

# CyberTracker XlsForm Reference Manual

## Purpose

The purpose of this training resource is to provide a guide on how to extend XlsForm to make use of CyberTracker features.

## Acknowledgements

This training resource leverages technology provided by ODK, KoBoToolbox, Survey123 and others. It is built on the popular XlsForm specification.

## Overview

This training resource covers the modification of XlsForm files to activate CyberTracker functionality.

## TABLE OF CONTENTS

- [Initial setup](#)
  - [namespace \(required\)](#)
  - [version \(recommended\)](#)
- [Settings](#)
  - [immersive](#)
  - [wizardMode](#)
  - [colors](#)
  - [esriLocationServiceUrl](#)
  - [sendLocationInterval](#)
- [Header](#)
  - [text](#)
  - [topText](#)
  - [button](#)
  - [hidden](#)
  - [qml](#)
  - [qmlBase64](#)
  - [qmlFile](#)
- [Content](#)
  - [frameWidth](#)
  - [style](#)
  - [padding](#)
  - [columns](#)
  - [lines](#)
  - [border](#)
  - [borderWidth](#)
  - [fontSize](#)
  - [fontBold](#)
  - [itemHeight](#)
  - [qml](#)

- [qmlBase64](#)
- [qmlFile](#)
- [Footer](#)
  - [buttons](#)
  - [back button](#)
  - [next button](#)
  - [save button](#)
  - [nextOrSave button](#)
  - [index button](#)
  - [options button](#)
  - [map button](#)
  - [hidden](#)
  - [qml](#)
  - [qmlBase64](#)
  - [qmlFile](#)
- [Save](#)
  - [snapLocation](#)
  - [targets](#)
  - [track](#)
- [Miscellaneous](#)
  - [fixCount](#)
  - [trackFileFormat](#)
- [Frequently Asked Questions](#)
  - [Which backends support XmlForm?](#)
  - [Are CyberTracker extensions visible to other tools?](#)

## Initial setup

These columns on the `settings` sheet are needed to begin using the CyberTracker extensions.

### namespace (required)

The `ct` namespace tells other XlsForm tools to ignore columns starting with `bind::ct:`.

title		version	namespaces	
My Form		2022101001	ct="http://cybertracker.org/xforms"	
◀ ▶	survey	choices	<b>settings</b>	○

### version (recommended)

The version field is used to track form versions as they change over time. While it is not required, it is a best practice to keep this field up to date. The [XlsForm specification](#) recommends the convention like 'yyyymmddrr'. For example, 2022021501 is the 1st revision from Feb 15th, 2022.

## Settings

immersive

Setting this to `true` causes the form UI to use the wizard exclusively, i.e. this is more like CyberTracker Classic. Default is `false`. `TRUE` and `FALSE` values also work.

title		bind::ct:immersive	
My Form		true	
◀ ▶	survey	choices	<b>settings</b>
			<input type="radio"/>

In the flow below, the user context is always within a sighting and each page typically holds one question. Pressing the **options** button navigates to a new page which shows the content of the current sighting on one tab and all sightings on the other. The user can edit previous sightings, but when editing is complete, the wizard will revert to the original sighting.

The image displays three mobile application screens side-by-side. The first screen, titled 'Question 1', features a list of animals: Rhino, Lion (highlighted), Buffalo, Leopard, Elephant, and Giraffe. The second screen, titled 'Current sighting', shows a list of questions: Introduction, Question 1 (highlighted), Question 2, Question 3, Question 4, Question 5, Question 6, Question 7, Question 8, and Question 9. The third screen, titled 'Saved sightings', lists two sightings: 'Lion' (dated Tue 2022/12/13 19:19:56) and 'Leopard' (dated Tue 2022/12/13 19:20:10), both marked with checkmarks. At the bottom of this screen is a red 'Submit' button. Each screen has a dark blue header with navigation icons (back, home, forward, settings) and a light gray footer with additional navigation icons (back, home, forward).

