola	Luanda
14	choice_label	Africa	Angola	Lubango
15	choice_label	Africa	Sudan	Port Sudan
16	choice_label	Africa	Sudan	Kartoum
17	choice_label	Europe	England	London
18	choice_label	Europe	England	Bournemouth
19	choice_label	Europe	England	Cardiff
20	choice_label	Europe	Spain	Madrid
21	choice_label	Europe	Spain	Barcelona
22	choice_label	Europe	Spain	Bilbao
23	choice_label	Europe	France	Paris
24	choice_label	Europe	France	Lyon
25	choice_label	Europe	France	Poitiers

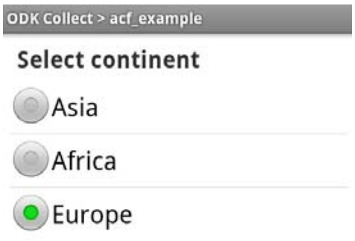
In the survey sheet we will add a new row of type 'cascading_select' followed by the name of the column of the lowest level (rightmost column in the Excel 'cascades' worksheet) of the hierarchy (in this case 'city'). The 'name' of this row element is user defined, in this case 'mycity'. For the content of the 'label' column in the Excel 'survey' sheet it is important to notice that the 'cascading_select' works differently than any of the question types we have been working with in the sense that a 'cascading_select' does not correspond to just one question but to as many questions as there are hierarchy levels defined in the Excel 'cascades' worksheet. In this case we have three levels so we will have three questions and thus three labels to display on the Android Smartphone screen. This might sound confusing but it's just how the cascading select works. The 'label' column on the Excel 'survey' sheet is overridden by the 'label' row on the Excel 'cascades' sheet. If you are confused just try it out and test it. You can then replace this example with your own cascading select.

Excel 'survey' worksheet



	A	B	C
1	**type**	**name**	**label**
4	cascading_select city	mycity	Where do you live?

This will present the user with three questions (which one shows up totally depends on previous input).

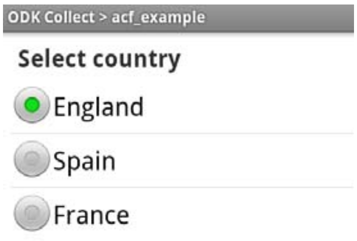


Select continent

○ Asia

○ Africa

● Europe

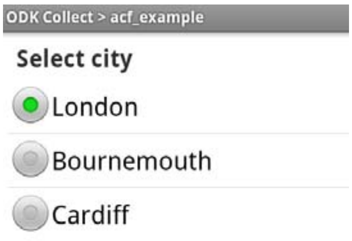


Select country

● England

○ Spain

○ France



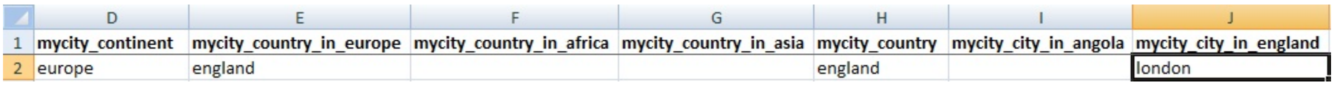
Select city

● London

○ Bournemouth

○ Cardiff

If you want to know the continent, the country, or the city of your submission, you can just look for variables called mycity_continent, mycity_country_in_continent, mycity_country and mycity_city_in_country in the exported data files. The “mycountry” bit was obtained from the name assigned to the cascading select in the survey sheet, and the “continent”, “country” and the “city” respectively from the ‘name’ row in the Excel ‘cascades’ sheet.



	D	E	F	G	H	I	J
1	mycity_continent	mycity_country_in_europe	mycity_country_in_africa	mycity_country_in_asia	mycity_country	mycity_city_in_angola	mycity_city_in_england
2	europe	england			england		london

The above is a simple test case, and may not seem impressive. But when you’re dealing with lots of variables the feature can be very effective.

Media

You can also make questions that show the user an ’image’, a ’video or an ’audio’ file as a way of asking a question. In order to do this, you will need to put the media in /odk/formname-media folder on your phone, and reference them from xlsform in a media column.

Logo

There is an easy way to show a logo at the top/beginning of your form. Simply upload an image file in the .png format called **form_logo.png**. You can do this in the Media section of the form settings page.

