

# Question Types

ODK Collect supports forms with a wide variety of question types. The exact functionality and display style of each question are specified in your [XLSForm](#) definition using the `type` and `appearance` columns.

## Tip

You can find an XLSForm with all available question types [here](#).

For simpler forms you can use [ODK Build](#), which provides a visual, drag-and-drop interface.

## Helpful terminology

### question

A prompt to the user, usually requesting a response. Questions are written as a single line in an XLSForm, and usually appear on a single screen in Collect.

### widget

A rendered question screen in Collect. The `type` and `appearance` of a question determine the widget that is displayed.

## Text widgets

All of the text widgets share the `text` type, and the inputs from them are saved as literal strings.

## Warning

If you are using Aggregate and expect answers to be more than 255 characters, you should [increase the database field length to over 255 characters](#).

## Default text widget

### type

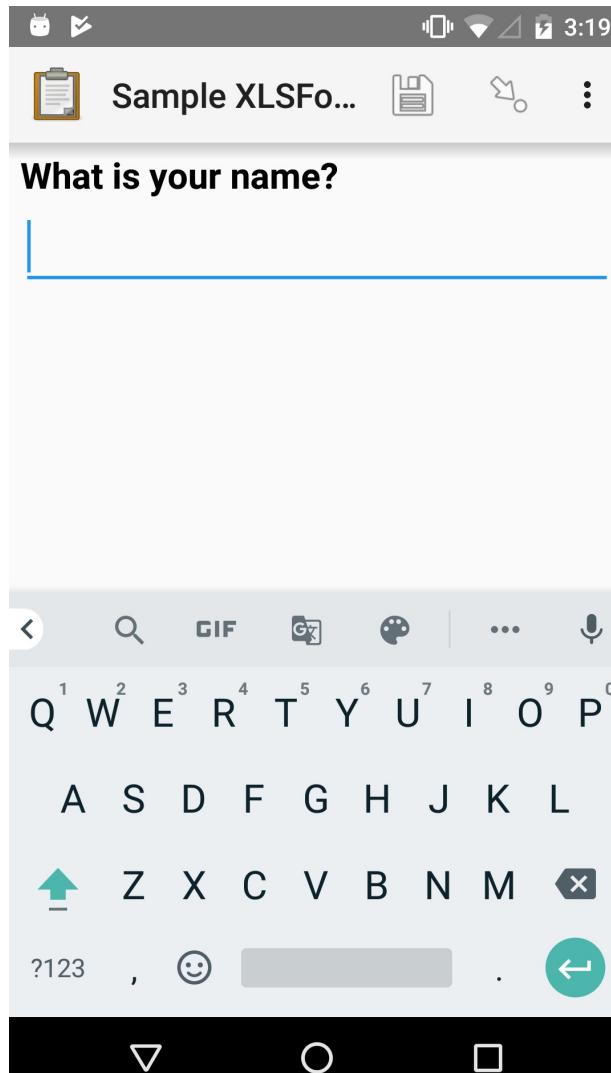
`text`

### appearance

`none`

A simple text input.

The text entry field expands as the user types, and line breaks can be included. The keyboard displayed depends on the Android device and user settings.



## XLSForm

type	name	label	appearance
text	name	What is your name?	survey

## Number text widget

type

text

appearance

numbers

A numerical input that treats the input as a string, rather than a number.

The number input accepts numerals ( 0123456789 ), hyphens ( - ), and decimal points ( . ). These are the only characters available on the number keypad displayed with this widget.

This is useful for phone numbers, ID numbers, IP addresses, and similar data. It can also be used in place of the [Integer widget](#) or [Decimal widget](#) if large numbers are needed. (The integer widget has a limit of nine digits, and the decimal widget has a limit of 15 characters.)



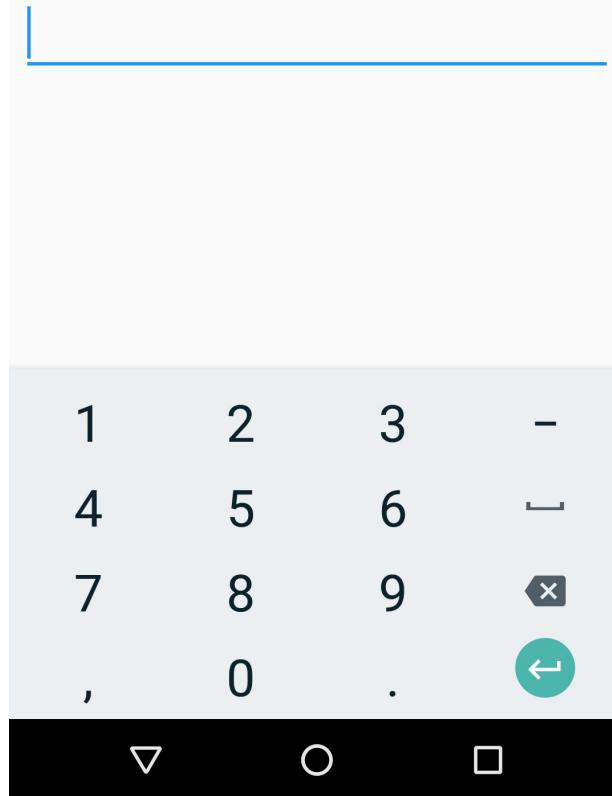
All widgets



## Text widgets

**String number widget**

text type with numbers appearance

**XLSForm**

survey

type	name	label	appearance	hint
text	string_number_widget	String number widget	numbers	text type with numbers appearance

## Note

This appearance can be combined with the [thousands-sep](#) appearance.**External app string widget**

type

`text`

appearance

`ex.*`

Launches an external app and receives a string input back from the external app. If the specified external app is not available, a manual input is prompted.

The external app widget is displayed when the `appearance` attribute begins with `ex:`. The rest of the `appearance` string specifies the application to launch.

## See also

[Launching External Apps](#)



All widgets



Text widgets

## Ex string widget

*text type with  
ex:change.uw.android.BREATHCOUNT  
appearance (can use other external apps)*

Launch

All widgets



Text widgets

## Ex string widget

*text type with  
ex:change.uw.android.BREATHCOUNT  
appearance (can use other external apps)*

Launch

The requested application is missing.  
Please manually enter the reading.

## XLSForm

survey					
type	name	label	appearance	hint	
text	ex_string_widget	Ex string widget	ex:change.uw.android.BREATHCOUNT	text type with ex:change.uw.android.BREATHCOUNT appearance (can use other external apps)	

# Number widgets

Number widgets collect and store number inputs — either [integers](#) or [floating-point decimals](#).

Number values can also be captured by the [Range widgets](#).

## Integer widget

type

`integer`

appearance

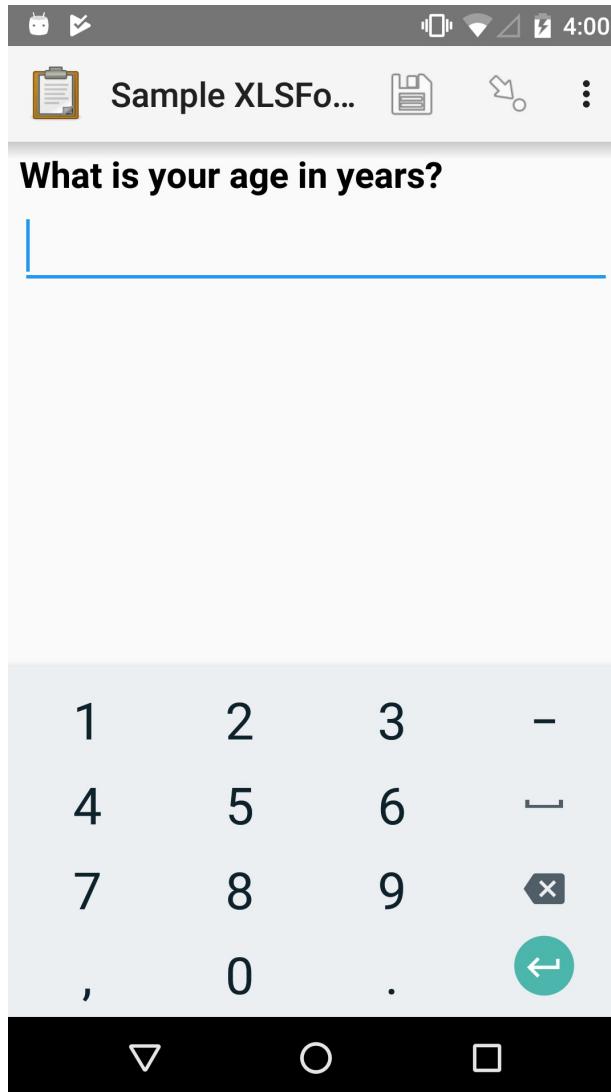
`none`

A whole number entry input.

Integer widgets will not accept decimal points, and the entry field has a limit of nine digits. If you need numbers larger than nine digits, see the [Number text widget](#).

The integer widget supports:

- [Thousands separators](#)
- [External apps](#)



## XLSForm

survey		
type	name	label
integer	age	What is your age in years?

## Decimal widget

type

`decimal`

appearance

`none`

A numeric input that will accept decimal points.

Decimal number entry is capped at 15 characters (14 digits and a decimal point). If you need numbers larger than 15 digits, see the [Number text widget](#).

The decimal widget supports:

- [Thousands separators](#)
- [External apps](#)



## Weight in kilograms.

1    2    3    -  
4    5    6    [ ]  
7    8    9      
,    0    .   

## XLSForm

survey		
type	name	label
decimal	weight	Weight in kilograms.

## Number widget appearance options

### Thousands separator

type

```
integer, decimal, ( text )
```

appearance

```
thousands-sep, ( numbers )
```

If `thousands-sep` is added to `appearance`, `integer`, `decimal`, and `number text` widgets will display their values using locale-specific thousands separators.

### Note

For locales that use the point separator (.), a space is used instead.



## All widgets



Numerical widgets

### Integer widget with thousands separators

*integer type with thousands-sep appearance. This appearance can also be applied to decimal and string numbers widgets*

1,000,000

1	2	3	-
4	5	6	—
7	8	9	✖
,	0	.	⬅
▽	○	□	

1	2	3	-
4	5	6	—
7	8	9	✖
,	0	.	⬅
▽	○	□	

### Number from an external app

type

integer, decimal

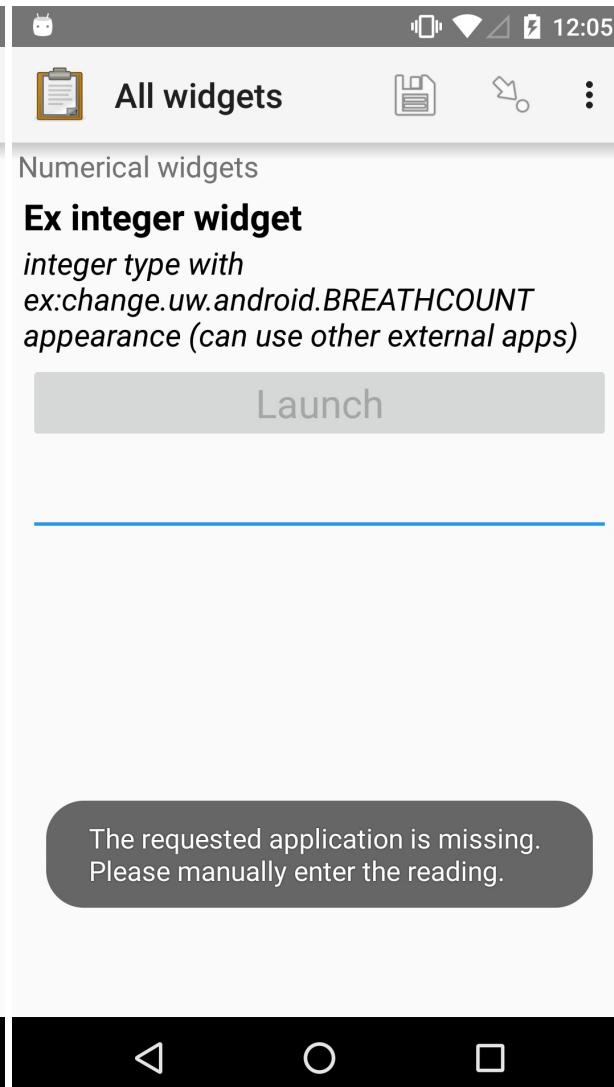
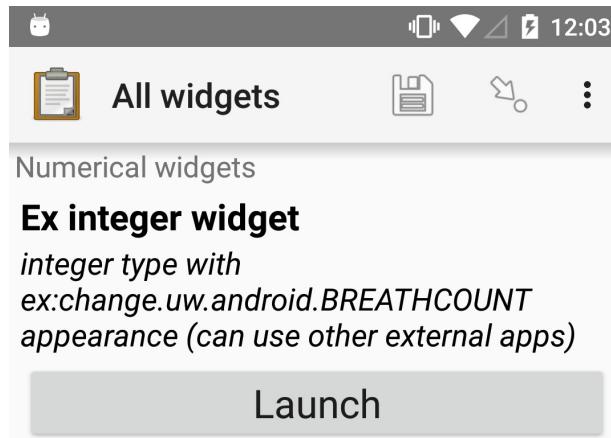
appearance

ex:\*

By specifying an external app in the `appearance`, your form can launch an external app and receive a number (integer or decimal) from the external app as input. If the specified external app is not available, a manual input is prompted.

See also

[Launching External Apps](#)



## XLSForm

survey					
type	name	label	appearance	hint	
integer	ex_integer_widget	Ex integer widget	ex:change.uw.android.BREATHCOUNT	integer type with ex:change.uw.android.BREATHCOUNT appearance (can use other external apps)	

## Date and time widgets

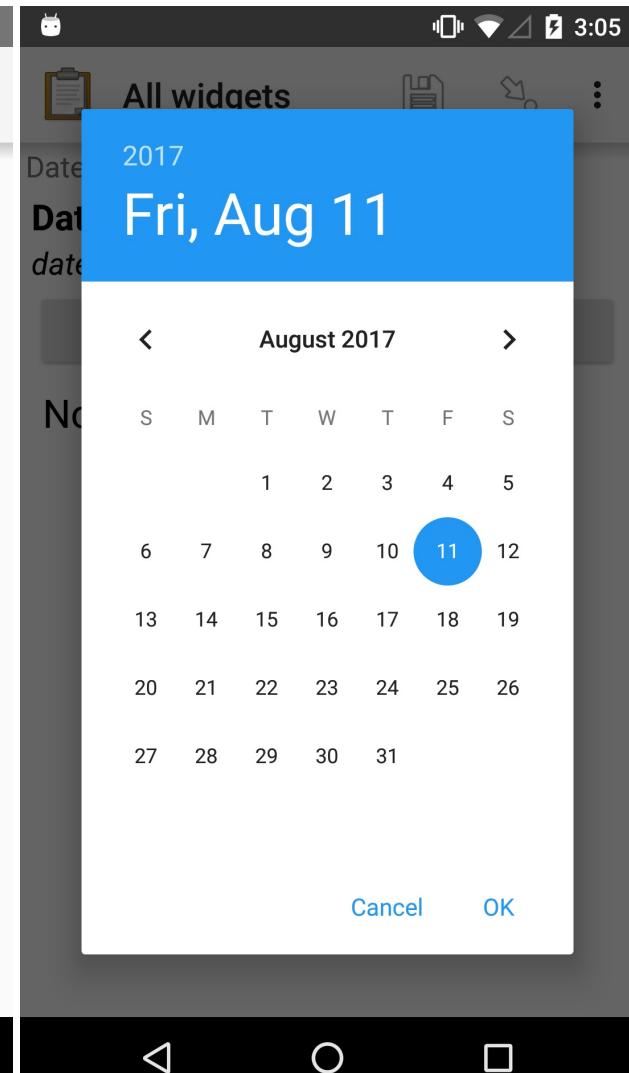
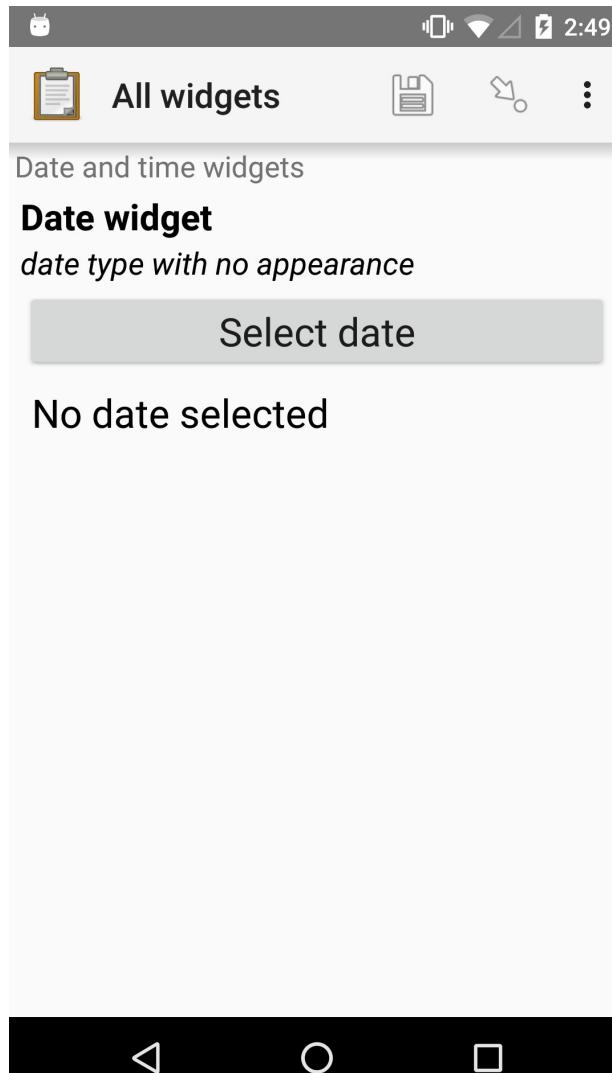
### Default date widget

type

date

appearance

none





All widgets



Date and time widgets

**Date widget***date type with no appearance*

Select date

Aug 11, 2017

**XLSForm**

survey				
type	name	label	hint	
date	date_widget	Date widget	date type with no appearance	

**Date widget with spinner input**

type

date

appearance

no-calendar

The `no-calendar` appearance displays a spinner-style date selection. This is especially appropriate for selecting dates more than one year in the past or future.