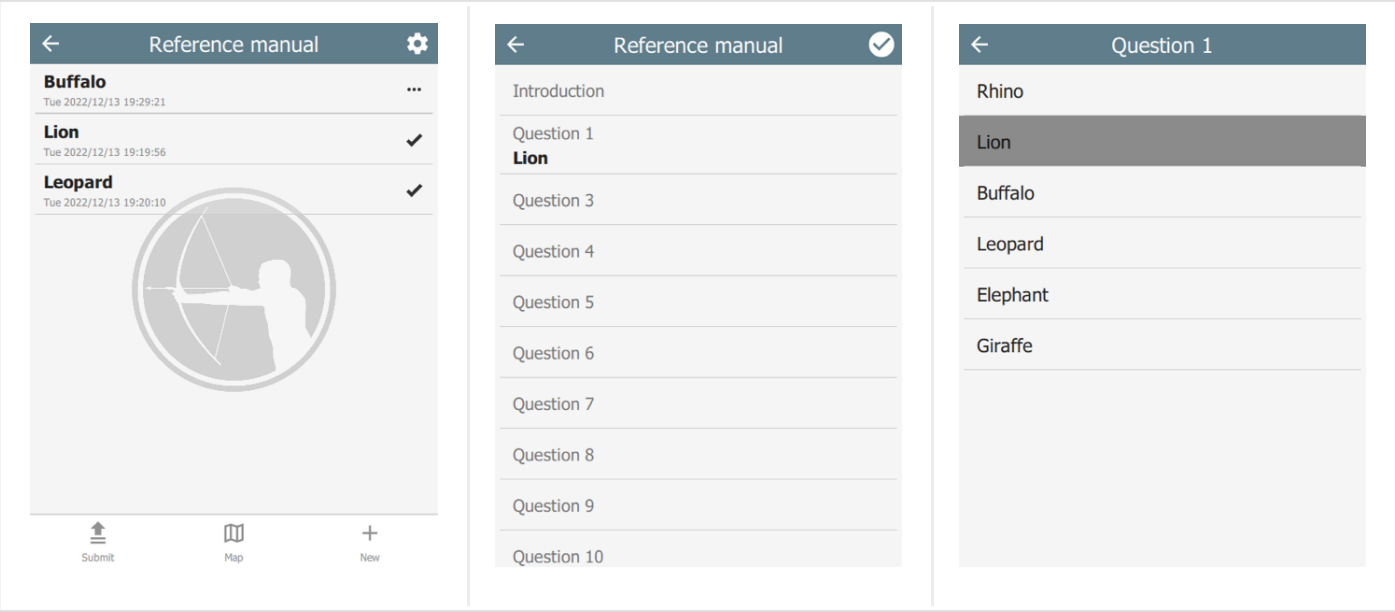
## wizardMode

If the `immersive` column is missing or set to `false`, then the UI reverts to *non-immersive* mode. In this case, there is a **Home** page which shows all sightings. The user returns to this screen after saving a sighting.

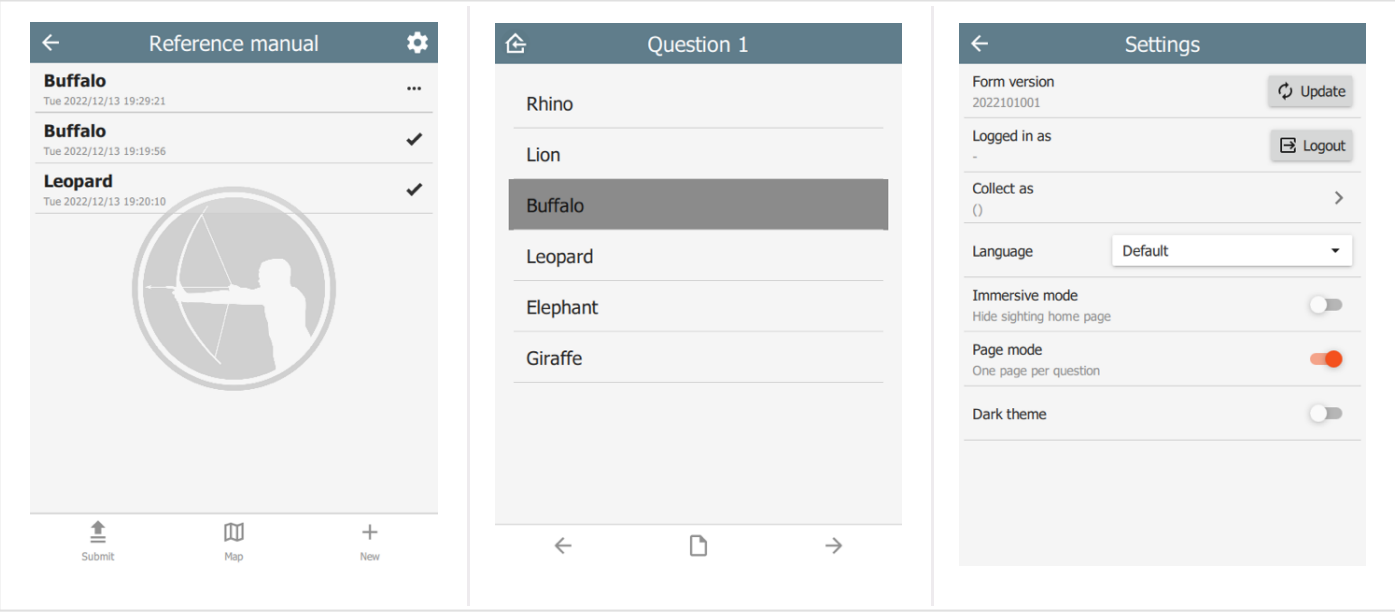
--	--	--

title	bind::ct:immersive	bind::ct:wizardMode
My Form	false	true
◀ ▶	survey	choices <b>settings</b> ○

If `wizardMode` is set to `false`, then all questions show on a single page. This mode is recommended when wanting to show all sighting data at once and is the most conventional.

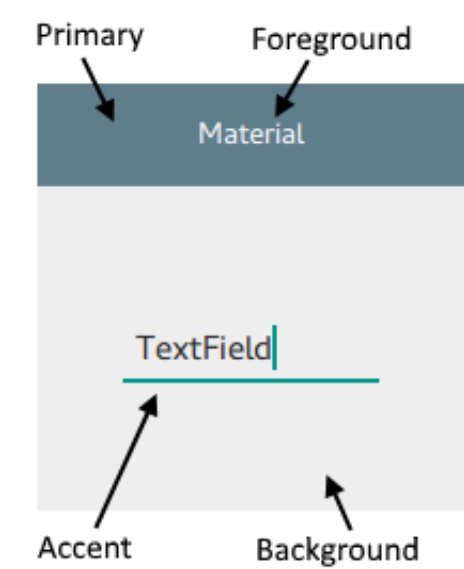


If `wizardMode` is set to `true`, then each question will appear on its own page with **Back** and **Next** toolbar buttons to navigate between questions. The user will still return to the **Home** page between sightings. `wizardMode` appears on the **Settings** page as **Page mode**.



# colors

The colors attribute specifies the color scheme used on the form. It follows the [Material Design](#) system:



title	bind::ct:colors.primary	bind::ct:colors.accent
My Form	#6200EE	green
◀ ▶	survey	choices <b>settings</b> ○

The following color fields are supported:

- primary
- accent
- foreground
- background
- darkPrimary
- darkAccent
- darkForeground
- darkBackground

Colors prefixed with 'dark' will be used when dark mode is activated. If dark colors are not provided, then 'primary' and 'accent' colors will be used, but 'foreground' and 'background' colors will be ignored.