Appendix I - All Question Types in “type” column, “survey” sheet

Description	Phone Properties
record start time of survey	***start***
record last time survey was opened for editing	***end***
record date in the phone	***today***
record an Unique device id number	***deviceid***
record imei number (not available for all devices)	***imei***
record phone number (not available for all devices)	***phonenumber***
Description	**Question Types**
Insert an Input text box	***text***
Insert an Input integer (number) box	***integer***
Insert an Input integer (decimal number) box	***decimal***
Insert a dialog box with options of select one of the choices given	***select_one [options] [or_other]***
Insert a dialog box with options of multiple select of the choices given	***select_multiple [options] [or_other]***
Insert a check box	***acknowledge***
Insert a date (dd/mm/yy) widget	***date***
Insert a date (dd/mm/yy) & time (hh:mm) widget	***datetime***
Insert a time (hh:mm) widget	***time***
Insert a GPS launcher	***geopoint***
Insert a Barcode launcher	***barcode***
Insert a Camera launcher	***image***
Insert a Audio recorder launcher	***audio***
Insert a Video recorder launcher	***video***
Insert a Video recorder launcher	***note***
Creates an instance variable with no body element that gets set by an xpath expression (specified in the calculation column)	***calculate***
Description	**Groups**
Create a group set of questions	***begin_group***
Defines where the set of questions ends	***end_group***
Create a repeating set of questions	***begin_repeat***
Defines where the repeating set of questions ends	***end_repeat***

Appendix II - All columns headings, "survey" sheet

Description	*Column Headers*
Establish the type of question and the form structure	***type***
Assigns a unique variable name that will serve as a reference to the survey question	***name***
Establish what text will be displayed to the user when opening the form with ODK Collect	***label***
Establish what hint or secondary text will be displayed to the user when opening the form with ODK Collect	***hint***
Provides limited range of possible answers that can be considered as valid	***constraint***
Specify the message displayed on the screen when the constrain is not true	***constraint_message***
Designate certain questions as "required" questions	***required (yes, no, TRUE, FALSE)***
Designate defaults answers in case the user does not enter anything	***default***
Only present a question to the enumerator if a condition, based on the response to a previous question, is true.	***relevant***
Designate certain questions as "read-only" questions	***read_only (yes, no, TRUE, FALSE)***
Perform dynamic calculations using the values of preceding questions	***calculation***
This three columns can be translated to any language we want , just setting column::Language (i.e. label::English, label::Swahili)	***hint/label/constrain_message::language***
Establish what image will be displayed to the user when arrives to the question with the image(it requires that in the /odk /media folder in the device exist a copy of the image to be displayed	***media::image***
Establish what audio will be played to the user when arrives to the question with the sound(it requires that in the /odk /media folder in the device exist a copy of the audio track to be played	***media::audio***
Establish what video will be displayed to the user when arrives to the question with the video(it requires that in the /odk /media folder in the device exist a copy of the video to be displayed	***media::video***
This column can be configured to show a different image depending on the language we are using	***media::image::language***
This column can be configured to play a different audio file depending on the language we are using	***media::audio::language***
This column can be configured to show a different image depending on the language we are using	***media::video::language***
Allows for certain changes in the formatting of Smartphone screens	***appearance***

Appendix III - All columns headings, “choices” sheet

Description	Choices Page Column headers
Refers to the select one , select multiple question	***list_name***
Assigns a unique variable name that will serve as a reference to the choice	***name***
Establish what text will be displayed to the user when opening a select question with ODK Collect	***label***
Establish what media will be displayed to the user when opening a select question with ODK Collect	***media***

Appendix IV - All columns headings, “settings” sheet

Description	Form Properties
Defines how the form name will appear on ODK (if missing assigned to form_id)	***form_title***
Defines how the form name will appear on ODK Aggregate(If missing assigned to xls name)	***form_id***
For encryption purposes	***public_key***
Predefined Url of submission	***submission_url***
Set the language the form will appear by default	***default_language***

Appendix V - All variable references, “survey” sheet

Description	Form Variable References
- Prior question	***${variable_name}***
- Current question	***.***
- In name column	***${variable_name}***

Appendix VI - Appearance, “survey” sheet

Appearance (For more controlled views using widgets)	
- Multiple questions on a screen	***field-list***
- Select One Widgets	
-- Spinner widget - click button to provide a response	***minimal***
-- Table of answers	***label***
-- Goes with label, no label on answer just radios	***list-nolabel***
--show multiple select1s with a shared label in a field list.	***table-list***