All widgets



Date and time widgets

## Date Widget

date type with no-calendar

Select date

No date selected

All widgets



Date and time widgets

## Date Widget

date

Select date

No

Jul

10

2016

Aug

11

2017

Sep

12

2018

Cancel

OK

<

O

□

<

O

□

## XLSForm

survey

type	name	label	appearance	hint
date	date_widget_nocalendar	Date Widget	no-calendar	date type with no-calendar appearance

## Month and year only

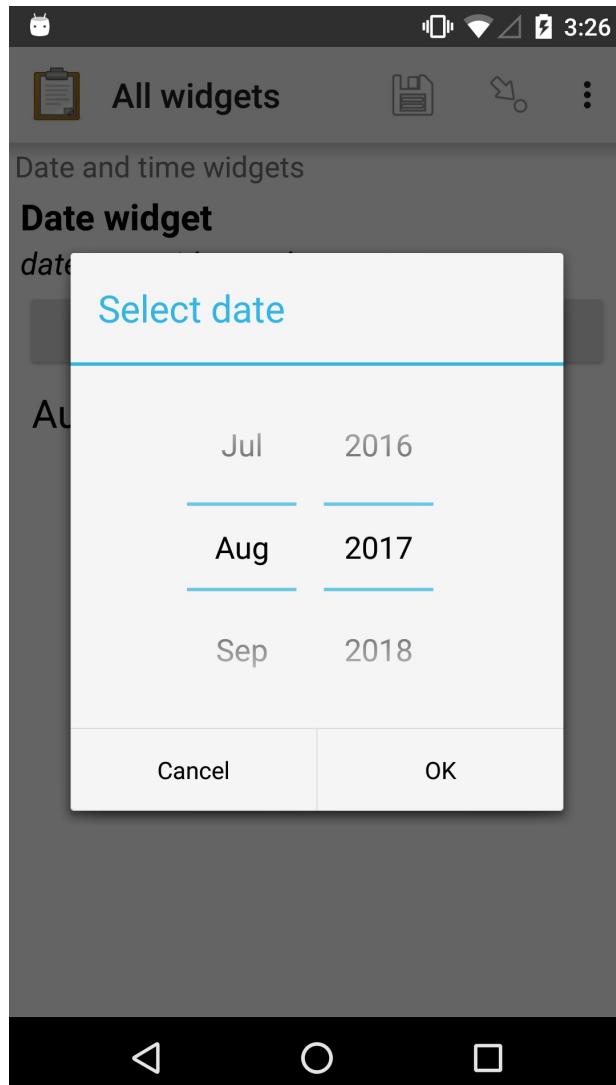
type

date

appearance

month-year

Collects only a month and year.



## XLSForm

survey				
type	name	label	appearance	hint
date	date_widget_month_year	Date widget	month-year	date type with month-year appearance

## Year only

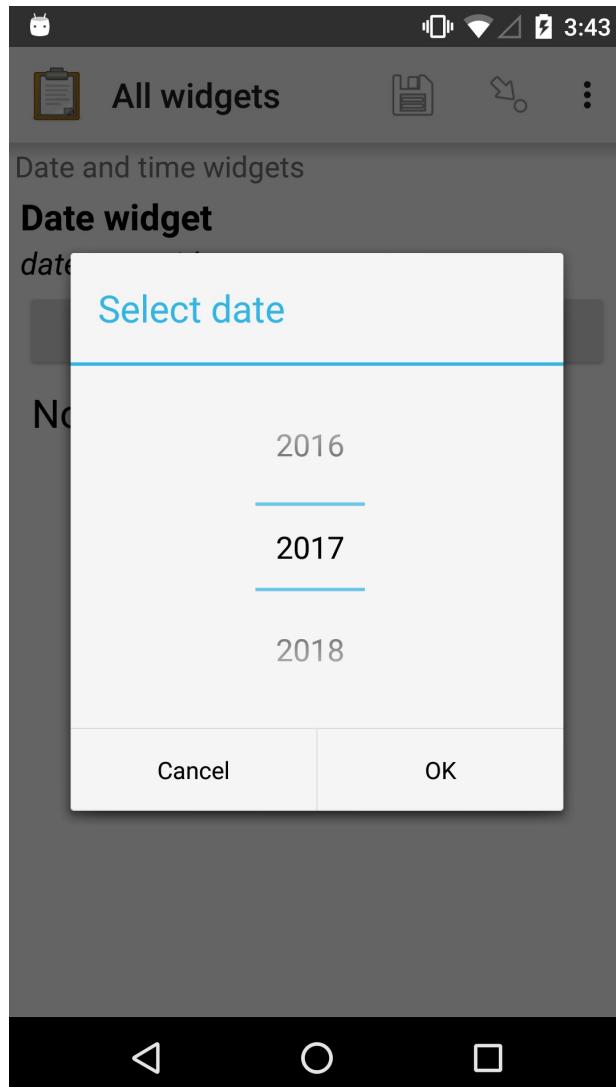
type

date

appearance

year

Collects only a year.



## XLSForm

survey					
type	name	label	appearance	hint	
date	date_widget_year	Date widget	year	date type with year appearance	

## Date widgets with non-Gregorian calendars

Collect supports several non-Gregorian calendars.

### Note

The non-Gregorian calendar is used only on input. The dates are converted and stored as standard Gregorian dates

### Coptic calendar

type

date

appearance

coptic



All widgets



Date and time widgets

## Coptic date widget

*date type coptic appearance*

### Select date

4 Meshir 1733

5 Paremhat 1734

6 Parmouti 1735

5 Paremhat 1734 (Mar 14, 2018)

[CANCEL](#) [OK](#)



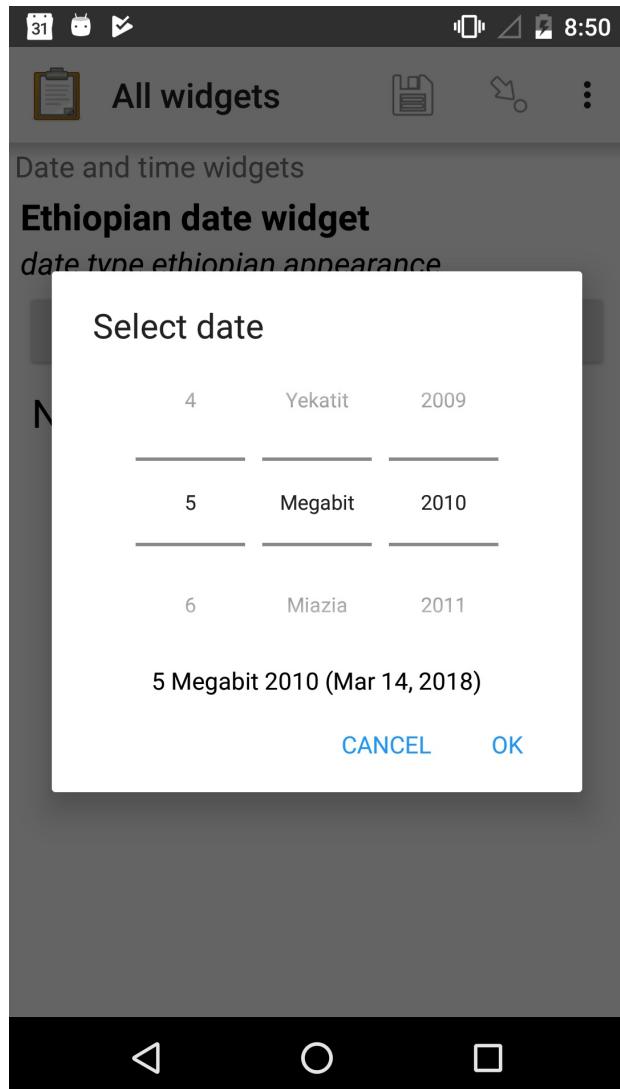
## Ethiopian calendar

type

[date](#)

appearance

[ethiopian](#)



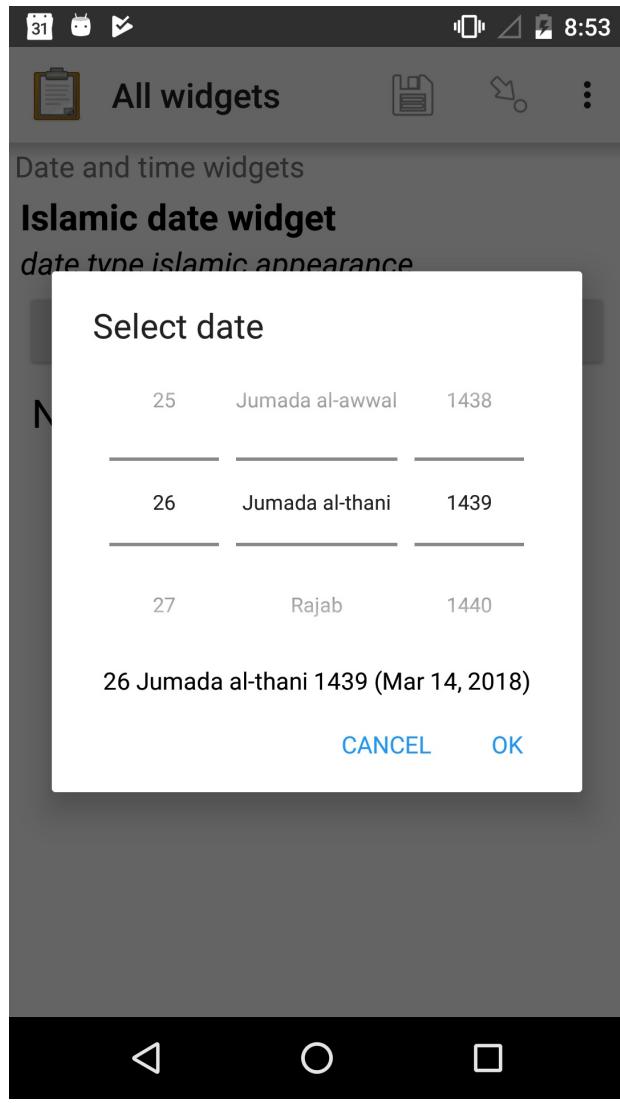
## Islamic calendar

type

`date`

appearance

`islamic`



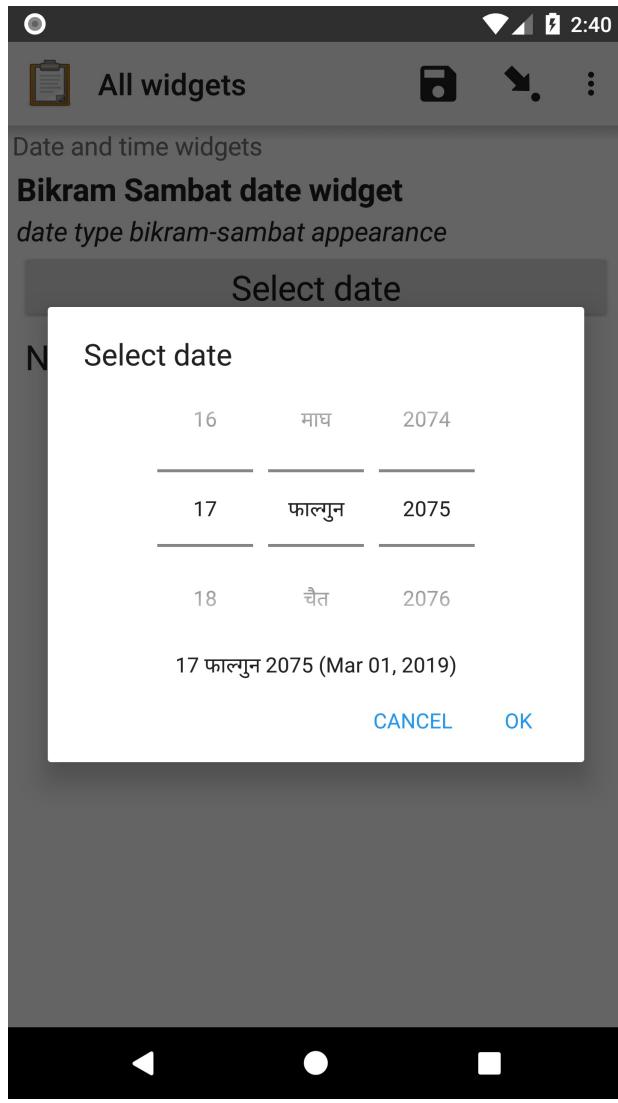
## Bikram Sambat calendar

type

`date`

appearance

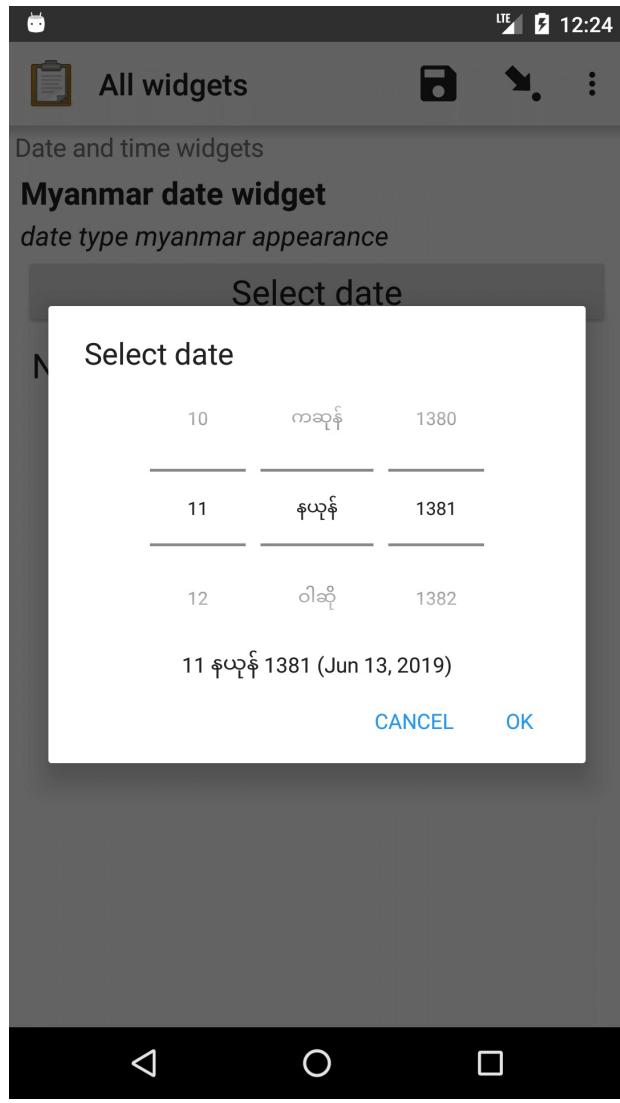
`bikram-sambat`



## Myanmar calendar

type

appearance



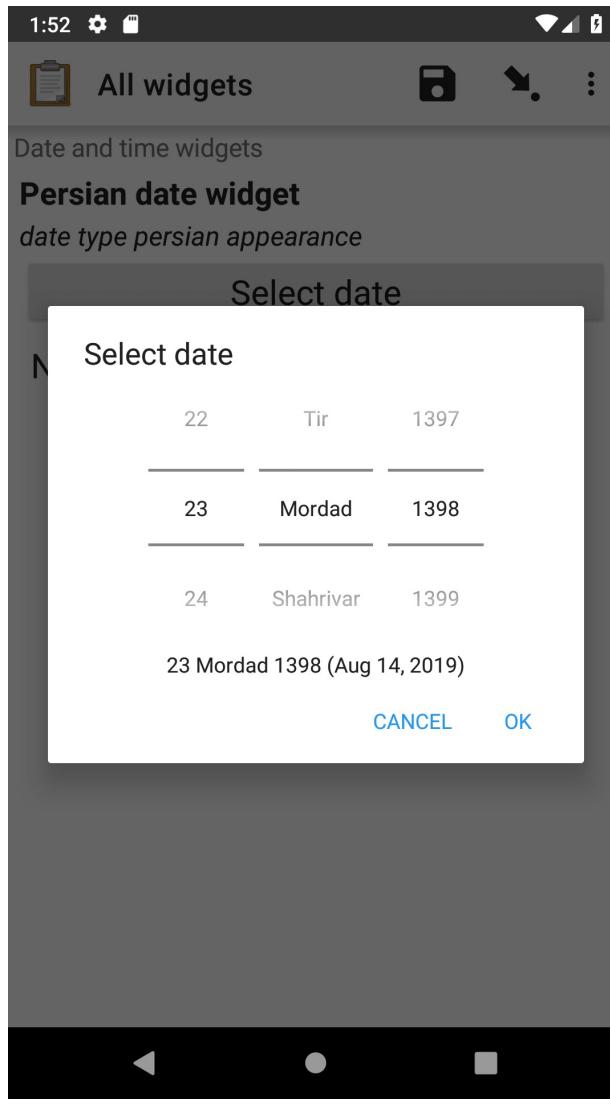
## Persian calendar

type

`date`

appearance

`persian`



## Time widget

type

`time`

appearance

`none`

A time selector. Captures only a specific time-of-day, not a date and time. For date and time, see the [Datetime widget](#).

The time widget does not accept any `appearance` attributes.

### Note

The time widget stores the time along with a time zone. This can cause unexpected behavior around [Daylight saving time](#).

For example, if you record a time before the clock change, and then view the time after the clock change, it will appear to be an hour off. This happens because the recorded time data is understood as a specific moment in time that is being "translated" into your current, local time zone.

A similar problem occurs when moving between geographic time zones.

This makes the time widget unsuitable for abstract time-of-day questions such as *What time do you usually wake up?* For questions like this, you may want to use a [Minimal select widget](#). You can set the options at whatever level of accuracy you need — for example, 15 or 30 minute increments. Alternatively, you could use the select widget for hours, and an [Integer widget](#) for minutes.



Sample XLSFo...

**What time do you usually wakeup?****Select time**

No time selected



Sample XLSFo...

**What time do you usually wakeup?****Select time**

No

**Select time**

7                    37

8 : 38 AM

9                    39 PM

Cancel

OK





What time do you usually wakeup?

Select time

06:30



## XLSForm

survey		
type	name	label
time	wakeup	What time do you usually wakeup?

## Datetime widget

A date and time selector.

For date only, see [Default date widget](#). For time only, see [Time widget](#).



Sample XLSFo...

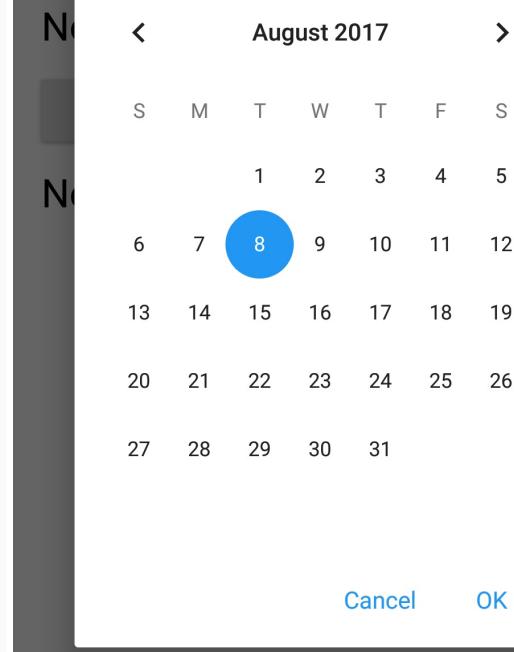
When was the last time you ate?

Select date

No date selected

Select time

No time selected





Sample XLSFo...



When was the last time you ate?

Select date

Aug 08, 2017

Select time

No time selected



Sample XLSFo...



When was the last time you ate?

Select date

Aug

No

Select time

7 50

8 : 51

AM

9 52

PM

Cancel

OK





**When was the last time you ate?**

Select date

Aug 08, 2017

Select time

No time selected



## XLSForm

survey		
type	name	label
dateTime	previous_meal	When was the last time you ate?

### Note

The [Datetime widget](#) supports the [no-calendar](#) spinner-style appearance.

## Select widgets

Select widgets display choices to pick from. Single selects allow selecting a [single choice](#), and multi selects allow [selecting multiple choices](#).

The choices for a select question can be included on a sheet named **choices** directly in an XLSForm or attached as an [external dataset](#).

The order of the choices can be [randomized](#) for any of the select types described below. The list of choices available can also be [filtered](#) based on answers to previous questions. Selects from internal datasets can [include images as choices](#).