Colors can also be provided as a JSON object:

title	bind::ct:colors			
My Form	{ "primary": "#6200EE", "accent": "green" }			
◀ ▶	survey	choices	<b>settings</b>	○

## esriLocationServiceUrl

When using Survey123, CyberTracker supports uploading locations and tracks to a hosted feature service. In this case, the feature service is specified in the esriLocationServiceUrl column:

title	bind::ct:esriLocationServiceUrl			
My Form	https://services6.arcgis.com/.../FeatureServer			
◀ ▶	survey	choices	<b>settings</b>	○

The feature service contains three layers: Tracks (point layer), Last Known Locations (point layer) and Track Lines (Polyline layer).

If this service is not specified, then tracks are placed in a file type question of the sighting which specifies snapTrack in the save section of the survey question.

## sendLocationInterval

CyberTracker can send the current location at regular intervals, separately from a the track system. This value is in seconds and is user configurable - this specifies the default value. For Survey123, esriLocationServiceUrl must be configured.

title	bind::ct:sendLocationInterval			
My Form	30			
◀ ▶	survey	choices	<b>settings</b>	○




# Header

The header object supports custom header attributes. If no header object is specified, then the default header is used. By default, the header title is taken from the question label.

## text

Custom header text.

type	name	bind::ct:header.text
select_one animal	Animal	Custom question text
◀ ▶	<b>survey</b>	choices   survey   ○

 Custom question text

Rhino

Lion


Buffalo

Leopard

Elephant

Giraffe

←




→

## topText

Custom smaller text above main title.

type	name	bind::ct:header.topText
select_one animal	animal	Custom top text
◀ ▶	<b>survey</b>	choices   survey   ○



Custom top text  
Question 1

Rhino

---

Lion

---

Buffalo

---

Leopard




---

Elephant

---

Giraffe

---

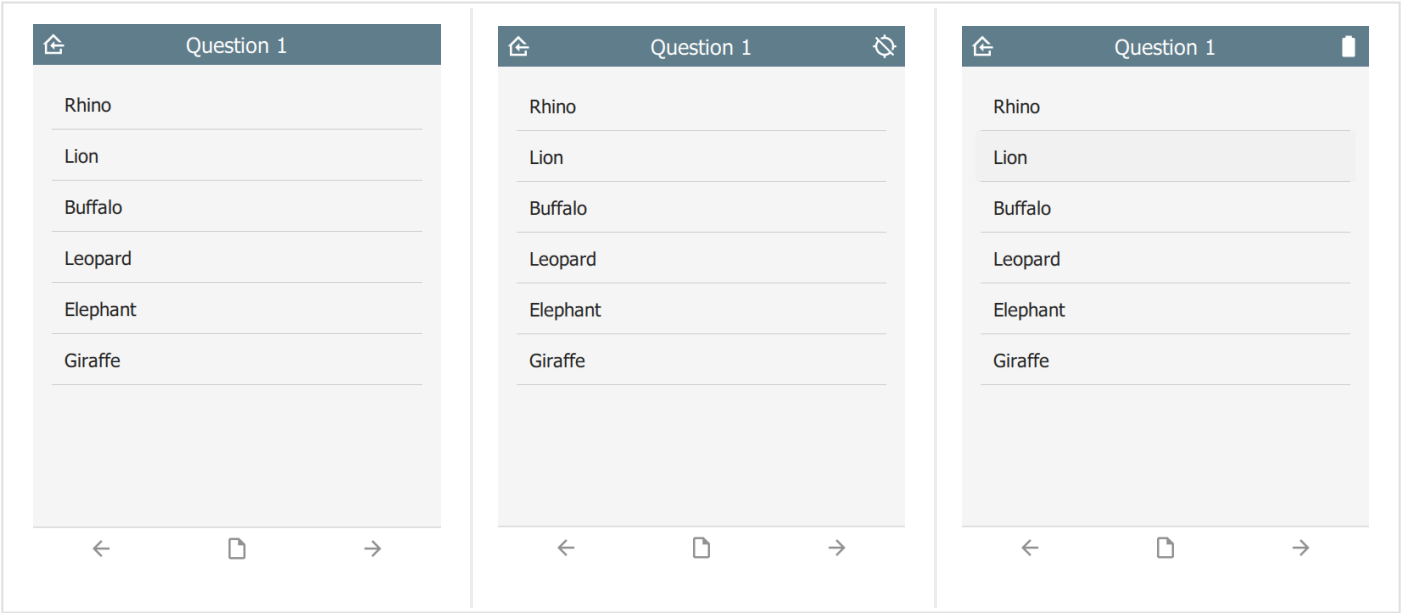
## button

Type of the button in the top-right corner. Valid values are:

- empty - by default no button is shown
- `track` - the current state of the GPS track system
- `battery` - the current state and level of the battery

type	name	bind::ct:header.button
select_one animal	animal	track
◀ ▶	<b>survey</b>	choices   survey   ○

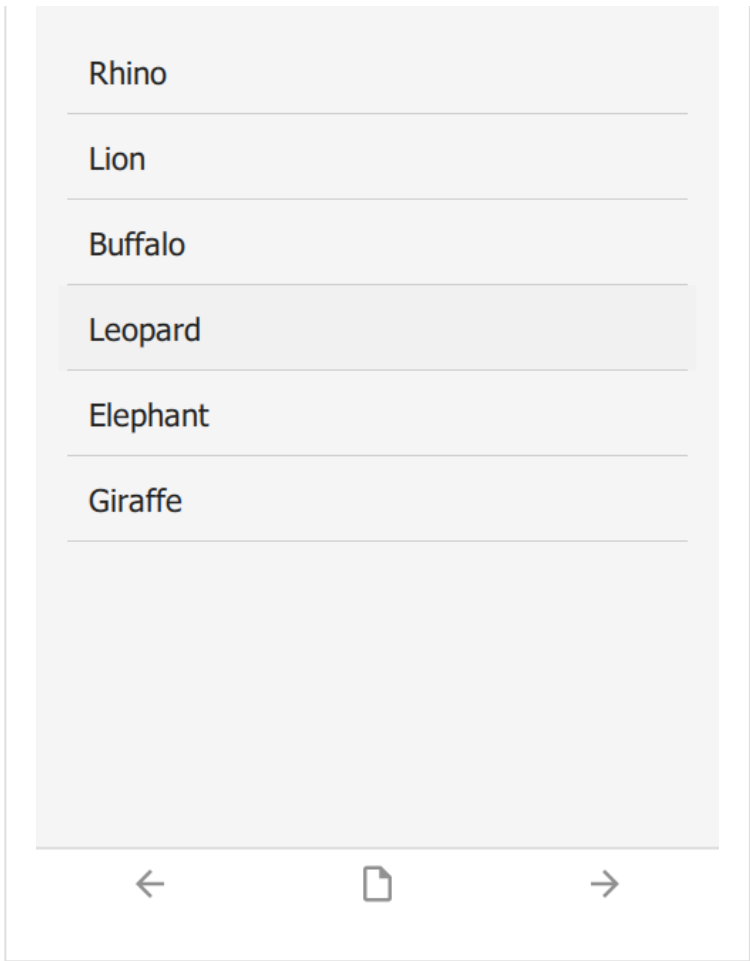
Tapping on the button will provide more information, e.g. the track frequency or the battery level.



hidden

If `true` then the header is hidden. Default is `false`.

type	name	bind::ct:header.hidden
select_one animal	animal	false
◀ ▶	<u>survey</u>	choices   survey   ○



qml

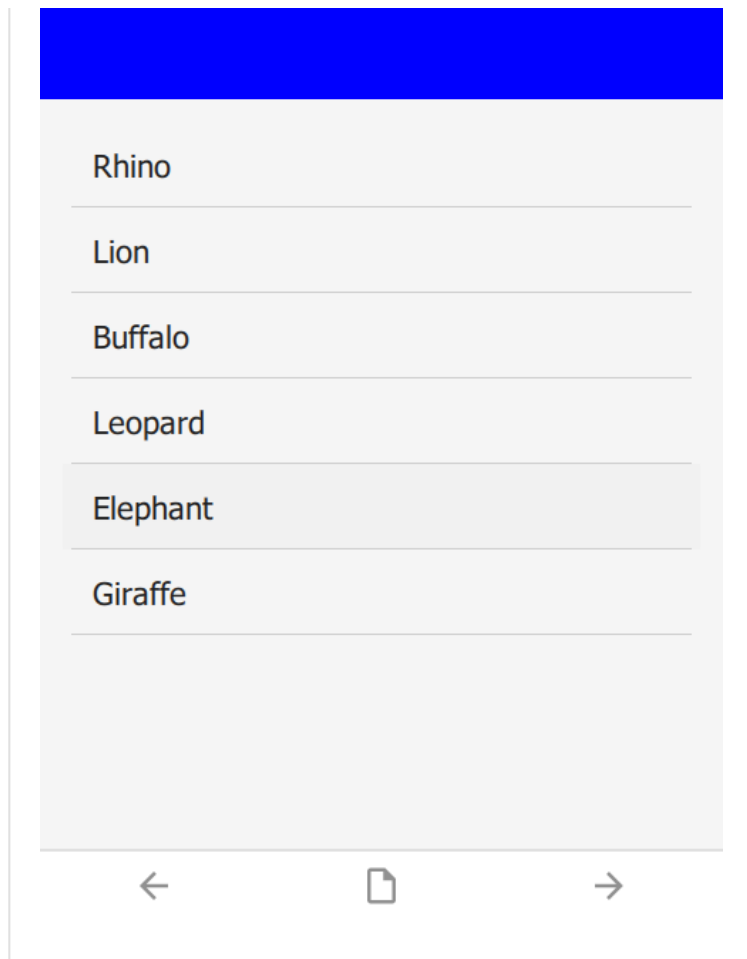
A [QML](#) fragment to use instead of the built-in header. See Developer section. For example:

type		name		bind::ct:header.qml	
select_one animal		animal		qml fragment	
◀ ▶	<b>survey</b>	choices	survey	<input type="radio"/>	

To set the header to a blue rectangle, replace `qml fragment` above with the following:

```
import QtQuick 2.15

Rectangle {
    color: "blue"
    height: 64
}
```



## qmlBase64

Base64 encoded QML (see **qml** above).

## qmlFile

Name of a QML file which exists alongside other project files. This is not supported on ODK or KoBoToolbox, but can be used in Survey123.