Selects can be displayed in different ways using [appearances](#).

The **choices** sheet for defining internal datasets has at least three columns:

### list\_name

A set of choices for a single question share a common `list_name`. The value of `list_name` is included in the `type` column on the **survey** sheet.

### name

The identifier for a specific choice. This value is what is stored on the completed form. If you [refer to a select response using a variable](#), the `name` string is returned.

As in the **survey** sheet, the `name` for a choice must not include spaces.

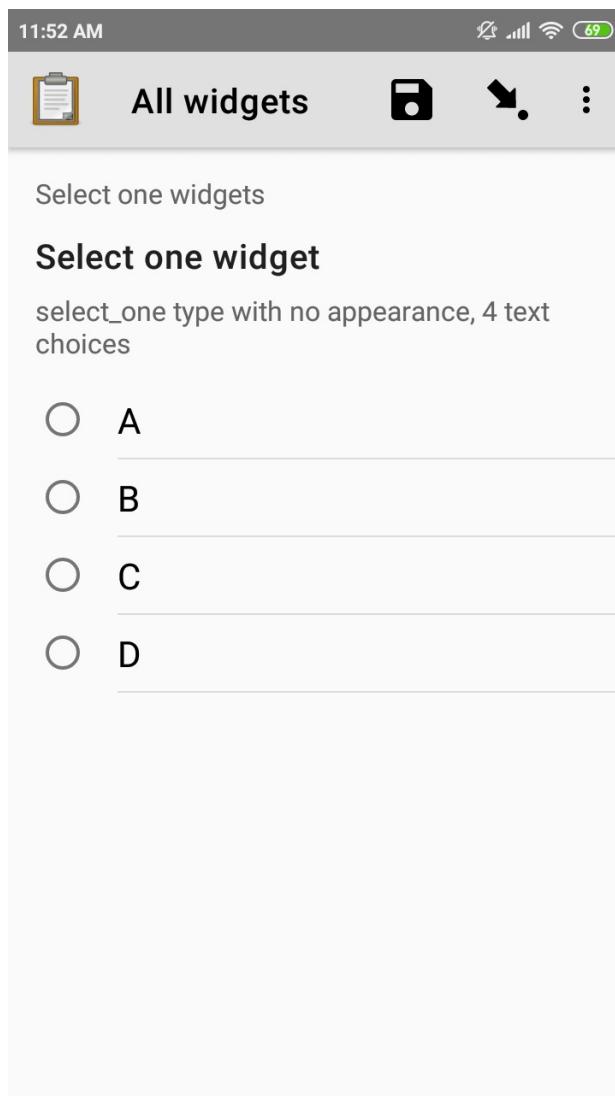
### label

The user-facing text displayed for the choice.

## Single select widget

### type

```
select_one {list_name}
```



## XLSForm

survey

<b>type</b>	<b>name</b>	<b>label</b>	<b>hint</b>
select_one opt_abcd	select_one_widget	Select one widget	select_one type with no appearance, 4 text choices

choices		
list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

## Multi select widget

type

```
select_multiple {list_name}
```

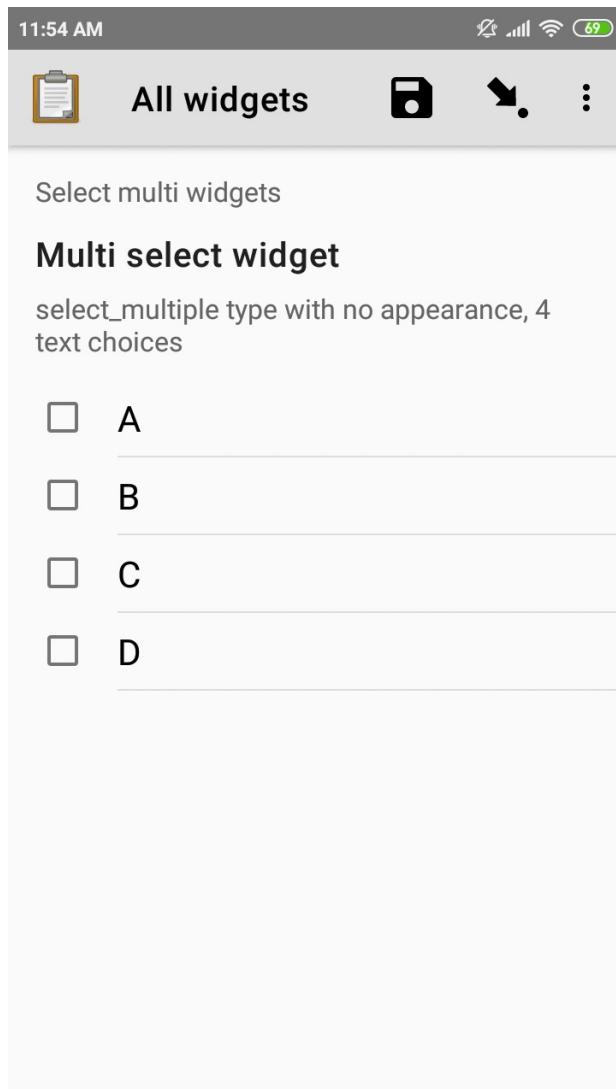
appearance

*none*

Multi select questions allow selecting multiple answers. The response for the question will be the space-separated choices made by the user, in the order that they were selected.

### Note

The multi select widget supports all of the same `appearance` attributes as the [Single select widget](#) excluding the `quick` appearance.



## XLSForm

survey			
type	name	label	hint
select_multiple opt_abcd	select_multi_widget	Multi select widget	select_multiple type with no appearance, 4 text choices

choices			
list_name	name	label	image
opt_abcd	a	A	
opt_abcd	b	B	
opt_abcd	c	C	
opt_abcd	d	D	

## Select from external dataset

Data files in CSV, GeoJSON or XML format can be attached to form definitions. These [external datasets](#) can be used as data sources for selects. The question type for single selection is `select_one_from_file` and for multiple selection, it is `select_multiple_from_file`. The full filename of the dataset including the extension goes after the type.

Selects from external datasets can be used in all the same ways as internal selects. For example, they can be displayed differently using [appearances](#) or filtered using [choice filters](#).

type

```
select_one_from_file {file.extension}
```

### XLSForm

survey		
type	name	label
select_one_from_file hospitals.csv	hospital	Select hospital

hospitals.csv	
name	label
hospital_a	Hospital A
hospital_b	Hospital B
hospital_c	Hospital C
hospital_d	Hospital D

## Customizing the label and value

When using an [external dataset](#) as a data source for a select, the underlying value for each choice comes from:

- CSV file: the `name` column
- GeoJSON file: the `id` top-level element if it exists or the `id` property as a fallback
- XML file: the `name` child element

The label for each choice comes from:

- CSV file: the `label` column
- GeoJSON file: the `title` property (follows [the GeoJSON simplestyle specification](#))
- XML file: the `label` child element

In some cases, it may not be convenient to rename your columns to match these defaults. If you have a dataset from another source and different column names, you can use the `parameters` column in your XLSForm to specify which columns to use.

For example, to use `feature_id` for the underlying value and `human_name` for the label:

### XLSForm

survey			
type	name	label	parameters
select_one_from_file hospitals.csv	hospital	Select hospital	value=feature_id,label=human_name

feature_id	human_name
hospital_a	Hospital A
hospital_b	Hospital B

## Select appearances

Selects can be styled in various ways using the `appearance` column in an XLSForm. Unless otherwise indicated, the appearances described below can combine with single or multiple selects with either internal or external data sources.

### Minimal select widget

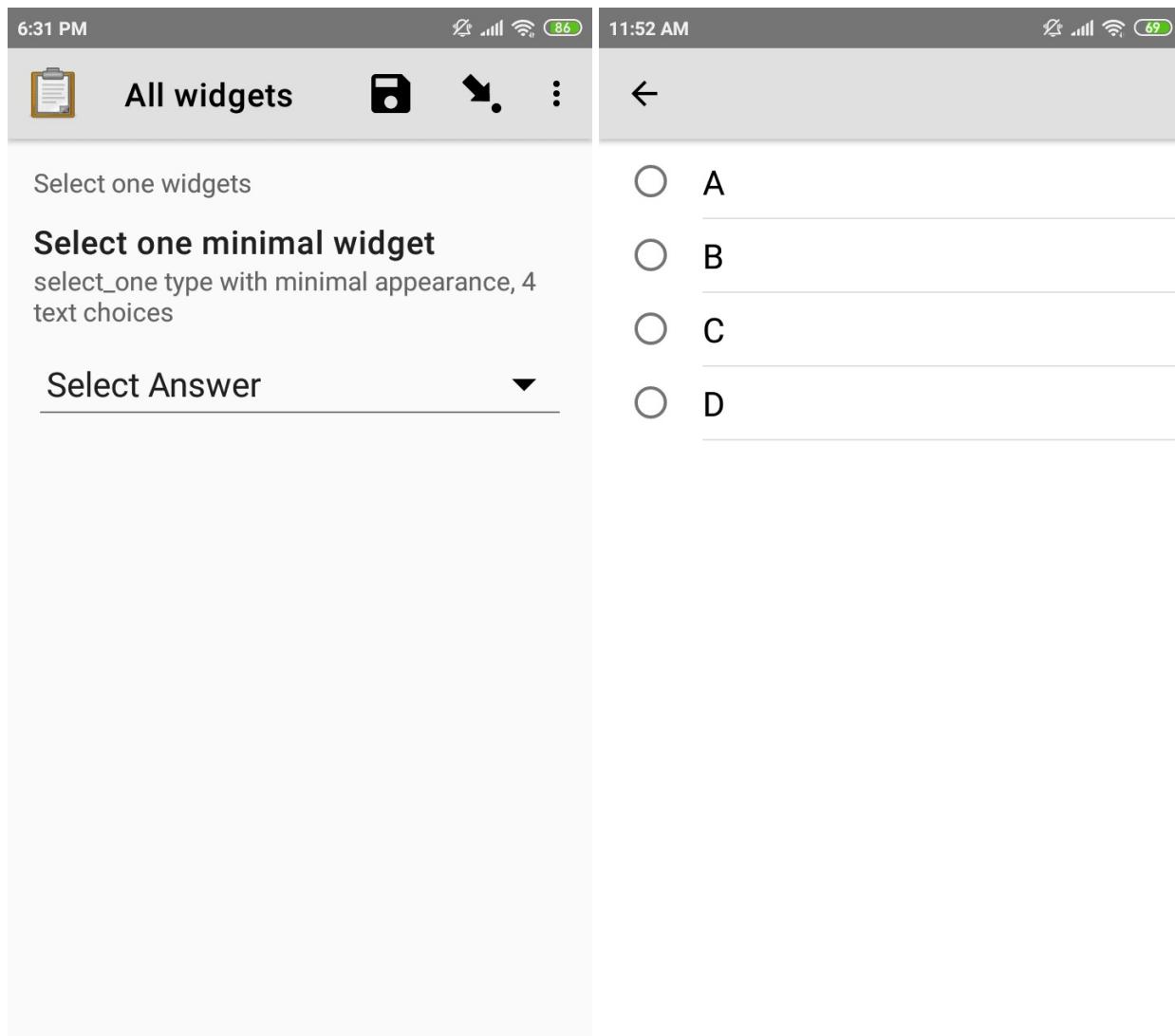
type

```
select_one {list_name}
```

appearance

```
minimal
```

Adding the `minimal` appearance shows the choices in a compact way. This is particularly helpful when the list of choices is long and the select question is displayed on [the same screen as other questions](#). It is often combined with [the autocomplete appearance](#).



## XLSForm

survey

type	name	label	appearance	hint
select_one opt_abcd	select_widget	Select widget	minimal	select_one type with minimal appearance, 4 text choices

choices		
list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

## Select widget with autoadvance

type

```
select_one {list_name}
```

appearance

```
quick
```

When the `quick` appearance is added, the form advances immediately to the next question once a selection is made.



The `quick` appearance can only be used with single selection.



## XLSForm

survey				
type	name	label	appearance	hint
select_one opt_abcd	select_one_autoadvance_widget	Select one autoadvance widget	quick	select_one type with quick appearance, 4 text choices

choices		
list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

## Select widget with autocomplete

type

```
select_one {list_name}
```

appearance

```
autocomplete
```

The `autocomplete` appearance allows the enumerator to filter the list of available choices. This is especially helpful for questions with a large number of choices.

11:52 AM



11:53 AM



All widgets

Select one widgets

### Select one search widget

select\_one type with autocomplete appearance, 4 text choices

b

A

B

C

q w e r t y u i o p  
a s d f g h j k l  
z x c v b n m   
?123 ,  .

All widgets

Select one widgets

### Select one search widget

select\_one type with autocomplete appearance, 4 text choices

b

B

q w e r t y u i o p  
a s d f g h j k l  
z x c v b n m   
?123 ,  .

## XLSForm

survey																			
type	name	label	appearance	hint															
select_one opt_abcd	select_one_autocomplete_widget	Select one widget	autocomplete	select_one type with autocomplete appearance, 4 text choices															
choices																			
<table border="1"> <thead> <tr> <th>list_name</th><th>name</th><th>label</th></tr> </thead> <tbody> <tr> <td>opt_abcd</td><td>a</td><td>A</td></tr> <tr> <td>opt_abcd</td><td>b</td><td>B</td></tr> <tr> <td>opt_abcd</td><td>c</td><td>C</td></tr> <tr> <td>opt_abcd</td><td>d</td><td>D</td></tr> </tbody> </table>					list_name	name	label	opt_abcd	a	A	opt_abcd	b	B	opt_abcd	c	C	opt_abcd	d	D
list_name	name	label																	
opt_abcd	a	A																	
opt_abcd	b	B																	
opt_abcd	c	C																	
opt_abcd	d	D																	

## Select widget with columns-pack appearance

type

```
select_one {list_name}
```

appearance

```
columns-pack
```

When the `columns-pack` appearance is added, the app tries to accommodate as many choices in a single line as possible. If the choice labels have different lengths, they will not be in even columns.



All widgets



Select one widgets

### Select one widget with packed columns

select\_one type with columns-pack appearance, 4 text choices. Choices are packed in to fit horizontal space with minimal padding.

- A
- B
- C
  
- D

### XLSForm

survey																			
type	name	label	appearance	hint															
select_one opt_abcd	select_widget	Select one widget	columns- pack	select_one type with columns-pack appearance, 4 text choices															
choices																			
<table border="1"> <thead> <tr> <th>list_name</th><th>name</th><th>label</th></tr> </thead> <tbody> <tr> <td>opt_abcd</td><td>a</td><td>A</td></tr> <tr> <td>opt_abcd</td><td>b</td><td>B</td></tr> <tr> <td>opt_abcd</td><td>c</td><td>C</td></tr> <tr> <td>opt_abcd</td><td>d</td><td>D</td></tr> </tbody> </table>					list_name	name	label	opt_abcd	a	A	opt_abcd	b	B	opt_abcd	c	C	opt_abcd	d	D
list_name	name	label																	
opt_abcd	a	A																	
opt_abcd	b	B																	
opt_abcd	c	C																	
opt_abcd	d	D																	

### Select widget with columns appearance

type

```
select_one {list_name}
```

appearance

```
columns
```

When the `columns` appearance is added, the app puts choices in 2, 3, 4 or 5 columns depending on the screen size.

Select widgets support image choices. The images are referenced in the **choices** sheet, and the image files need to be included in the `media` folder.

See [Including media files in choices](#) to learn more about including images in surveys.



All widgets



Select one widgets

### Select one widget with fixed columns

select\_one type with columns appearance, 4 text + image choices. Choices are layed out in fixed-width columns based on screen width.

A       B       C



D



### XLSForm

survey					
type	name	label	appearance		hint
select_one abcd_icon	select_widget	Select one widget	columns	select_one type with columns appearance, 4 text + image choices	
choices					
list_name	name	label	image		
abcd_icon	a	A	a.jpg		
abcd_icon	b	B	b.jpg		
abcd_icon	c	C	c.jpg		
abcd_icon	d	D	d.jpg		

### Select widget with columns-n appearance

type

```
select_one {list_name}
```

appearance

```
columns-n
```

When the `columns-n` appearance is added, the app puts choices in n columns.



All widgets



Select one widgets

### Select one widget with fixed column count

select\_one type with columns-2 appearance, 4 text + image choices. The available screen width is divided into 2 columns of the same width. Column counts between 1 and 10 are allowed.

 A B C D

### XLSForm

survey					
type	name	label	appearance		hint
select_one abcd_icon	select_widget	Select one widget	columns-2		select_one type with columns-2 appearance, 4 text + image choices
choices					
list_name	name	label	image		
abcd_icon	a	A	a.jpg		
abcd_icon	b	B	b.jpg		
abcd_icon	c	C	c.jpg		
abcd_icon	d	D	d.jpg		

### Select widget with no-buttons appearance

type

`select_one {list_name}`

appearance

`no-buttons`

When the `no-buttons` appearance is added, the app displays choices without the selection radio button. If images are specified for choices, only the images are displayed. This is particularly useful for building a grid of images.



All widgets



Select one widgets

### Select one widget with packed columns and no buttons

select\_one type with columns-pack no-buttons appearance, 4 image choices. Choices are packed in to fit horizontal space with minimal padding and no buttons are displayed.



## XLSForm

survey					
type	name	label	appearance		hint
select_one abcd_icon	select_widget	Select one widget	columns-pack no-buttons		select_one type with columns-pack no-buttons appearance, 4 image choices
choices					
list_name	name	label	image		
abcd_icon	a	A	a.jpg		
abcd_icon	b	B	b.jpg		
abcd_icon	c	C	c.jpg		
abcd_icon	d	D	d.jpg		

## Likert widget

type

```
select_one {list_name}
```

appearance

```
likert
```

A single-select question can be styled as a [Likert scale](#). Options can include text, images or both. If both are provided, images appear above text.

If adding images, note that the images are referenced in the choices sheet, and the image files need to be included in the media folder. See [Including media files in choices](#) to learn more about including images in choices.



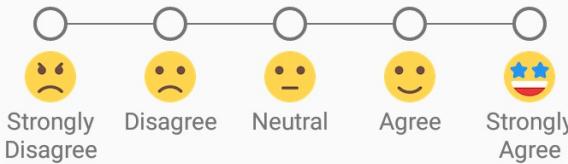
All widgets



Select one widgets

## Likert widget

Images made by Pixel Perfect and downloaded from flaticon.com



## XLSForm

survey				
type	name	label	appearance	hint
select_one likert	likert_widget	Likert Widget	likert	select_one type with Likert appearance, 5 image choices (strongly_disagree.jpg, disagree.jpg, neutral.jpg, agree.jpg, strongly_agree.jpg)
choices				
list_name	name	label	image	
likert_widget	strongly_disagree	Strongly Disagree	strongly_disagree.jpg	
likert_widget	disagree	Disagree	disagree.jpg	
likert_widget	neutral	Neutral	neutral.jpg	
likert_widget	agree	Agree	agree.jpg	
likert_widget	strongly_agree	Strongly Agree	strongly_agree.jpg	

## Select one from map widget

New in version 2022.2.0: ODK Collect v2022.2.0

### ⚠ Warning

The *map* appearance on selects currently only supports single selection of points and is not yet available in web forms (Enketo).