# Content

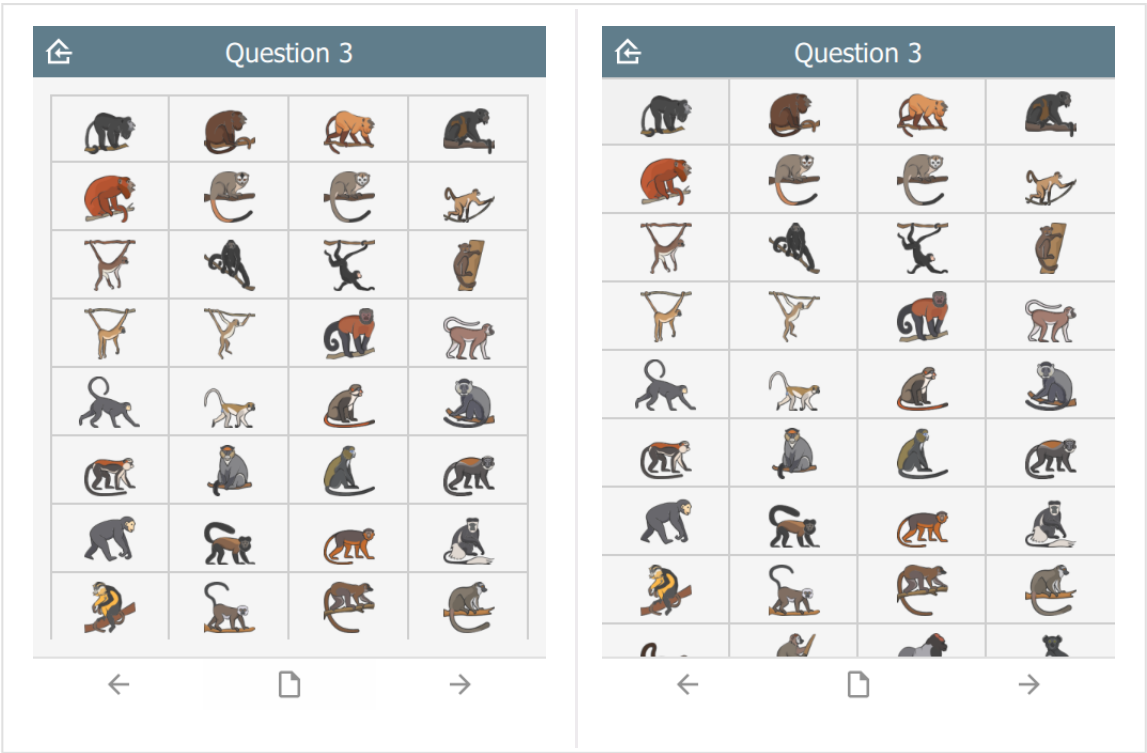
The content section is the middle part of the screen between the header and footer. By default it automatically selects a control for the question type, e.g. a date selector for a date question. By specifying a custom content object, more styles are available. This is especially useful for customizing lists.

## frameWidth

Frame width around the content area of the page. Default is 16.

type	name	bind::ct:content.frameWidth
select_one animal	animal	0
<div><div><div>◀▶</div><div>survey</div><div>choices</div><div>survey</div><div><div></div></div></div></div>		

In this case, `frameWidth` was set to `0` in the second image.



## style

The visual appearance of the question.

type	name	bind::ct:content.style
------	------	------------------------


select_one animal	animal	IconOnly
◀ ▶	<b>survey</b>	choices   survey   ○


For **select\_one** questions


- (not specified)
- IconOnly
- TextOnly
- TextBesideIcon
- TextUnderIcon


Single select  
Default view


Search


 Alouatta caraya


 Alouatta guariba

 Alouatta macconelli

 Alouatta palliata

 Alouatta seniculus

 Aotus azarae





 Aotus vociferans





◀





📄





▶





Single select  
IconOnly style





   





   





   





   





   





   





   

◀

📄

▶

Single select  
TextOnly style

Alouatta caraya

Alouatta guariba

Alouatta macconelli

Alouatta palliata

Alouatta seniculus

Aotus azarae

Aotus vociferans

Ateles geoffroyi

Ateles hybridus brunneus

Ateles paniscus 2

...


◀

📄


▶

Single select  
TextBesideIcon style


Alouatta caraya




Alouatta guariba




Alouatta macc...




Alouatta palliata




Alouatta senic...




Aotus azarae




Aotus vociferans




Ateles geoffroyi




Ateles hybridu...




Ateles paniscu...




Ateles paniscus




Avahi laniger




Brachyteles ar...




Brachyteles ar...




Cebus apella




Cercocebus agi...




Cercopithecid...




Cercopithecus ...



Cercopithecus ...



Cercopithecus ...



◀

📄

▶

Single select  
TextUnderIcon style

 Alouatta car...

 Alouatta gua...

 Alouatta ma...

 Alouatta pall...

 Alouatta sen...

 Aotus azarae

 Aotus vocifer...

 Ateles geoffr...

 Ateles hybrid...

 Ateles panis...

 Ateles panis...

 Avahi laniger

 Brachyteles ...

 Brachyteles ...

 Cebus apella

◀

📄


▶


For **select\_multiple** questions:


- (not specified)
- IconInlay
- IconOnly
- TextOnly
- TextBesideIcon


Multiple select  
Default view


Search


 Afropavo congensis ☒

 Alectrurus tricolor ☐


 Amazona amazonica ☐

 Amazona autumnalis ☒


 Amazona vinacea ☐

 Anthracoceros albirostris ☐

Multiple select  
IconInlay style



Multiple select  
IconOnly style



Multiple select  
TextOnly style

☒ Afropavo congensis

☐ Alectrurus tricolor

☐ Amazona amazonica

☐ Amazona autumnalis

☒ Amazona vinacea

☐ Anthracoceros albirostris

☐ Aratinga solstitialis

☒ Ara ararauna


☐ Ardea humbloti

☐ Bangsia aureocincta


☐ ...

Multiple select  
TextBesideIcon style


☒ Afropavo congensis




☐ Alectrurus tricolor




☐ Amazona amazonica




☒ Amazona autumnalis




☐ Amazona vinacea




☒ Anthracoceros albirostris



☐ Aratinga solstitialis



☐ Ara ararauna



For **range** questions:

type	name	parameters	bind::ct:content.style	bind::ct:content.c
range	animal_count	start=1 end=50 step=1	Grid	5
<div><div><div>◀▶</div><div><div>survey</div>choices</div>survey</div><div><input type="radio"/></div></div>				