The different [basemap sources](#) currently have different performance. If Collect feels slow when creating the map or when selecting a choice, please describe what you are experiencing [on the forum](#). If you have many choices to include on a map, try a provider other than Google or Mapbox. You can also use a [choice filter](#) to reduce the number of choices that get mapped.

## Note

The only appearance that can combine with selection from map is *quick*.

type

```
select_one {list_name}
```

appearance

```
map
```

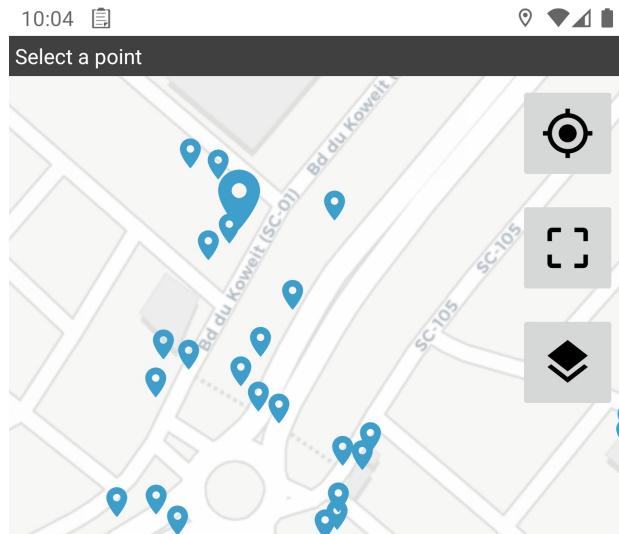
If the choices that you want users to select from have locations, you can display them on a map. Each choice must have a `geometry` property that specifies the choice's geometry.

You can use a [GeoJSON attachment](#) as a source of choices to map. Alternately, you can add a `geometry` column to the **choices** tab or to an [external CSV file](#).

When using a `geometry` column instead of a GeoJSON file, the geometry must be specified in [the ODK format](#). For example, you could attach data collected using another ODK form and make sure that the column containing `geopoint` values has name `geometry`.

## Note

Choices with invalid geometries are silently ignored. There will be no message displayed to a user when it happens.



## Restaurant Délicia

timestamp: 2016-06-20T23:05:32Z

version: 2

changeset: 40169427

user: Jean-Marc Liotier

uid: 160042

amenity: restaurant

id: node/4149452189

Select



When the map is first opened, it centers on the device's current location. There are buttons on the right to recenter on the current location and to show all available points.

Point choices are represented by map markers (●). Tapping on a marker increases its size and displays all of the

choice's properties at the bottom of the screen. Those properties are from:

- additional columns when choices are specified the **choices** tab or an [external CSV file](#)
- the `properties` object when choices are specified in a GeoJSON file

Under the properties, there is a button to save the currently-selected feature to the form.

All of a choice's properties including `geometry` can be used in the rest of the form (see [referencing values in datasets](#)) including in [choice filter](#) expressions. Even if the choices are specified from a GeoJSON file, the `geometry` property is made available to the form in [the ODK format](#), NOT as GeoJSON.

If your geospatial data comes from an external source, you can [customize the label and underlying value](#).

If there is an [offline layer](#) specified, it will be displayed under the mapped choices.

## Select from image widget

type

```
select_one {list_name}, select_multiple {list-name}
```

appearance

```
image-map
```

The image map widget displays an [SVG](#) image with selectable regions.

To make an image with selectable regions:

1. Create or edit an `.svg` source file. Include `id` attributes on any elements you want to be selectable.
2. In the **choices** tab of your XLSForm, put the value of the `id` attributes in the `name` column. Add an appropriate human-friendly `label` to each choice.
3. In the **survey** tab of your XLSForm, put the `.svg` file name in the `image` column.
4. Include the `.svg` file [in your form's media folder](#).

### See also

#### Inkscape

An open source vector graphics editor.

#### SVG Documentation

From Mozilla Developer Network.

#### Free SVG Files

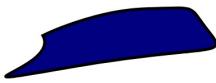
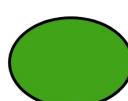
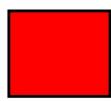
From Wikimedia Commons.



shapes



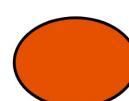
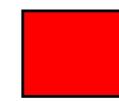
## Choose a shape



shapes

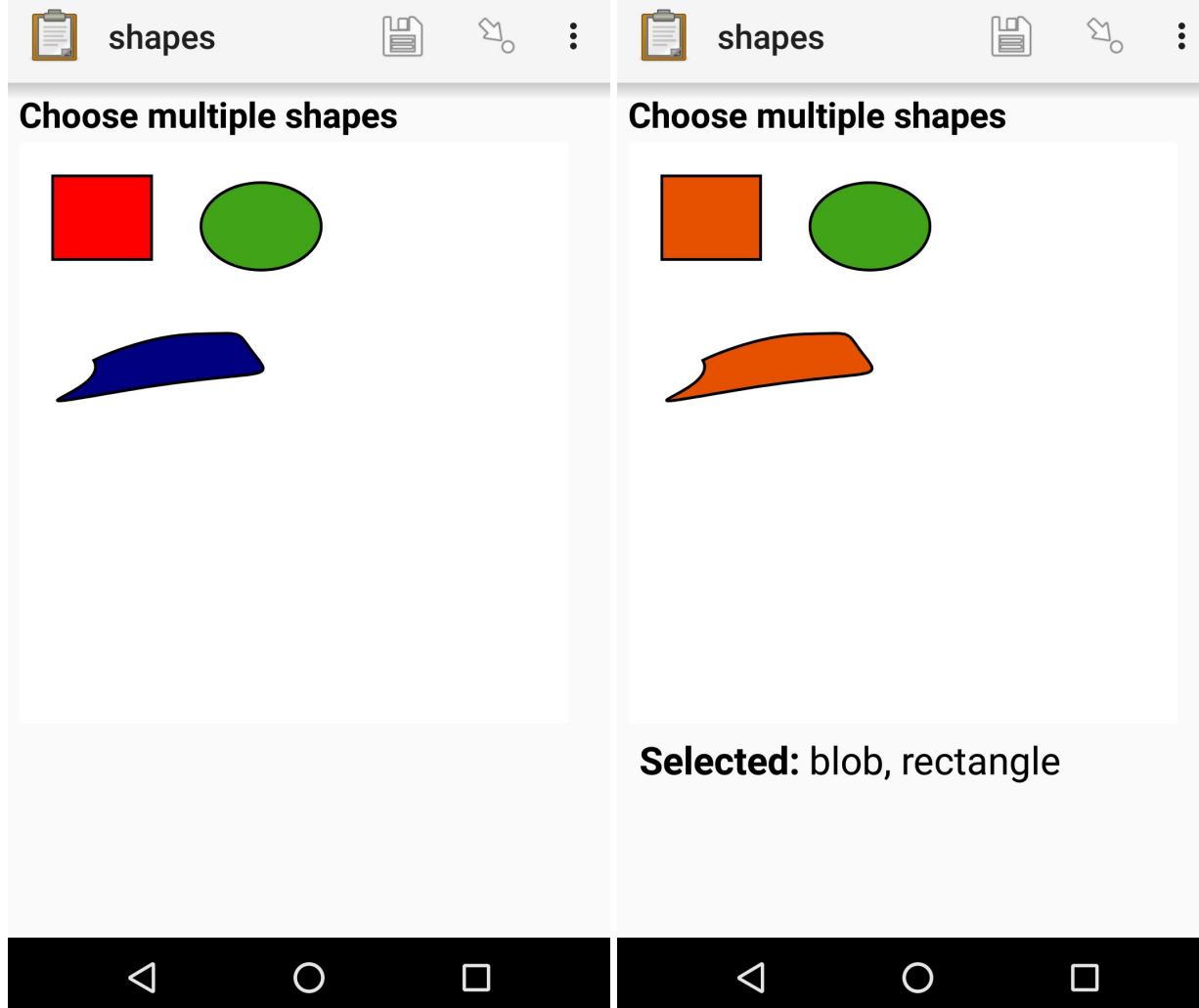


## Choose a shape



**Selected: ellipse**





## SVG

```
<svg width="640" height="480" xmlns="http://www.w3.org/2000/svg" xmlns:svg="http://www.w3.org/2000/svg">
  <title>shapes</title>
  <g>
    <title>Layer 1</title>
    <path id="path" fill="#000080" stroke="#000000" stroke-width="5" d="m125,382c33,56 -193,97 48,55c241,-42 279,-15 241,-45c-15,241 -15,241 -15,241"/>
    <rect id="rect" fill="#FF0000" stroke="#000000" stroke-width="5" x="52" y="53" width="176" height="149"/>
    <ellipse id="ellipse" fill="#41A317" stroke="#000000" stroke-width="5" cx="423" cy="143" rx="107" ry="78"/>
  </g>
</svg>
```

## XLSForm

survey				
type	name	label	appearance	image
select_one shapes	choose-shape	Choose a shape	image-map	shapes.svg
select_multiple shapes	choose-shapes	Choose multiple shapes	image-map	shapes.svg

choices		
list_name	name	label
shapes	path	blob
shapes	rect	rectangle
shapes	ellipse	ellipse

## Including media files in choices

As with questions themselves, choices can include [media](#) (image, video, or audio files):

choices						
list_name	name	label	image	video	audio	
opt_media	a	A	a.jpg			
opt_media	b	B		b.mp4		
opt_media	c	C			c.mp3	

### See also

For images, you can [specify a bigger image for panning and zooming](#) using the `big-image` column. This is not compatible with the `no-buttons` appearance.

### Note

`select_one` and `select_multiple` questions using the `no-buttons` appearances will not display media buttons next to choices. However, if a choice has audio, it will be played when the choice is selected.

## Randomizing choice order

To reduce bias, choice order can be randomized for any of the select question types described above. To display the choices in a different order each time the question is displayed, set **randomize** to **true** in the `parameters` column of the XLSForm **survey** sheet:

### XLSForm

survey			
type	parameters	name	label
select_one opt_abcd	randomize=true	select_one_random_widget	Select one with random choice order set on each display

choices		
list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

In the example above, each time the question is displayed, the choices will be in a different order. It is often preferable to pick one order that the choices will always be displayed in for a given filled form. This can be accomplished by setting an integer seed for the randomization.

### XLSForm

survey				
type	parameters	name	label	calculation
calculate		my_seed		once(substr(decimal-date-time(now()), 10))
select_one opt_abcd	randomize=true,seed=\${my_seed}	select_one_widget	Select one with random choice order set once per filled form	

choices

list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

This seed can also be used to recreate the order choices were displayed in. See [the XForms spec](#) for a description of the randomization algorithm used.

### Note

In the example above, the integer seed is created from the last 8 numbers of the `decimal-date-time()` which is unlikely to repeat across devices. In the seed expression, `once()` is important because it makes sure the seed is not changed if the same filled form is opened more than once.

## Including "other" as a choice

### Warning

We do not recommend using `or_other` because it does not support multiple languages or `choice_filter`. Instead, add your own "other" question and use form logic to have it appear as needed.

On the **survey** sheet, in the `type` column, after the type and the `list_name`, you can add `or_other`. This will add "Other" as an additional option to your choice list. The `name` value of the choice when selected will be `other`.

## Rank widget

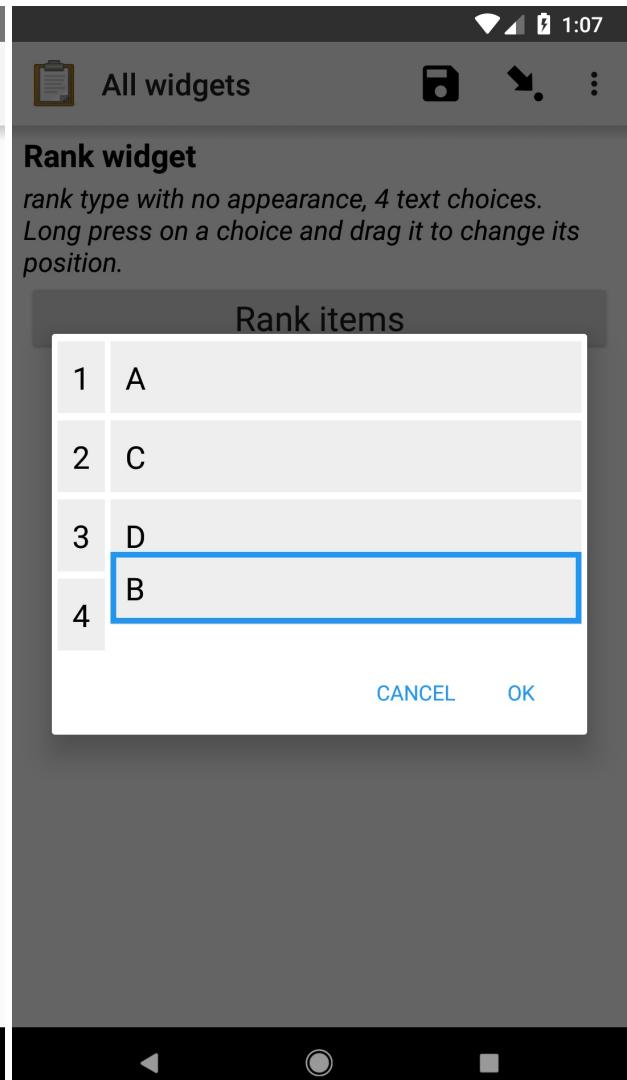
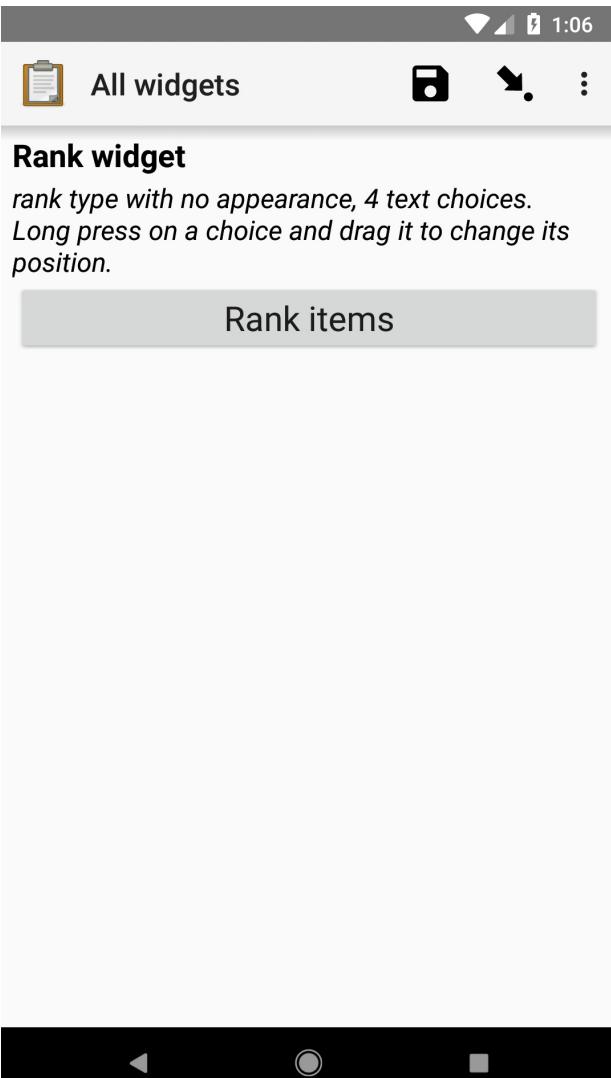
The rank widget allows the user to order options from a list. The value saved in the form and sent to the server is a space-separated ordered list of the options.

Like with [Select widgets](#), the options are listed on a sheet named **choices** in an XLSForm.

To change the order of the options in the list, tap the `Rank items` button. In the resulting dialog, long press on an item and once it gets a border around it, drag it up or down to change the order. If no `default` is provided, the value for the question is blank until the user taps `OK` in the ranking dialog.

`type`

```
rank {list_name}
```





All widgets



## Rank widget

*rank type with no appearance, 4 text choices.  
Long press on a choice and drag it to change its position.*

### Rank items

1. A
2. C
3. D
4. B



## XLSForm

survey

type	name	label	hint
rank opt_abcd	rank_widget	Rank widget	rank type with no appearance, 4 text choices

choices

list_name	name	label
opt_abcd	a	A
opt_abcd	b	B
opt_abcd	c	C
opt_abcd	d	D

## Location widgets

Location widgets capture one or more points representing locations on Earth. Each point is represented as four numbers separated by spaces: latitude, longitude, altitude in meters, and accuracy radius in meters.

For example, if a Collect user captured a point while at the coordinates 12°22'17.0"N 1°31'10.9"W, with a reported accuracy radius of 17.4 meters, and at 305 meters above sea level, the geopoint representation would be:

12.371400 -1.519700 305 17.4

Multiple points that form lines or shapes are separated by semicolons.

### See also

[Select from map](#) for displaying existing geo features on a map for users to select from.

### Note

The accuracy radius is an estimate of what Android calls the [radius of 68% confidence](#): there is a 68% chance that the true location falls within this radius. This is an estimate reported by the Android system based on the available sensors (GPS, network, etc). The accuracy radius itself may be more or less reliable depending on the sensor(s) used and current conditions.

To get an accurate location quickly, ensure devices have a clear view of the sky. For even faster points, consider "warming" the GPS with a [start-geopoint](#) question. See [improving location performance](#) for more.

### Note

Since v1.30, when a mock location provider is detected, the accuracy is set to 0. Achieving such perfect accuracy is not possible using GPS so that indicates it comes from a mock provider.

In v2021.3 and later, you can opt out of this behavior by setting **allow-mock-accuracy** to **true** in the **parameters** column of your question in your XLSForm **survey** sheet. This is useful for external GPS devices that require Android's mock provider feature.

## Geopoint widget

type

geopoint

appearance

none

Captures the current geolocation from the device. The location is displayed in degrees-minutes-seconds (DMS) notation and is stored in [decimal degrees](#) with altitude and accuracy. Learn more about the format of resulting data in [the location widgets section](#).

This question type shows a dialog with the current accuracy and lets the data collector decide when to capture the point. For capturing location without data collector intervention, see [start-geopoint](#). For a geopoint with a user-selected location, see [placement-map](#).

### XLSForm with optional parameters

survey				
type	name	label	hint	parameters
geopoint	geopoint_widget	Geopoint widget	geopoint type	capture-accuracy=10 warning-accuracy=10 allow-mock-accuracy=true

There are three parameters that can be used to customize a `geopoint` question's behavior:

`capture-accuracy` : when the device accuracy reaches this value or better, the point will be automatically captured and the dialog will close. If you always want data collectors to make an explicit decision about accepting a point, set this value to 0. Defaults to 5 (meters), a target that can usually be reached by modern devices given enough time. We generally do not recommend setting this value to below 3 (meters) unless you are using an external GPS device. You can also [set an accuracy constraint](#).

`warning-accuracy` : when the device accuracy is this value or worse, the dialog is red and displays a message stating that the accuracy is unacceptable. There is no enforcement of the threshold so if a data collector needs to capture a point with an unacceptable accuracy (e.g. because they can't wait any longer), they can do so. Set this value to the same value as `capture-accuracy` if you generally always want your data collectors to wait until the point is automatically captured. Defaults to 100 (meters), about the length of a city block. In extreme conditions such as under dense forest canopy, any reported accuracy may be considered acceptable. In that case, you can set this value to a very large number.

`allow-mock-accuracy` : set to `true` to use an external GPS device that uses the mock GPS provider. Otherwise, any location captured from a mock provider will have an accuracy of 0.

A dialog is used to give data collectors feedback on the location they are capturing:



**4** Point will be saved at 5m

**5** Time elapsed: 00:34

**6** Satellites: 0

[Cancel](#) [Save](#)

The dialog is designed to guide the data collector to capture a point with the best reported accuracy possible. The current accuracy is shown at the top of the dialog (1). A message below it (2) gives a qualitative assessment of the accuracy (e.g. unacceptable, poor) and suggested action (e.g. wait). The progress bar (3) gives a visual representation of progress towards an acceptable accuracy.

The bottom half of the dialog displays troubleshooting information. The first line (4) shows the accuracy at which the point will be automatically captured. This is configured by the `capture-accuracy` parameter. You can ask data collectors to watch time elapsed (5) and let you know if it is systematically taking them a long time to get high-accuracy points. This may indicate an issue with their device.

You can also train data collectors to use time elapsed to take some action. For example, you can let them know to capture any point available after waiting for 2 minutes. Number of satellites (6) can be useful when capturing points outdoors. A low number of satellites (under 4) may indicate that something is wrong with the device or its position. See [Improving Location Performance](#).

#### Tip

You can use `selected-at()` to require geopoints meet a particular threshold. For example, if you need points with an accuracy better than 10 meters, use this constraint:

`selected-at(${geopoint_widget}, 3) < 10.`

The `3` in the above constraint references accuracy, the third value in the [geopoint data type](#). Use `1` to reference latitude, `2` for longitude, and `4` for altitude.

## Geopoint with map display

type

`geopoint`

appearance

`maps`

The default [Geopoint widget](#) does not display a map to the user. When the appearance attribute is `maps`, the widget displays a map to help the user get oriented and confirm that the selected point is correct and sufficiently accurate.

When the device's geolocation is available, it is displayed on the map by a blue cross. A blue shaded circle around the cross represents the accuracy radius of the geolocation. The "add marker" button at the top right of the screen can be tapped to add a point at the location indicated by the middle of the blue cross. The selected point is represented by a small circle with a red outline.

When the map view is opened again with a selected point, the map is centered on that point. To change the selection, first tap the "trash" icon and then select a new point.

For a geopoint with a location that the user can manually select or adjust, see [Geopoint with user-selected location](#).

survey				
type	name	label	appearance	hint
geopoint	geopoint_widget_maps	Geopoint widget	maps	geopoint type with maps appearance

## Geopoint with user-selected location

type

geopoint

appearance

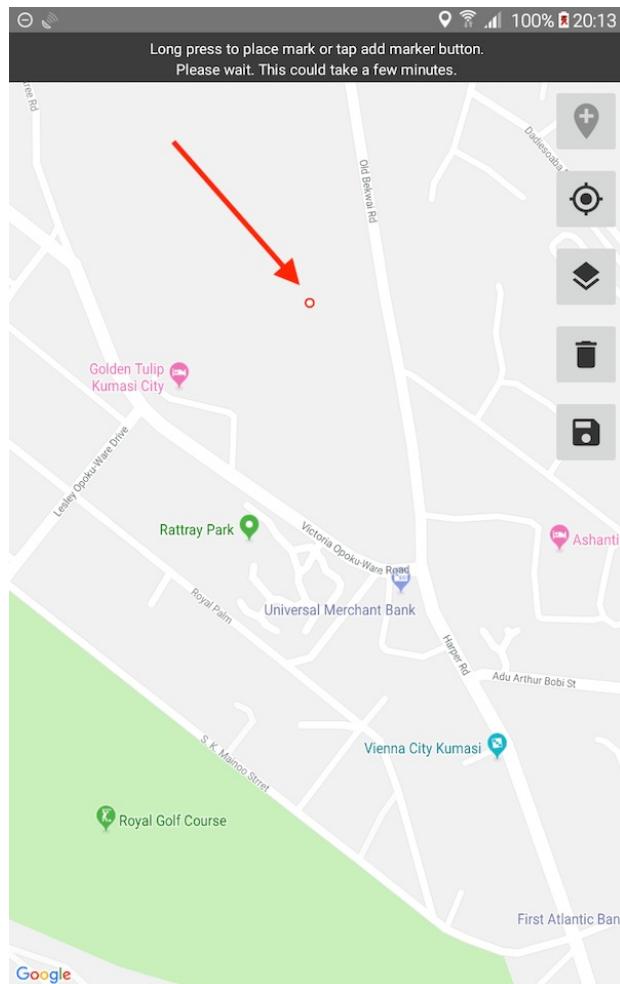
placement-map

The default [Geopoint widget](#) does not allow the user to place the point anywhere other than the device's current geolocation.

A geopoint with the appearance attribute `placement-map` allows the user to select any point from a map. The user can either long press to place the point anywhere, or, if the device knows its geolocation, tap on the "add point" button at the top right of the screen. The selected point is represented by a small circle with a red outline (see arrow in screenshot).

The save button saves the selected point and returns to the question screen. If the point was selected by long pressing, the accuracy radius and altitude will both be 0. If the device's geolocation was selected, the accuracy radius will be greater than 0.

When the map view is opened again with an existing point, the map is centered on the selected point. To change the selection, first tap the "trash" icon and then select a new point.



<b>type</b>	<b>name</b>	<b>label</b>	<b>appearance</b>	<b>hint</b>
geopoint	geopoint_widget_placementmap	Geopoint widget	placement-map	geopoint type with placement-map appearance

## Geotrace widget

type

`geotrace`

appearance

`none`

A series of points. Identical to [geoshape](#) except that the first and last point may be different and at least 2 points are required.

Points can be entered either by tapping the screen to place each point, or by taking readings of the device's geolocation over time. On a map, each coordinate is represented by small circles with red outlines. These are connected by red lines.

To collect a geotrace, first select the location-recording mode by tapping the "add point" button in the upper right side of the screen. The selected mode will be displayed in the gray bar at the bottom of the screen. While point collection is ongoing, the "add marker" button changes to a "pause" button. The "back arrow" button can be used to remove the last-entered point either when actively collecting points or when paused. Any point can be manually moved at any time by tapping on it and dragging it. The mode can only be changed if an existing line is first cleared by tapping the "trash" button. Recording must be paused to clear the existing line.

 Tip

Points that were entered by tapping or adjusted by dragging will always have an accuracy radius of 0. Points that were read from the device location will never have an accuracy radius of 0.

Once the trace has been saved, the coordinates of its points will be displayed on the question screen. The trace can be opened for manual editing by tapping to add more points, moving existing points or deleting the last-added point. After a trace has been saved once, it cannot be added to in manual or automatic location recording modes.

The three location recording modes are:

Placement by tapping

The user taps the device to place points.

Manual location recording

The user chooses when to tap the "record a point" button at the top of the screen to capture the device geolocation at that moment.

Automatic location recording

The user is prompted to select a recording interval and accuracy requirement. If the accuracy requirement is set to None, points are always collected at the recording interval. If the accuracy requirement is set to any other value, a point will only be captured if it meets the requirement. For example, given a recording interval of 20s and an accuracy requirement of 10m, the app places a point at the device location every 20s if the location is accurate to 10m or better.

 Warning

If you are using Aggregate and you would like to collect more than 5 points at a time, you should [increase the database field length to over 255 characters](#). Otherwise, additional points will be lost.

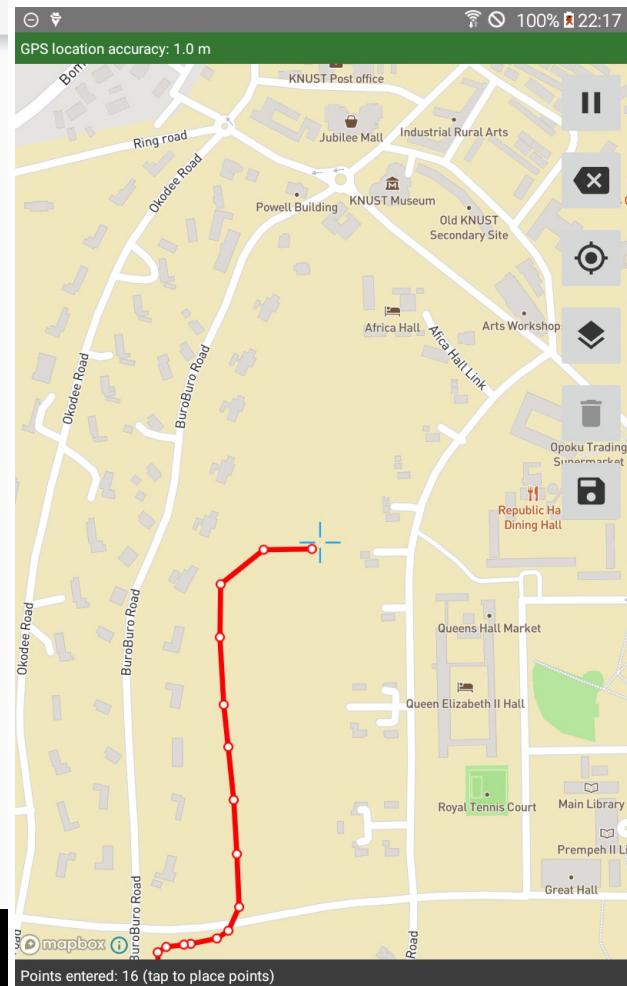


Sample XLSFo...



Where have you been?

Start GeoTrace



## XLSForm

survey		
type	name	label
geotrace	trace_example	Where have you been?

## Geoshape

type

geoshape

appearance

none

A series of points that form a closed polygon. Identical to [geotrace](#) except that the first and last point are always the same and at least 3 points are required.

Points can be entered either by tapping the screen to place each point, or by taking readings of the device's geolocation over time. On a map, each coordinate is represented by small circles with red outlines. These are connected by red lines.

To collect a geoshape, first select the location-recording mode by tapping the "add point" button in the upper right side of the screen. The selected mode will be displayed in the gray bar at the bottom of the screen. While point collection is ongoing, the "add marker" button changes to a "pause" button. The "back arrow" button can be used to remove the last-entered point either when actively collecting points or when paused. Any point can be manually moved at any time by tapping on it and dragging it. The mode can only be changed if an existing line is first cleared by tapping the "trash" button. Recording must be paused to clear the existing line.

## Tip

Points that were entered by tapping or adjusted by dragging will always have an accuracy radius of 0. Points that were read from the device location will never have an accuracy radius of 0.

Once the shape has been saved, the coordinates of its points will be displayed on the question screen. The shape can be opened for manual editing by tapping to add more points, moving existing points or deleting the last-added point. After a shape has been saved once, it cannot be added to in manual or automatic location recording modes.

The three location recording modes are:

### Placement by tapping

The user taps the device to place points.

### Manual location recording

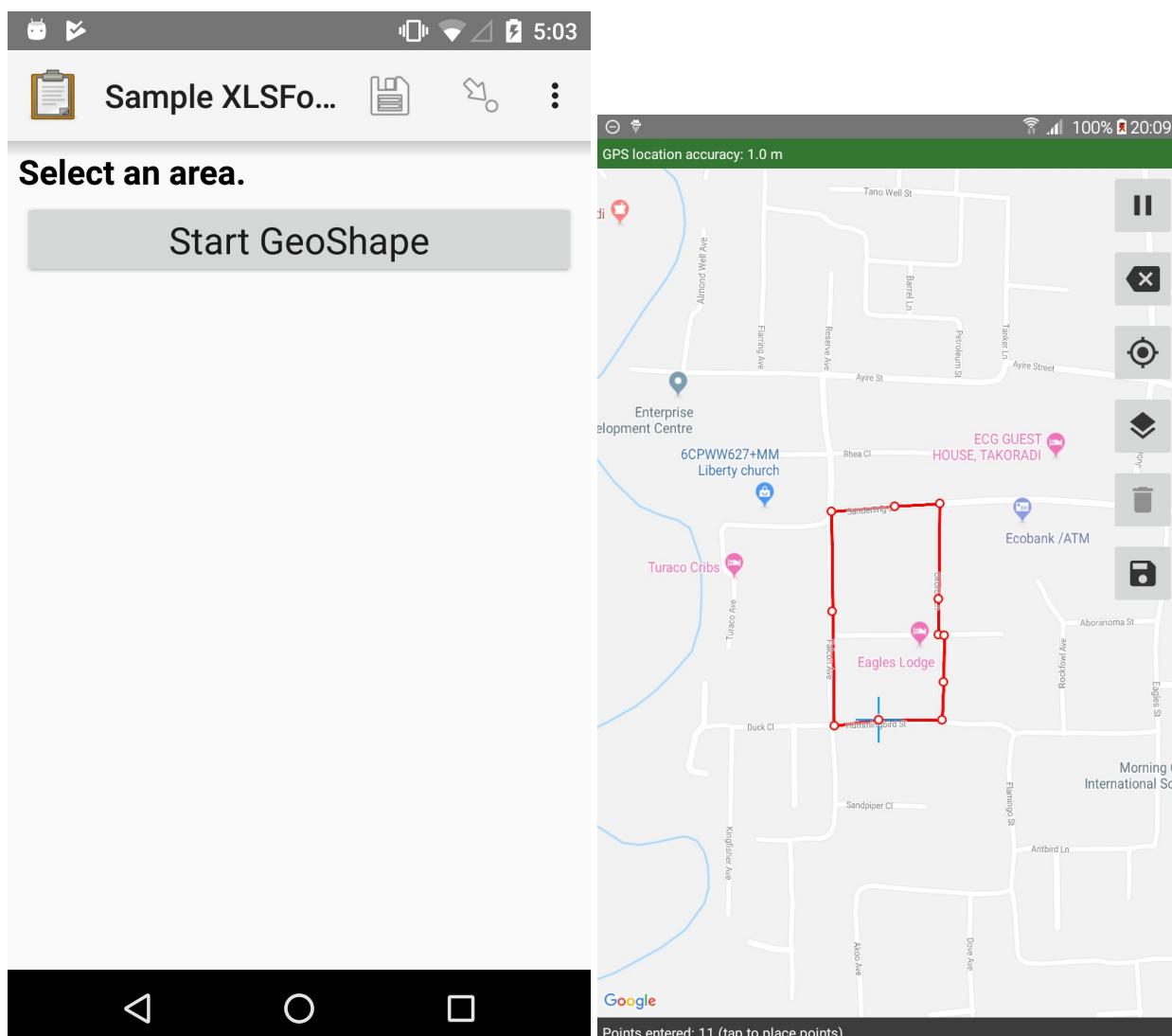
The user chooses when to tap the "record a point" button at the top of the screen to capture the device geolocation at that moment.

### Automatic location recording

The user is prompted to select a recording interval and accuracy requirement. If the accuracy requirement is set to None, points are always collected at the recording interval. If the accuracy requirement is set to any other value, a point will only be captured if it meets the requirement. For example, given a recording interval of 20s and an accuracy requirement of 10m, the app places a point at the device location every 20s if the location is accurate to 10m or better.

### Warning

If you are using Aggregate and you would like to collect more than 5 points at a time, you should [increase the database field length to over 255 characters](#). Otherwise, additional points will be lost.



survey		
type	name	label
geoshape	shape_example	Select an area

## Calculating the area of a geoshape

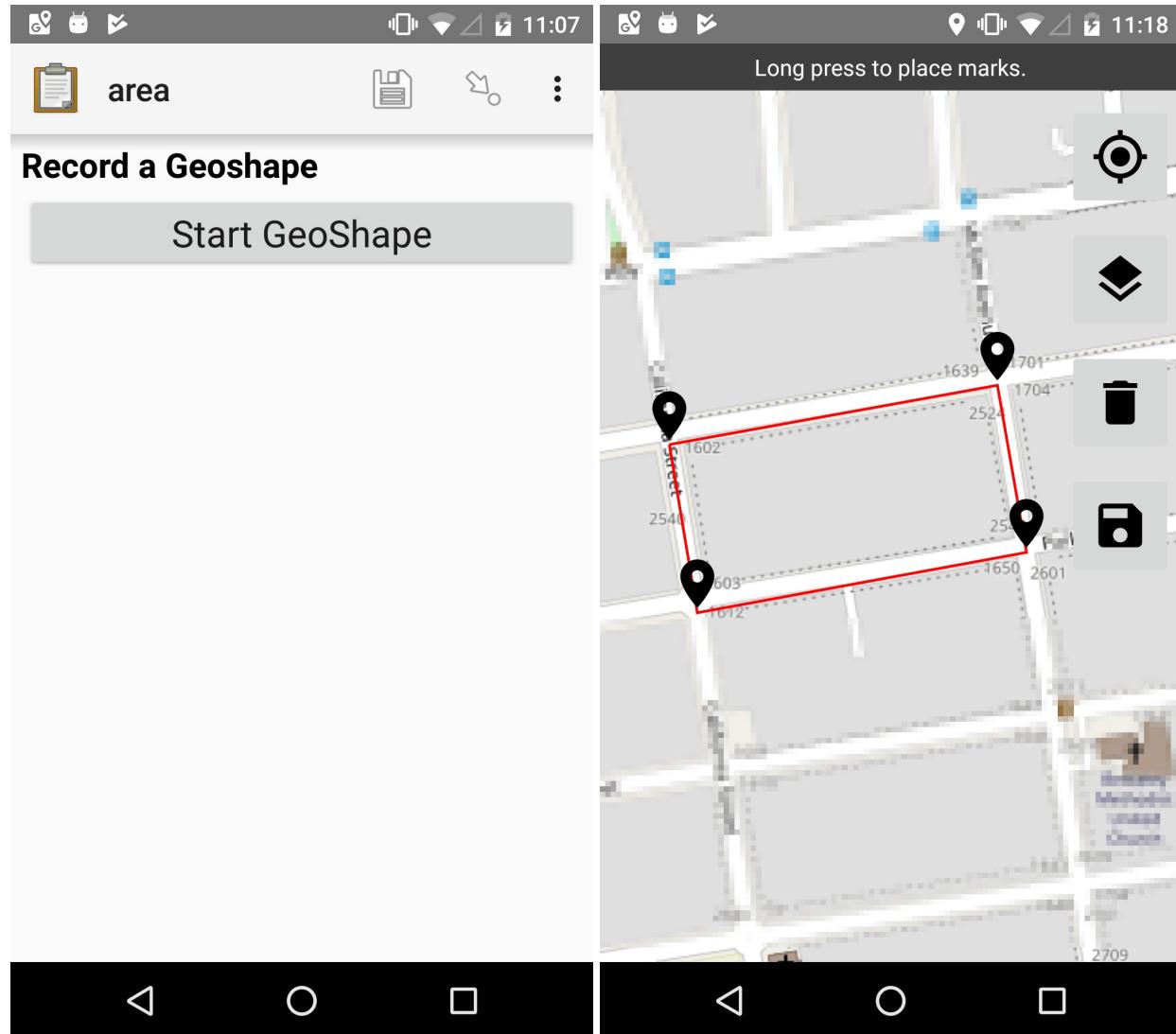
type

calculate

calculation

`area(${geoshape})`

The `area()` function calculates the land area, in square meters, of a polygon defined in a [Geoshape](#). The value will be included in your completed survey data, and can also be used in later widgets in the form.





area



## Record a Geoshape

View or Change GeoShape



area



The area of the recorded geoshape  
is:

**19322.2 m<sup>2</sup>**



## XLSForm

type	name	label	calculation
geoshape	shape	Record a Geoshape	
calculate	shape_area		area(\${shape})
calculate	rounded_shape_area		round(\${shape_area}, 2)
note	shape_area_note	The area of the recorded geoshape is: \${rounded_shape_area} m <sup>2</sup>	

## Bearing widget

type

decimal

appearance

bearing

Captures a compass reading, which is stored as a decimal.