Range

Grid style, 5 columns

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65

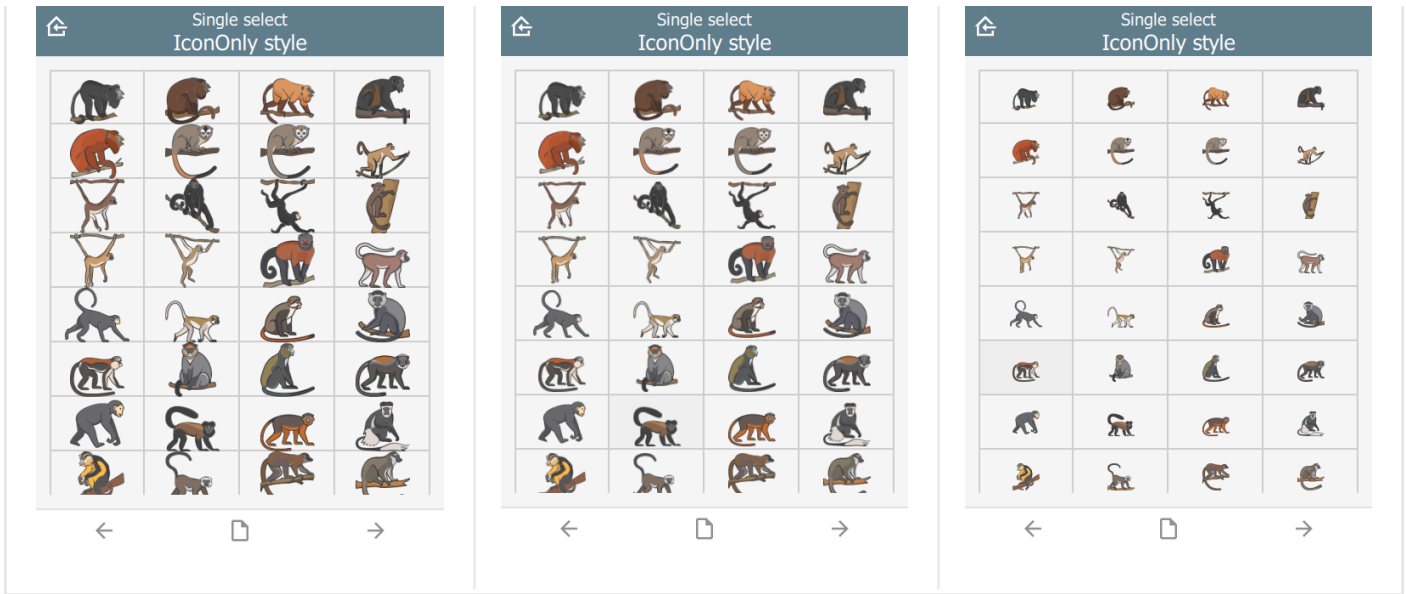
Ignored for other question types.

padding

The padding space between grid items. Requires `style` to be set.

type	name	bind::ct:content.padding
select_one animal	animal	8
<div><div><div></div><div></div></div><div><div><div>survey</div></div><div>choices</div><div>survey</div><div><div></div></div></div></div>		

Padding values are `0`, `4` and `8`.

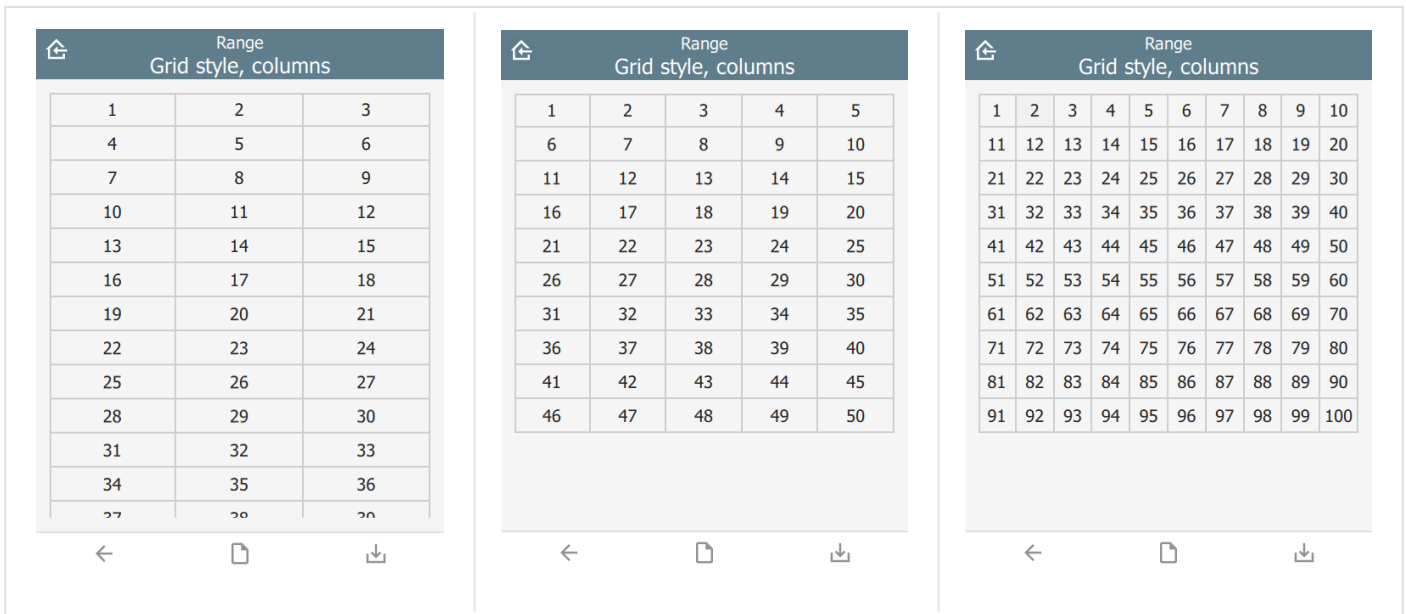


## columns

Number of columns for grids. Requires `style` to be set. Defaults to 2.

type	name	bind::ct:content.columns
select_one animal	animal	4
◀ ▶	<b>survey</b>	choices   survey   <input type="radio"/>

Column values are `3`, `5` and `10`.