All widgets

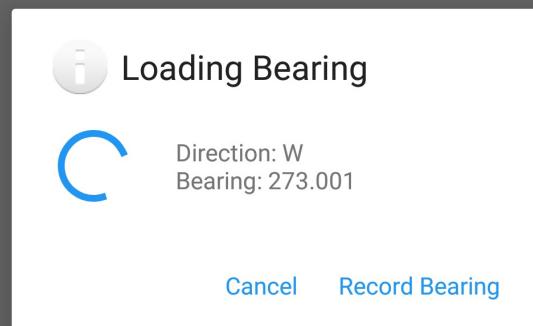


Numerical widgets

## Bearing widget

*decimal type with bearing appearance*

Record Bearing





All widgets



Numerical widgets

## Bearing widget

*decimal type with bearing appearance*

Replace Bearing

271.538



## XLSForm

survey				
type	name	label	appearance	hint
decimal	bearing_widget	Bearing widget	bearing	decimal type with bearing appearance

## OpenMapKit widget

[OpenMapKit](#) allows you to add questions about OpenStreetMap features in a Collect-rendered form.For more details, see the [OpenMapKit](#) documentation.

## Image widgets

### Tip

Image files can be very large. We recommend always including a [maximum image size in form design](#). Also, consider making test submissions to your server with the Internet conditions you expect when gathering data to make sure that you can send files of the size you expect.

## Default image widget

type

`image`

appearance

`none`

Captures an image from the device. The user can choose to take a new picture with the device camera, or select an image from the device photo gallery.



All widgets



Image widgets

**Image widget***image type with no appearance***Take Picture****Choose Image****XLSForm**

survey

<b>type</b>	<b>name</b>	<b>label</b>	<b>hint</b>
image	image_widget	Image widget	image type with no appearance

**Image widget with annotation**

type

`image`

appearance

`annotate`Adding the `annotate` appearance allows the user to draw on the image before submitting it.**Tip**

If you have a standard image to annotate, you can add that image's filename in the `default` column. For example, put `template.png` in the `default` column and Central will prompt you to attach a png to the form. Anyone who fills out the form will see the same image.

To enforce that this default image gets annotated, you can use a constraint such as `not(. = 'jr://images/template.png')`. This works because Collect renames images after annotation.

Also see [select from image](#).



1:47



All widgets



Image widgets

**Annotate widget***image type with annotate appearance*

Take Picture

Choose Image

Markup Image





1:51



All widgets



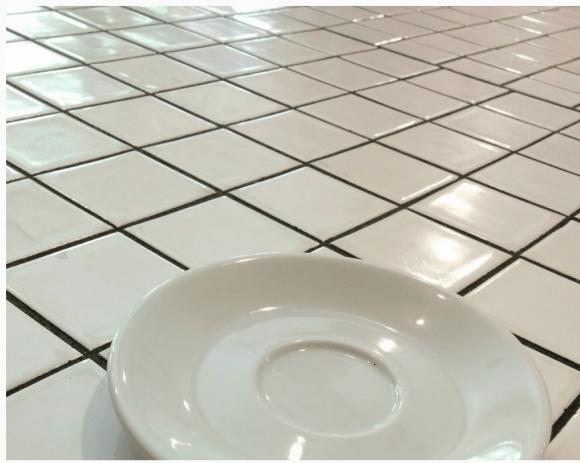
Image widgets

**Annotate widget***image type with annotate appearance*

Take Picture

Choose Image

Markup Image





## XLSForm

survey					
type	name	label	appearance		hint
image	annotate_image_widget	Annotate widget	annotate		image type with annotate appearance

## Image widget with required new image

type

image

appearance

new

An image widget that does not include a `Choose Image` button. This requires the user to take a new picture.



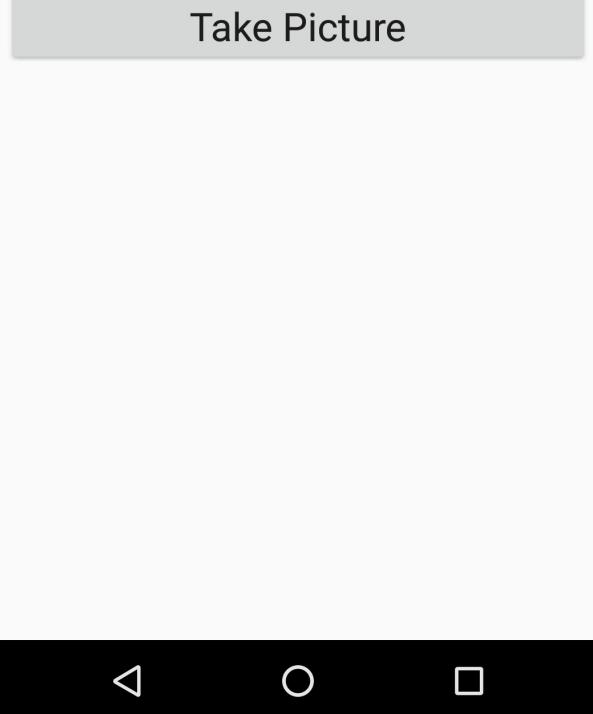
All widgets



Image widgets

**Image widget without Choose button**

*image type with new appearance (can also be added with annotate appearance and on audio and video types)*


Take Picture

**XLSForm**

type	name	label	appearance	hint
image	image_widget_no_choose	Image widget without Choose button	new	image type with new appearance (can also be added with annotate appearance and on audio and video types)

**Self portrait (*selfie*) image widget**

type

`image`

appearance

`new-front`

Takes a picture using the front-facing ("selfie") camera. The `Choose image` button is not displayed.

*Changed in version 1.15:* Prior to v1.15, the appearance attribute for this was `selfie`. The old appearance attribute will continue to work on existing forms, but new forms should use the `new-front` appearance.



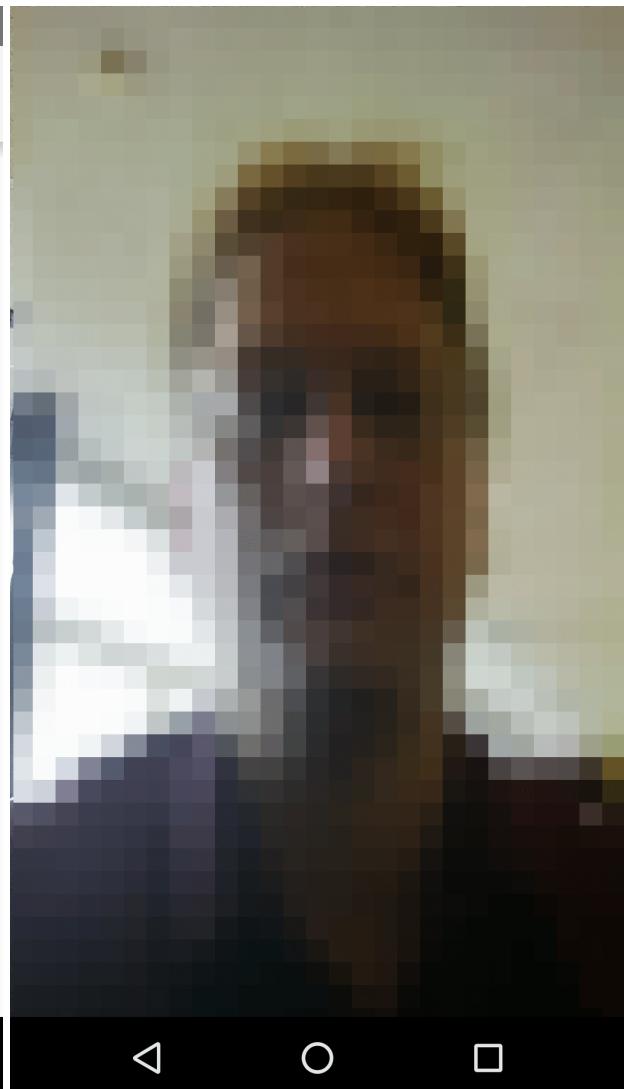
self-portrait-w...

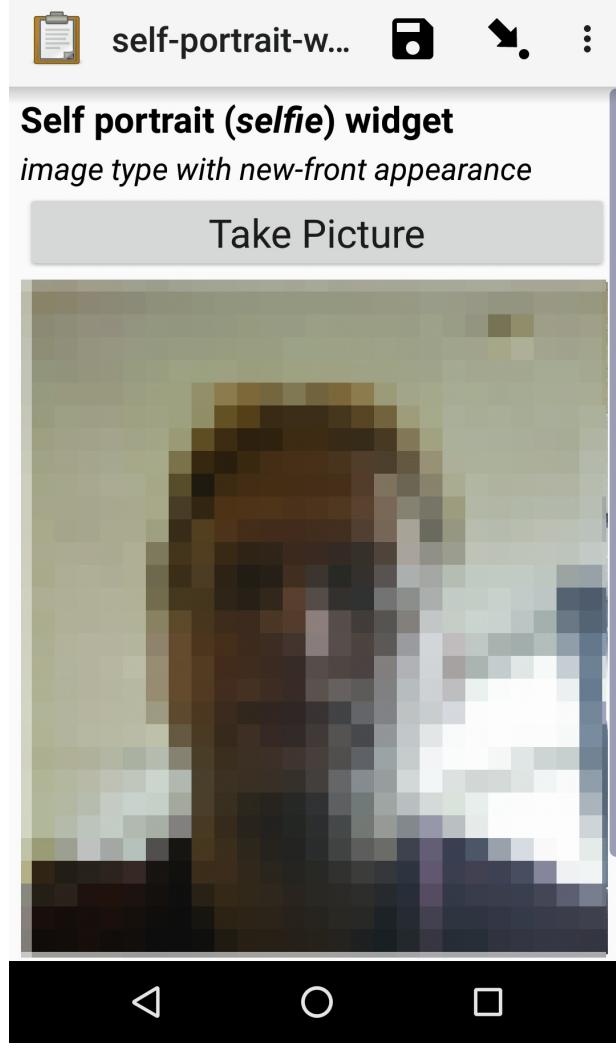


## Self portrait (selfie) widget

*image type with new-front appearance*

Take Picture





## XLSForm

survey

<b>type</b>	<b>name</b>	<b>label</b>	<b>hint</b>	<b>appearance</b>
image	self-portrait	Self portrait ( <i>selfie</i> ) widget	image type with new-front appearance	new-front

## External app image widget

New in version 1.30.

Launches an external app and receives an image back from the external app. If the specified external app is not available, it is not possible to use the widget.

The external app image widget is displayed when the `appearance` attribute begins with `ex:`. The rest of the `appearance` string specifies the application to launch.

See also

[Launching External Apps](#)

12:54



## All widgets



Image widgets

### External image widget

image type with ex:com.example.collectanswers provider(questionImage="") appearance (can use other external apps)

[Launch](#)



< BACK

NEXT >



## XLSForm

survey

type	name	label	appearance	
image	ex_image_widget	External image widget	ex:com.example.collectanswersprovider(questionImage="")	image type with ex:com.example.co appearance (can us

## Draw widget

type

[image](#)

appearance

[draw](#)

Provides the user a drawing pad and collects the drawn image.



1:29



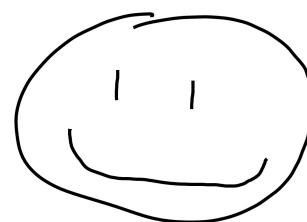
All widgets

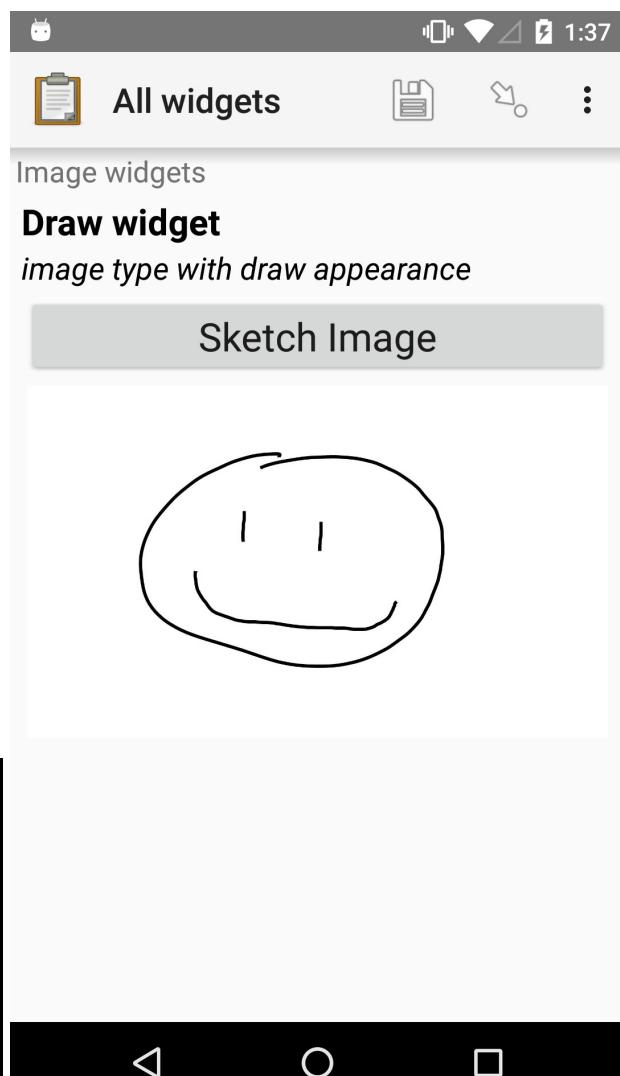


Image widgets

**Draw widget***image type with draw appearance*

Sketch Image





## XLSForm

survey					
type	name	label	appearance		hint
image	draw_image_widget	Draw widget	draw		image type with draw appearance

## Scaling down images

Images created with any of the image widgets described above can be automatically scaled down on save by using the `max-pixels` parameter. If the long edge of the image is larger than the maximum size specified, the image is resized proportionally so that the long edge matches the provided pixel value. This is useful to reduce the upload size when bandwidth is limited.

### ⚠ Warning

All scaled down jpg images are saved with 80% quality. That means in some rare cases when:

- a jpg image is attached not captured
- and the attached file has quality lower than 80%
- and the difference between its original size and the value specified using `max-pixels` is not big enough

the size of the output image might be even bigger than the original one.

Available in Collect since v1.10.0 and in XLSForm since 7/2018.

## XLSForm

In the parameters column, write `max-pixels=` followed by the desired maximum length of the long edge in pixels.

survey					
type	name	label	parameters		hint
image	my_scaled_image	Scaled image	max-pixels=1024		image scaled to a max long edge of 1024 pixels

# Audio widgets

## Default audio widget

type

audio

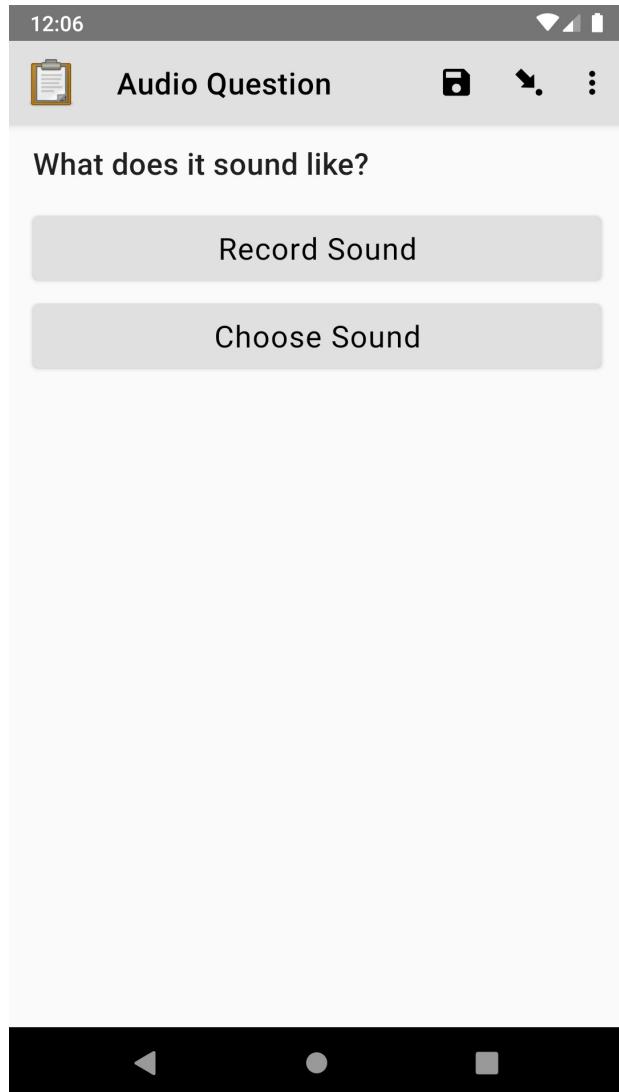
appearance

none

Records audio using the device's microphone or a connected external microphone. By default, an [internal recorder](#) is used.

### Tip

We recommend you use the [built-in audio recorder](#) because you can customize audio quality and record while filling out other questions. Built-in recording is available in Collect v1.29 or later.



## XLSForm

survey		
type	name	label
audio	bird_recording	What does it sound like?

## Tip

Audio files can be very large so if you record audio in your form, make sure that you consider your audio quality settings. Also, consider making test submissions to your server with the Internet conditions you expect when gathering data to make sure that you can send files of the size you expect.

Android devices can make many sounds during use and these will be included in recordings. We recommend turning off sounds from button presses, camera shutters and notifications before recording.

## Using the built-in audio recorder

type

audio

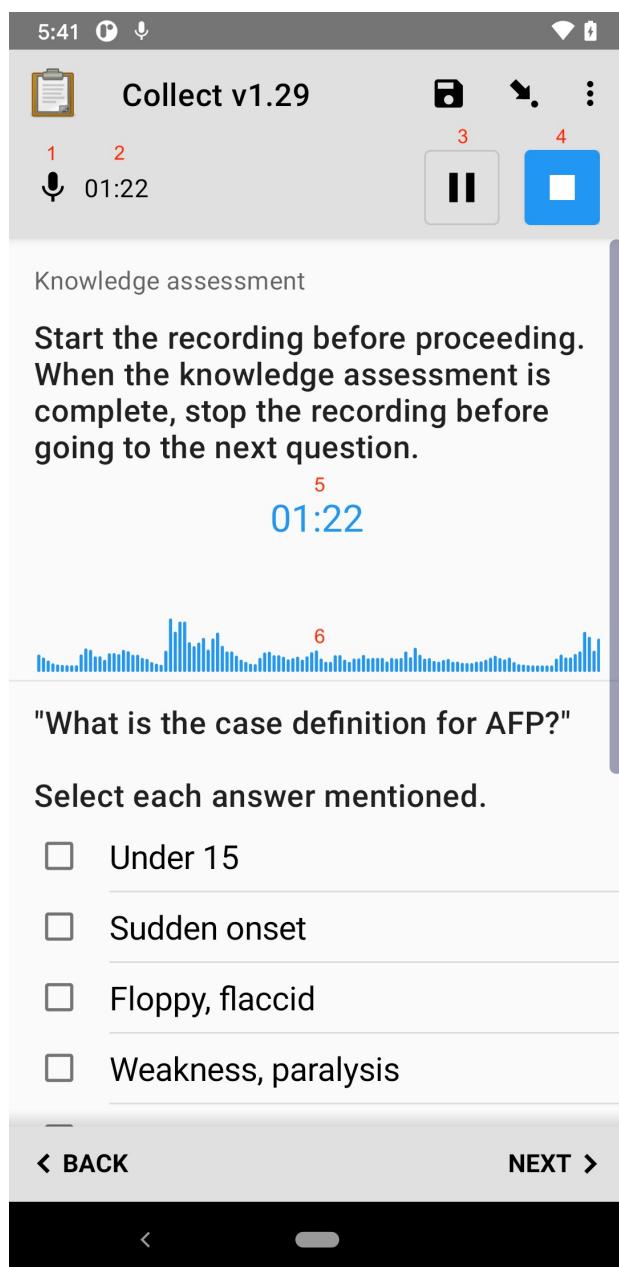
appearance

none

New in version 1.29: [ODK Collect v1.29.0](#)

The built-in audio recorder makes it possible to capture audio without having to install an external app.