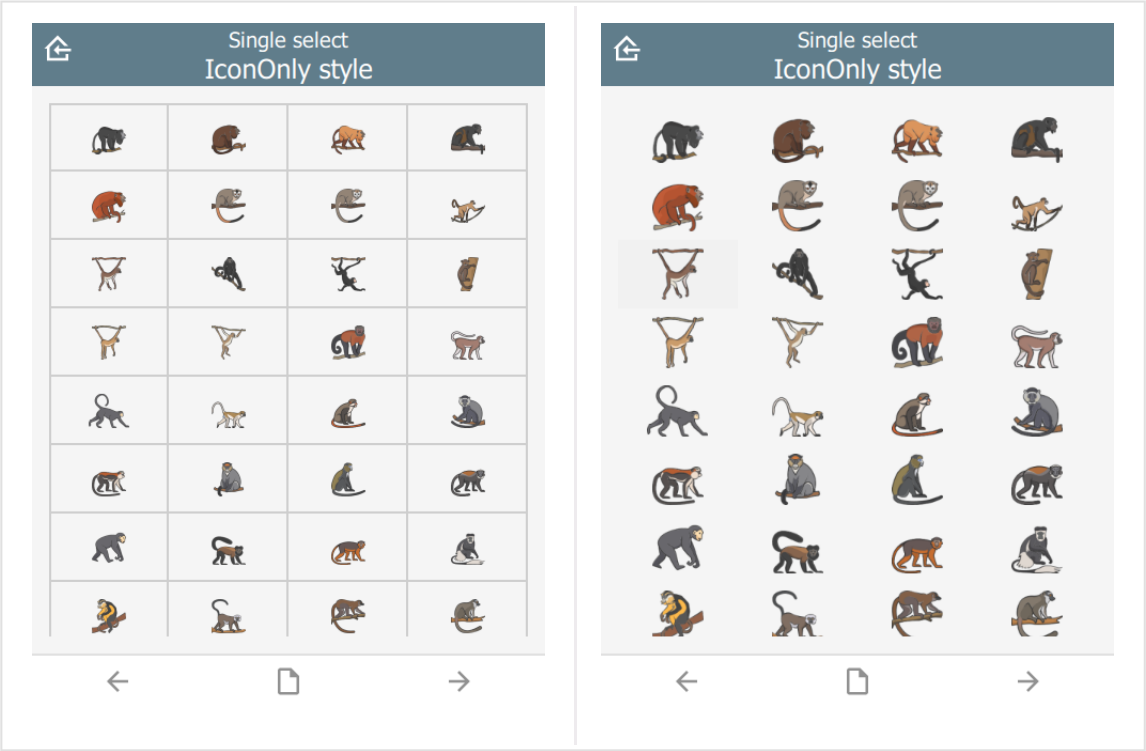


## lines

Show lines between cells for grids. Requires `style` to be set. Defaults to true.

type	name	bind::ct:content.lines
select_one animal	animal	false
◀ ▶	<b>survey</b>	choices   survey   ○

Lines value is `true` and `false`.



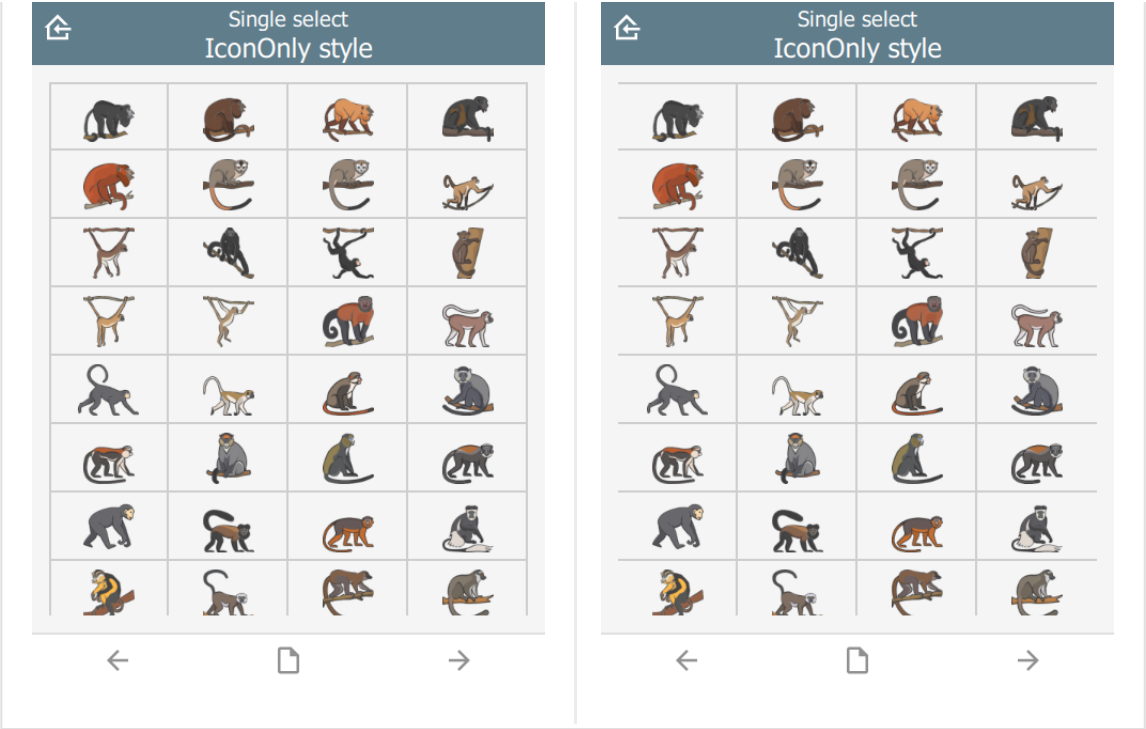
border

Show border around the outside of a grid. Requires `style` to be set. Defaults to `false` if `frameWidth` is 0, `true` otherwise.

type	name	bind::ct:content.border
select_one animal	animal	true
◀ ▶	<b>survey</b>	choices   survey   ○

Border value is `true` and `false`.