It also enables recording while filling out other questions and is designed to continue recording even if the user switches to another app or if the phone screen is locked.



When built-in audio recording is enabled and recording is initiated, a recording control bar appears at the top of the screen. At the top left of this bar is an icon to represent whether recording is currently ongoing or paused (1). To the right of this icon is the current length of the recording (2).

## ⚠ Warning

Pause is only available on Android 7.0 and above. On lower Android versions, the pause button is hidden.

At the right of the control bar are a pause button (3) and a stop button (4). When the pause button is tapped, recording is temporarily suspended and the button icon changes to a microphone. When the microphone is tapped, recording is resumed. Recording can be paused and resumed as many times as desired. When the stop button is tapped, the recording is ended and can no longer be modified.

Recording status is also displayed below the audio question text. There is a time representing the current length of the recording (5) and a diagram (6) representing the volume of the recording over time. The diagram provides confirmation that the microphone is working and can help a user ensure an even, sufficient volume.

Other questions can be included on the same screen as a built-in recording question. As shown in the screenshot above, this makes it possible to capture quantitative content while recording. To achieve this, put the questions in a [field list](#).

During recording, the user is prevented from leaving the current question screen. However, it is safe to use other applications or to lock the device screen.

Once recording is stopped, the control bar disappears. The recording is made available for playback below the question text.

To replace the audio captured, first delete the current file and then record again.

In some rare cases such as the device running out of space, the recording may complete successfully but not be attached to the form. If this happens, a dialog will be displayed explaining that the file is available but needs to be accessed manually. You can find these files in the `recordings` folder of the [Collect directory](#). This folder is never cleared so consider emptying it yourself once you have retrieved its files.

## Customizing audio quality

*New in version 1.29:*

[ODK Collect v1.29.0](#), Central v1.1.0.

The quality of audio recordings can be customized using the `quality` parameter. If a `quality` is specified, the built-in recorder is always used, regardless of Collect settings. If no `quality` is specified and [external app recording has been disabled](#), `normal` is used. The available quality values are:

Value	Extension	Encoding	Bit rate	Sample rate	File size
normal	.m4a	AAC	64kbps	32kHz	~30MB/hour
low	.m4a	AAC	24kbps	32kHz	~11MB/hour
voice-only	.amr	AMR	12.2kbps	8kHz	~5MB/hour

### Tip

We'd recommend only using `voice-only` for one-on-one interviews in a quiet place as otherwise there might be too much detail loss. `low` will sound compressed but speech is generally intelligible, even if multiple people are talking at once. `normal` is similar to typical podcast settings and will sound good on most devices.

It's a good idea to test the different qualities out with the device (and any other equipment) you'll be using in the field to see which one fits your use case and setup best.

## XLSForm

In the parameters column, write `quality=` followed by the desired value.

survey			
type	name	label	parameters
audio	voice_only_audio	Voice audio	quality=voice-only
audio	normal_audio	Normal audio	quality=normal

## Changing audio quality during form entry

If it's a possibility that an individual question could need different qualities depending on context you can use [relevance](#)

to switch between them:

## XLSForm

survey				
type	name	label	parameters	relevance
select_one yes_no	is_quiet	Are you currently in a quiet location with only one person speaking at a time?		
audio	recording_voice_only	Please record	quality=voice-only	<code> \${is_quiet} = 'yes'</code>
audio	recording_normal	Please record	quality=normal	<code> \${is_quiet} = 'no'</code>

choices		
list_name	name	label
yes_no	yes	Yes
yes_no	no	No

## Recording with an external app

type

audio

appearance

none

parameters

`quality=external`

Setting `quality` to `external` will cause Collect to use an external app to record audio rather than the built-in recorder. You can also [configure Collect to always use an external app for recording](#) and set no `quality` parameter.

Some Android devices provide a default application for audio recording. Others do not, and the user will need to install an audio recording app.

Any app that responds to `android.provider.MediaStore.Audio.Media.RECORD_SOUND_ACTION` should be compatible. We recommend [Axet Audio Recorder](#).

## Getting audio from a custom external app

*New in version 1.30.*

Launches an external app and receives an audio file back from the external app. If the specified external app is not available, it is not possible to use the widget.

The external app audio widget is displayed when the `appearance` attribute begins with `ex:`. The rest of the `appearance` string specifies the application to launch.

 See also

[Launching External Apps](#)



## All widgets



Media widgets

**External audio widget**

audio type with ex:com.example.collectanswers provider(questionAudio="") appearance (can use other external apps)

**Launch**

&lt; BACK

NEXT &gt;

**XLSForm**

survey

<b>type</b>	<b>name</b>	<b>label</b>	<b>appearance</b>	
audio	ex_audio_widget	External audio widget	ex:com.example.collectanswersprovider(questionAudio="")	audio type with ex:com.example.collectanswers provider(questionAudio="") appearance (can use other external apps)

**Video widgets**

Video files can be very large. We recommend configuring video options for every device you intend to use for data collection. Also make submissions to your server with the Internet conditions you expect when gathering data to make sure that you can send files of the size you expect. Note that Central has a [100MB file upload size limit by default](#).

**Default video widget**

Records video, using the device camera.



Sample XLSFo...



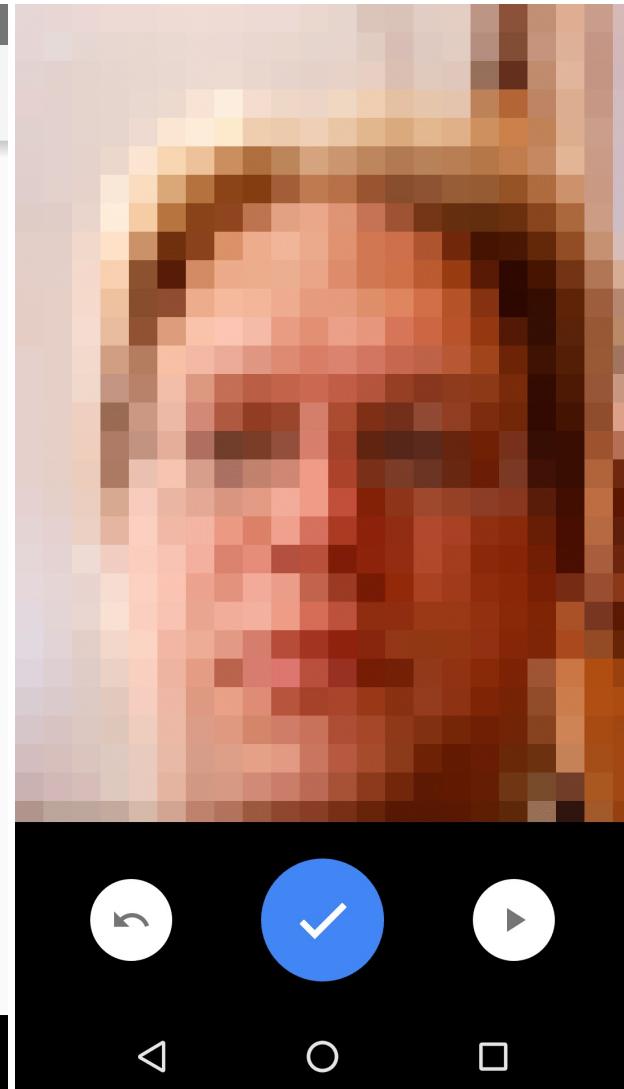
**Please record a video of yourself  
blinking.**

*Three times is probably sufficient.*

Record Video

Choose Video

Play Video





**Please record a video of yourself blinking.**

*Three times is probably sufficient.*

Record Video

Choose Video

Play Video



## XLSForm

survey

type	name	label	hint
video	blinking	Please record a video of yourself blinking.	Three times is probably sufficient.

## External app video widget

*New in version 1.30.*

Launches an external app and receives a video file back from the external app. If the specified external app is not available, it is not possible to use the widget.

The external app video widget is displayed when the `appearance` attribute begins with `ex:`. The rest of the `appearance` string specifies the application to launch.

See also

[Launching External Apps](#)



Media widgets

### External video widget

video type with ex:com.example.collectanswers provider(questionVideo="") appearance (can use other external apps)

**Launch****Play Video**

&lt; BACK

NEXT &gt;



### XLSForm

survey

<b>type</b>	<b>name</b>	<b>label</b>	<b>appearance</b>	
video	ex_video_widget	External video widget	ex:com.example.collectanswersprovider(questionVideo="")	video type with ex:com.example.collectanswers provider(questionVideo="") appearance (can use other external apps)

## File upload widget

### Default file upload widget

New in version 1.15: [ODK Collect v1.15.0](#)

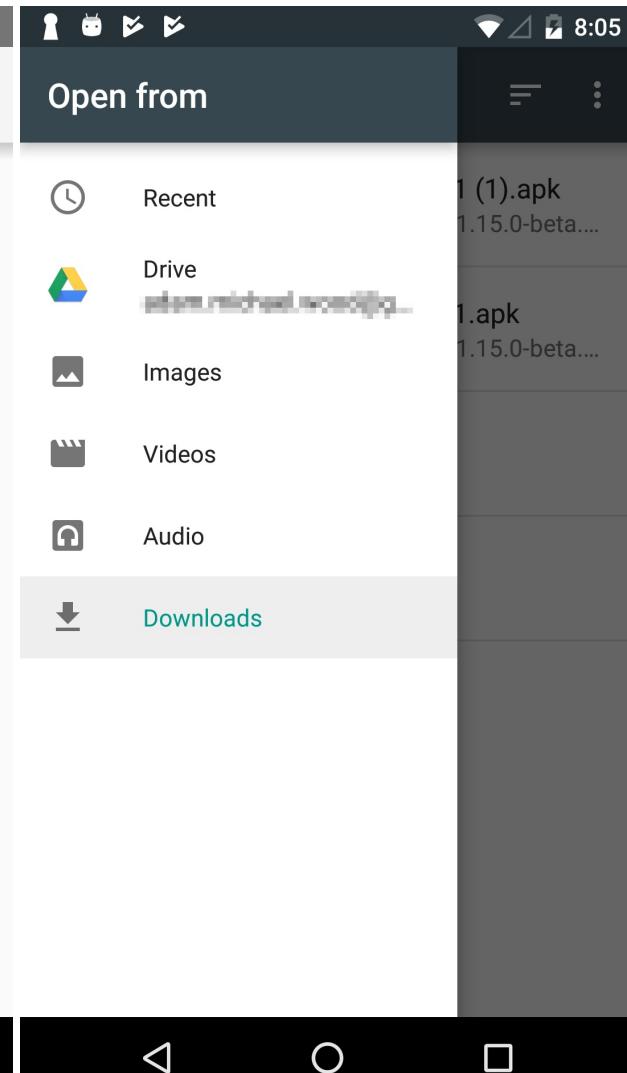
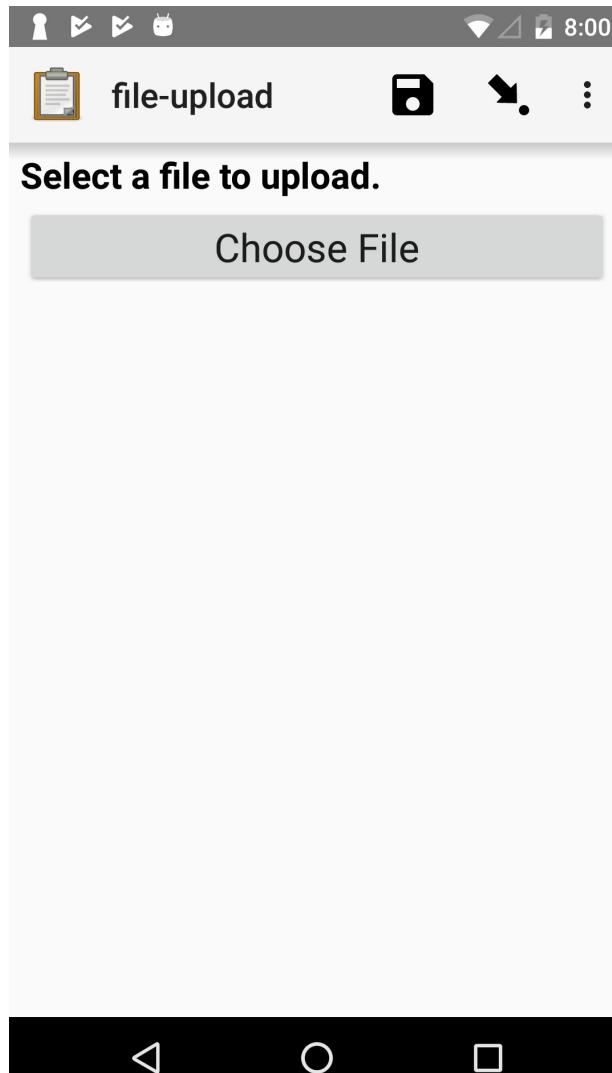
Uploads any file from the device to the form.

**⚠ Warning**

Users can upload **any** file type, which includes potentially malicious files. You should not include this widget unless you trust the people using the form.

Even then, you should take precautions before downloading or opening files.

- Run an antimalware scan.
- Verify the file is a type you expect (such as a `.pdf` document), and not a **potentially dangerous file** (such as `.exe` or `.ini`).



## XLSForm

survey		
type	name	label
file	some-file	Select a file to upload.

## External app file widget

New in version 1.30.

Launches an external app and receives an arbitrary file back from the external app. If the specified external app is not available, it is not possible to use the widget.

The external app file widget is displayed when the `appearance` attribute begins with `ex:`. The rest of the `appearance` string specifies the application to launch.

### See also

[Launching External Apps](#)

### Warning

This widget accepts files of any type. Learn more about the risk [above](#). You should only specify an external application that you trust.



Media widgets

**External file widget**

file type with ex:com.example.collectanswersprovider(questionFile="") appearance (can use other external apps)

**Launch**

1612958132936.pdf

&lt; BACK

NEXT &gt;

**XLSForm**

survey

type	name	label	appearance	
file	ex_file_widget	External file widget	ex:com.example.collectanswersprovider(questionFile="")	file type with ex:com.example.collectanswersprovider(questionFile="") appearance (can use other external apps)

**Barcode widget**

Scans, decodes, and captures the content of a barcode, using the device camera.

The following barcode formats are supported:

- UPC-A
- UPC-E
- EAN-8
- EAN-13
- Code 39
- Code 93
- Code 128
- Codabar
- ITF
- RSS-14
- RSS-Expanded
- QR Code
- Data Matrix
- Aztec (beta)
- PDF 417 (beta)
- MaxiCode

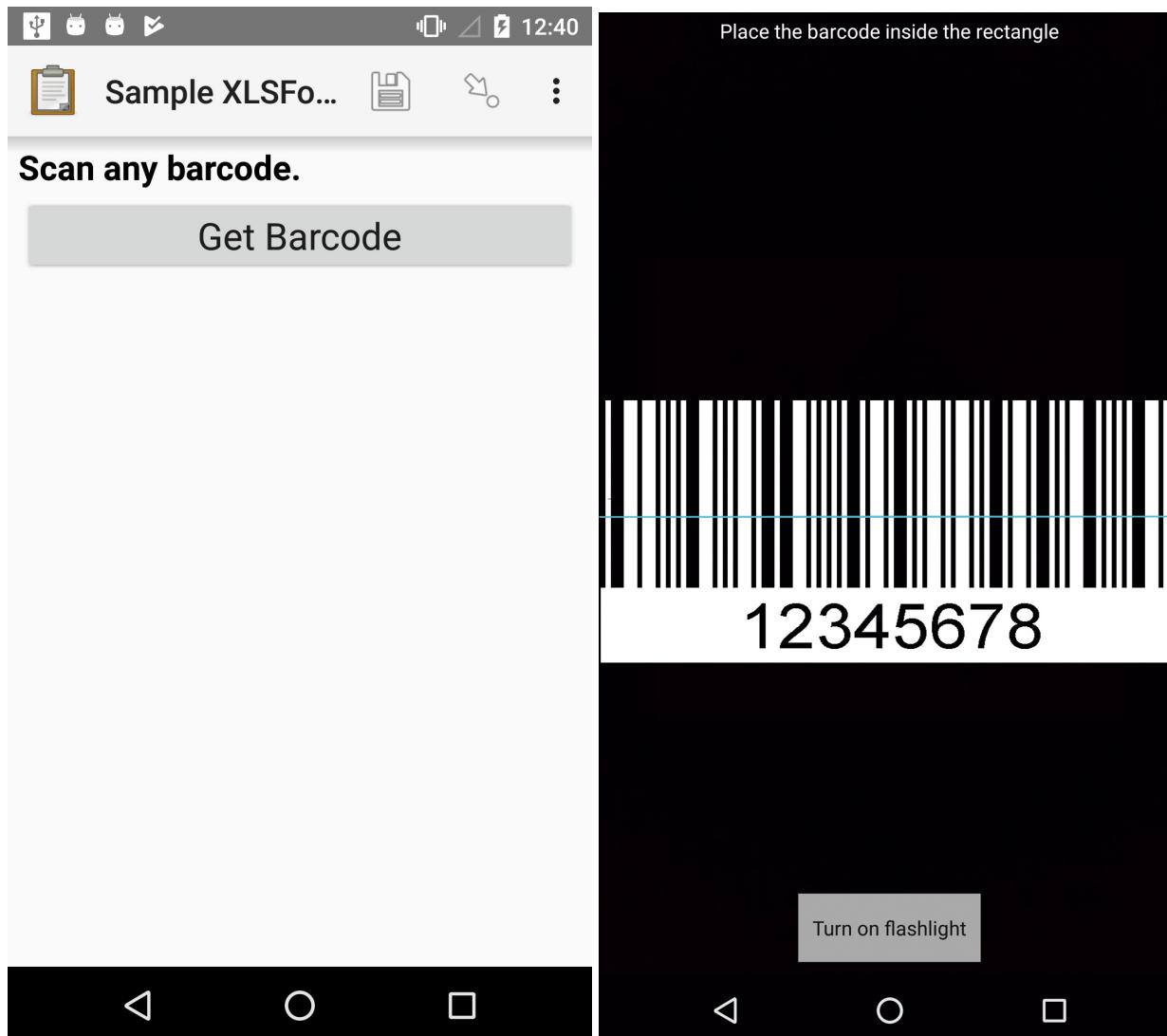
#### Note

Barcode scanning is built into Collect versions 1.7.0 and greater.

Versions of Collect prior to 1.7.0 require the [Barcode Scanner app](#) to be installed.

## Default barcode widget

The flash can be used as a light source when scanning barcodes in a poorly lit environment.





Sample XLSFo...

**Scan any barcode.****Replace Barcode**

671360013525



## XLSForm

survey		
type	name	label
barcode	barcode_example	Scan any barcode.

### ⚠ Warning

It is recommended not to make barcode questions required because even when using high quality and waterproof codes things can go wrong and some of them might be unreadable for the camera. To handle such cases, it might be a good idea to add a [Default text widget](#) as a fallback option to let enumerators enter the code manually.

## Self portrait (*selfie*) barcode widget

In some cases a front camera may work better. The flash can't be used in this case.

## XLSForm

survey			
type	name	label	appearance
barcode	barcode_example	Scan any barcode.	front

## Range widgets

Range widgets allow the user to select numbers from within a range that is visually represented as a number line. The parameters of the range widget are defined by `start`, `end`, and `step` values defined in the `parameters` column of your XLSForm. The parameter values can be integers or decimals.

## Default range widget with integers

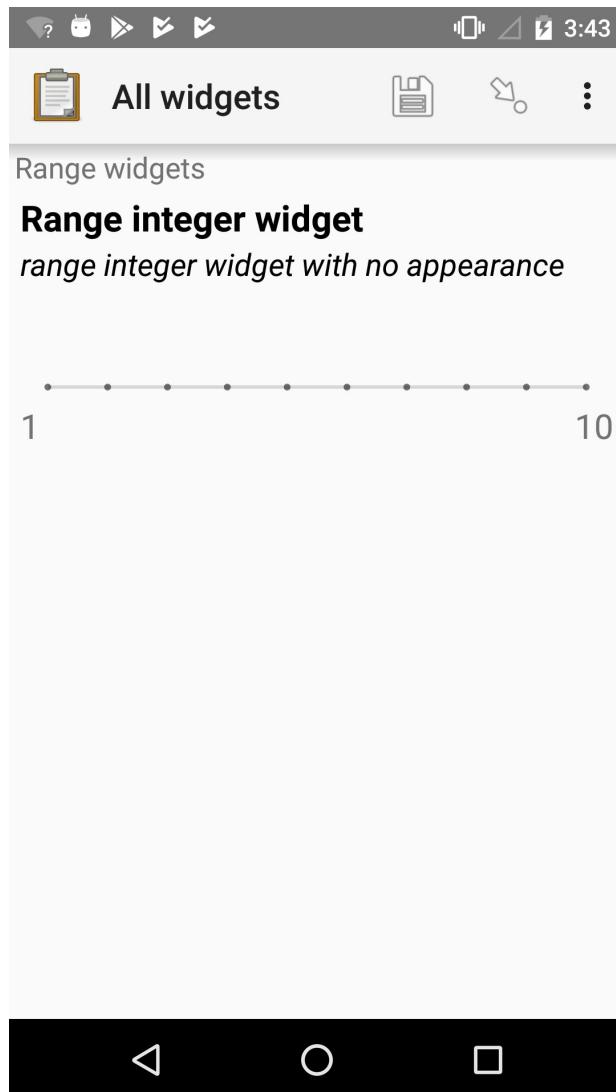
type

range

appearance

none

If all three parameter values are integers, the input will be stored as an integer.



## XLSForm

survey

type	name	label	appearance	hint	parameters
range	range_integer_widget	Range integer widget		range integer widget with no appearance	start=1;end=10;step=1

## Default range widget with decimals

type

range

appearance

none

If any of the parameter values are decimals, the input will be stored as a decimal.



All widgets



Range widgets

## Range decimal widget

*range decimal widget with no appearance*



## XLSForm

survey

type	name	label	appearance	hint	parameters
range	range_decimal_widget	Range decimal widget		range decimal widget with no appearance	start=1.5;end=5.5;step=0.5

## Vertical range widget

type

`range`

appearance

`vertical`

To display the range widget's number line vertically, use the `vertical` appearance. Both integers and decimals are supported.



All widgets



Range widgets

## Range vertical integer widget

*range integer widget with vertical appearance*



## XLSForm

survey					
type	name	label	appearance	hint	parameters
range	range_integer_widget_vertical	Range vertical integer widget	vertical	range integer widget with vertical appearance	start=1;end=10;step=1

## Range widget with picker

type

range

appearance

picker

When the `picker` appearance is added, the range widget is displayed with a spinner-style select menu in a dialog. The value between horizontal lines is the selected value. Users can scroll the spinner up and down or can tap on the value above to go up by one and on the value below to go down by one.



Range widgets

## Range picker integer widget

*range integer widget with picker appearance. This appearance can also be applied to a decimal range.*

Select value

No value selected

Range widgets

## Range picker integer widget

*range integer widget with picker*

*ap*

Number Picker

3

2

1

Cancel

OK

## XLSForm

survey						
type	name	label	appearance	hint	parameters	
range	range_integer_widget_picker	Range picker integer widget	picker	range integer widget with picker appearance	start=1;end=10;step=1	

## Range widget with rating

type

range

appearance

rating

When the `rating` appearance is added, the range widget is displayed with stars having equal spacing. Number of stars is calculated using the `end` parameter. When the user taps on an empty star, the stars up to and including that star will be filled. If the stars don't fit in the device width, they will wrap onto additional lines.



## Rating widget : 9

*rating widget appearance*



## XLSForm

survey					
type	name	label	appearance	hint	parameters
range	range_integer_widget_rating	Range rating widget	rating	range integer widget with rating appearance	end=9

## Note widget

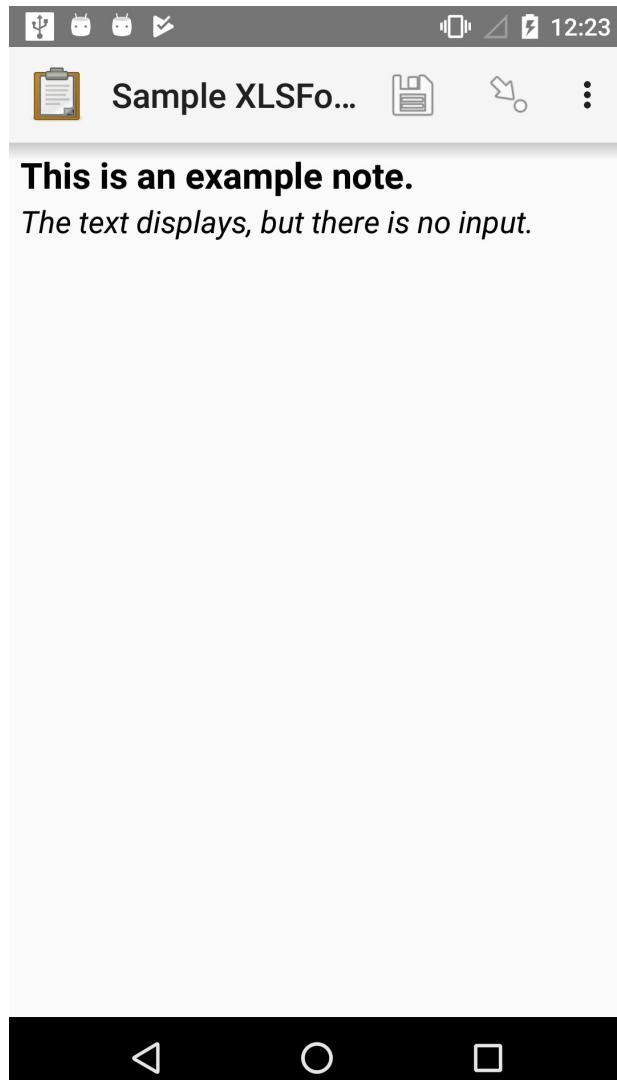
type

`note`

appearance

`none`

A note to the user, accepting no input. This example includes `hint` text.



## XLSForm

type	name	label	hint
note	note_1	This is an example note.	The text displays, but there is no input.

## URL widget

type

text

appearance

url

Provides a link which the user can open from the survey. Takes no input.

The URL to open is specified with `default`.



All widgets



Text widgets

## URL widget

*text type with url appearance and default value of <http://opendatakit.org/>*

Open Url

<http://opendatakit.org/>



## XLSForm

survey

type	name	label	appearance	hint	default
text	url_widget	URL widget	url	text type with url appearance and default value of <a href="http://getodk.org/">http://getodk.org/</a>	<a href="http://getodk.org/">http://getodk.org/</a>

## Printer widget

type

`text`

appearance

`printer:org.opendatakit.sensors.ZebraPrinter`

Connects to an external label printer, and prints labels that can contain a barcode, a QR code, or text.

See [Printing Labels with the Printer Widget](#) for complete details.



All widgets



Text widgets

## Ex printer widget

*text type with printer:org.opendatakit.sensors.ZebraPrinter*

Initiate Printing



## XLSForm

survey					
<b>type</b>	<b>name</b>	<b>label</b>	<b>appearance</b>	<b>calculation</b>	
text	ex_printer_widget	Ex printer widget	printer:org.opendatakit.sensors.ZebraPrinter	concat('123456789',' ','QR CODE',' ','Text')	

## Trigger/acknowledge widget

type

`trigger, acknowledge`

appearance

`none`

The trigger widget, also known as the acknowledge widget, presents a single checkbox.

A completed trigger response is stored as the string `ok`.

The example shown here includes the `required` attribute.



All widgets



## Trigger widget

Prompts for confirmation. Useful to combine with required or relevant.  
(type=trigger)

OK. Please continue.



All widgets



## Trigger widget

Prompts for confirmation. Useful to combine with required or relevant.  
(type=trigger)

OK. Please continue.

Sorry, this response is required!





All widgets



## Trigger widget

Prompts for confirmation. Useful to combine with required or relevant.  
(type=trigger)

OK. Please continue.



## XLSForm

survey

type	name	label	hint	required
trigger	my_trigger	Trigger widget	Prompts for confirmation. Useful to combine with required or relevant. (type=trigger)	true()

## Signature widget

type

image

appearance

signature

Collects a signature from the user.



2:01



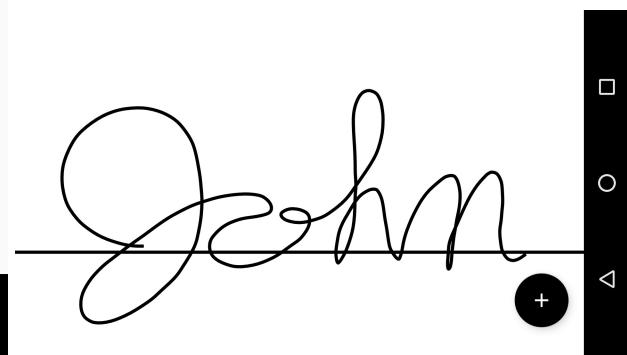
All widgets



Image widgets

**Signature widget***image type with signature appearance*

Gather Signature





All widgets



Image widgets

**Signature widget***image type with signature appearance*

Gather Signature

**XLSForm**

survey					
type	name	label	appearance	hint	
image	signature_widget	Signature widget	signature	image type with signature appearance	

**Grouping multiple widgets on the same screen**

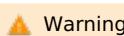
type

`begin_group`

appearance

`field-list`The `field-list` appearance attribute, applied to a group of widgets, displays them all on a single screen.

Relevance, constraint and calculation evaluation within the same screen is supported in Collect v1.22 and later.

Displaying [Repeating questions](#) on the same screen (inside a `field-list` group) is not supported.[Groups of questions](#) and [Repeating questions](#).**Grid of selects on the same screen**

If you have multiple select questions with the same choices, it can be helpful to group them on one screen.

The screenshot shows a mobile application interface for a survey. At the top, there's a header bar with icons for back, forward, and menu. Below the header, the title "Conditions" is displayed. Underneath the title, the section "Underlying conditions" is shown. There are two columns: "Conditions" and "Yes" (with a radio button) and "No" (with a radio button). Below this, five rows of questions are listed, each with a red asterisk indicating required input:

Conditions	Yes	No
* Pregnancy	<input type="radio"/>	<input type="radio"/>
* Post-partum (< 6 weeks)	<input type="radio"/>	<input type="radio"/>
* Immunodeficiency, including HIV	<input type="radio"/>	<input type="radio"/>
* Cardiovascular disease, including hypertension	<input type="radio"/>	<input type="radio"/>
* Diabetes	<input type="radio"/>	<input type="radio"/>

To do this, put your select questions in a `field-list` group and use the following `appearance` attributes:

#### `label`

Only the option labels are displayed, without checkboxes. This is used for the top row with the 'Yes' and 'No' options in the example above.

#### `list-nolabel`

Only checkboxes or radio buttons are displayed, without their labels. This is used for the question rows in the example above.

#### `list`

The labels are displayed along with checkboxes for multi-select questions and radio buttons for single-select questions. You could use this instead of having a `label` row to keep the option labels closer to the checkboxes or radio buttons.

## XLSForm

survey			
<code>type</code>	<code>name</code>	<code>label</code>	<code>appearance</code>
begin_group	underlying_conditions	Underlying conditions	field-list
select_one yes_no	condition_labels	Conditions	label
select_one yes_no	Comcond_preg	Pregnancy	list-nolabel
select_one yes_no	Comcond_partum	Post-partum (< 6 weeks)	list-nolabel
end_group	underlying_conditions		

choices		
<code>list_name</code>	<code>name</code>	<code>label</code>
yes_no	yes	Yes
yes_no	no	No

## Hidden questions

Not all question types render as visible widgets in Collect. Hidden fields collect and store values which are accessible as [variables](#) and available in [Central](#) and other data analysis tools.

## Metadata

Metadata questions capture information about the device or a survey collection event and are not visible to the user.

### ⚠ Warning

`start`, `end`, and `today` rely on device time. Depending on the device hardware, operating system, clock configuration, network configuration, the device time can be wrong. To reliably measure the time between events (e.g., to know how long a survey took) or to get a more complete record of user behavior within a form, consider using [form audit logging](#).

These items are dependent on the survey collection event:

- `start` — The datetime the survey was started in ISO 8601 format (e.g., 2019-09-27T09:45:10.854-07:00).
- `end` — The last datetime the survey was saved in ISO 8601 format.
- `today` — The date the survey was started in ISO 8601 format (e.g, 2019-09-27).
- `start-geopoint` — The geolocation when the survey was started. [Read more](#).

This item is defined at installation time and cannot be changed:

- `deviceid`

These items are defined in Collect, and [can be edited in Settings](#):

- `username`
- `phonenumbers`

## XLSForm

survey	
type	name
start	start
end	end
today	today
deviceid	deviceid
username	username
phonenumbers	phonenumbers
start-geopoint	start-geopoint

## Geolocation at survey start

### See also

[Audit log geolocation tracking](#)

### Note

Geolocation at survey start was added in Collect v1.23 and Central v1.0.0.

The `start-geopoint` question type is used to capture a single geolocation in [geopoint format](#) when the survey is first started. Questions of type `start-geopoint` may be given any allowable name. Although it is possible to have more than one `start-geopoint` question in a form, all will have the same value.

Any time a survey with a `start-geopoint` question is opened in Collect, the enumerator will see a warning that the form tracks device location. If the device battery is low, or if location tracking needs to be turned off for any reason, you can tap : ▶ Track location or turn off location providers in Android.

The first time that a survey with a `start-geopoint` question is opened, Collect will attempt to read the device's geolocation. The geolocation reading with the highest accuracy received in a 20-second window will be recorded. A location icon will be displayed in the Android status bar while the geolocation is being requested by Collect.

Geolocation is read using data from GPS, WiFi and possibly other signals so this feature should work in most environments.

If geolocation information is unavailable, the question will be left blank. Reasons for a blank value may include the enumerator turning off location providers, Collect not having location permissions, Google Play Services not being installed, the GPS not having satellite lock and more. No troubleshooting information is provided in the form submission.

Including a `start-geopoint` question may make it faster to get high-accuracy geolocation readings for other [location question types](#) by "warming" the GPS.

## Calculate

type

calculate

Calculate questions let you evaluate complex [expressions](#), storing the values for later use.

For more details, see [Calculations](#).

## Background audio recording

type

background-audio

New in version 1.30: [ODK Collect v1.30.0](#), Central v1.2.0

 See also

[Logging enumerator behavior](#), [audio questions](#)

### XLSForm

survey	
type	name
background-audio	my_recording

When a form includes a question of type `background-audio`, audio is recorded while the form is open and attached to the form submission as a single audio file. These recordings can be used for quality assurance, training, transcription, and more. Use background recording instead of an [audio question](#) when you want to make sure to record everything that happens during form filling.

By default, audio files will be saved in the `amr` format with a bitrate of 12.2kbps and a sample rate of 8kHz, resulting in a file size of about 5MB per hour. These settings correspond to the `voice-only` quality [for audio questions](#) and minimize file size while maintaining reasonable quality for a conversation between two people. You can override that default quality by specifying a value in the `parameters` column as described [for audio questions](#).

### XLSForm

survey		
type	name	parameters
background-audio	my_recording	quality=low

## Planning for background audio recording

Before adding background audio recording to your form, make sure that you have a plan for following local laws around audio recording. We generally recommend including a note at the beginning of the form to remind data collectors and participants that they are being recorded and to describe the purpose of the recording. Depending on the context, you may be required to ask for the consent of every person in speaking range of the microphone.

Additionally, you should make a plan for using the resulting audio files. Do you have someone who can listen to and make sense of many audio files? Could you get access to speech-to-text capabilities? Also consider whether your data collectors will be able to send audio files. Will they have access to a fast enough Internet connection? Will their Internet plan or number of credits allow them to send all the audio files you expect?

It can be helpful to combine background audio recording with [audit logging](#) to have more context while listening to the recording. Recording starts at the `form start` and `form resume` events and stops at the `form exit` event. You can use the difference between the `form start` time and the start time of an event to identify how far into the background recording a certain event happened.

## Background audio recording user interface

While recording is ongoing, an audio status bar is shown at the top of the screen. This bar helps remind data collectors that they are being recorded and provides visual feedback about audio volume.

If a data collector exits a form and then re-opens it for editing, audio recording is resumed. Audio recording continues as long as the form is open, even if another application is in the foreground or the screen is locked. Note that Collect settings can't be accessed directly from a form that has background recording. The audio file sent to the server will

include audio from every time that the form was opened for editing.

### Tip

Android devices can make many sounds during use and these will be included in recordings. We recommend turning off sounds from button presses, camera shutters and notifications before recording.

There is an override available to data collectors when recording might compromise safety or when consent to record can't be obtained but it's still important to capture some data. When the audio recording option is unchecked, audio recording ends immediately and any previously-recorded audio is deleted. Recording can't be resumed during the current form-filling session. Toggling the audio recording option back on indicates that the next form filling session should be recorded. If [audit logging](#) is enabled, a `background audio disabled` event will be logged if a data collector toggles off recording and a `background audio enabled` event will be logged if a data collector toggles it back on.

## Background audio recording troubleshooting

In some rare cases such as the device running out of space, the recording may complete successfully but not be attached to the form. If this happens, the recording may be available in the `recordings` folder of the [Collect directory](#). This folder is never cleared so consider emptying it yourself once you have retrieved its files.