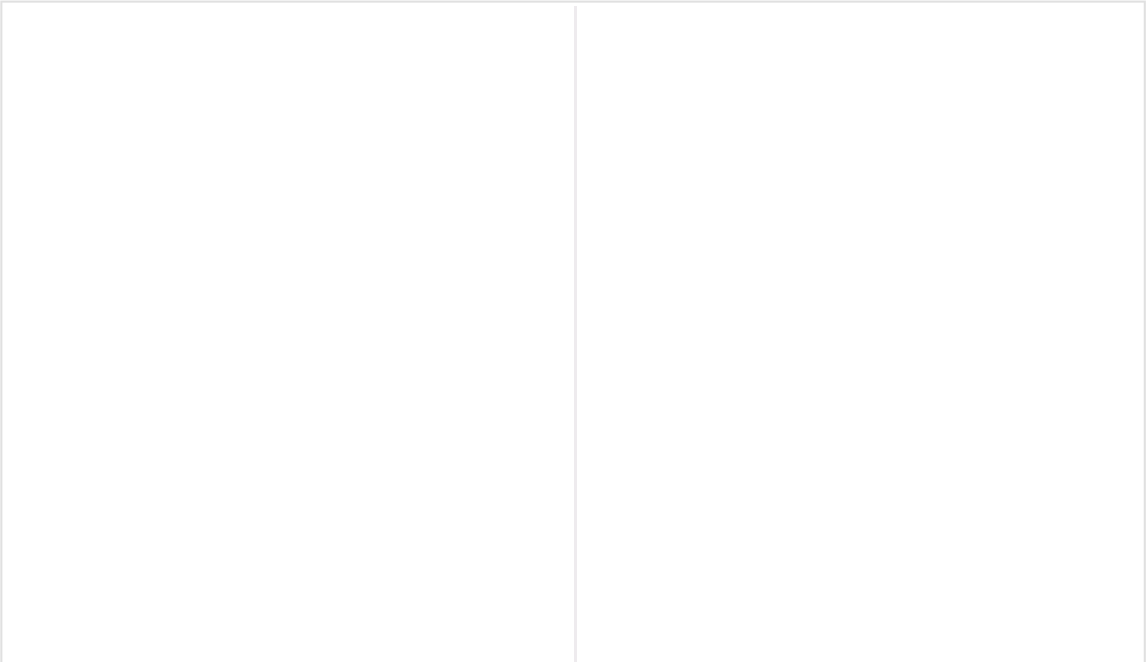


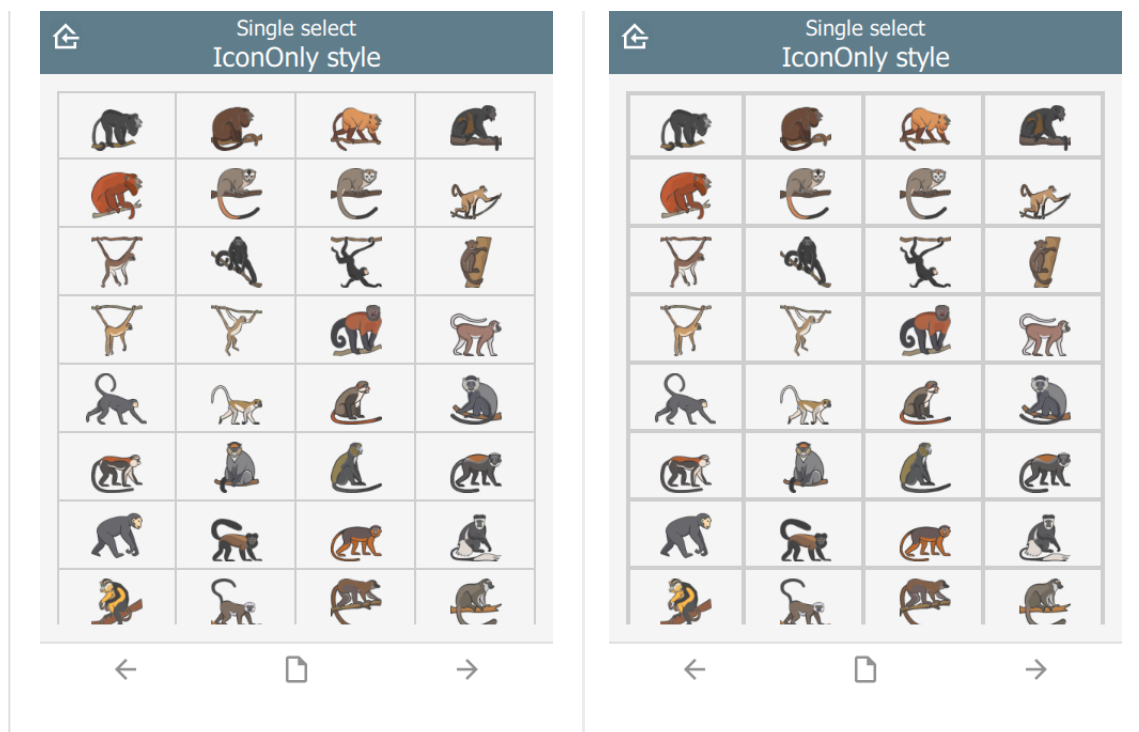
borderWidth

Border width for grid lines. Requires `style` to be set. Defaults to 2.

type	name	bind::ct:content.borderWidth
select_one animal	animal	2
◀ ▶	<b>survey</b>	choices   survey   ○

borderWidth value is 2 and 4.





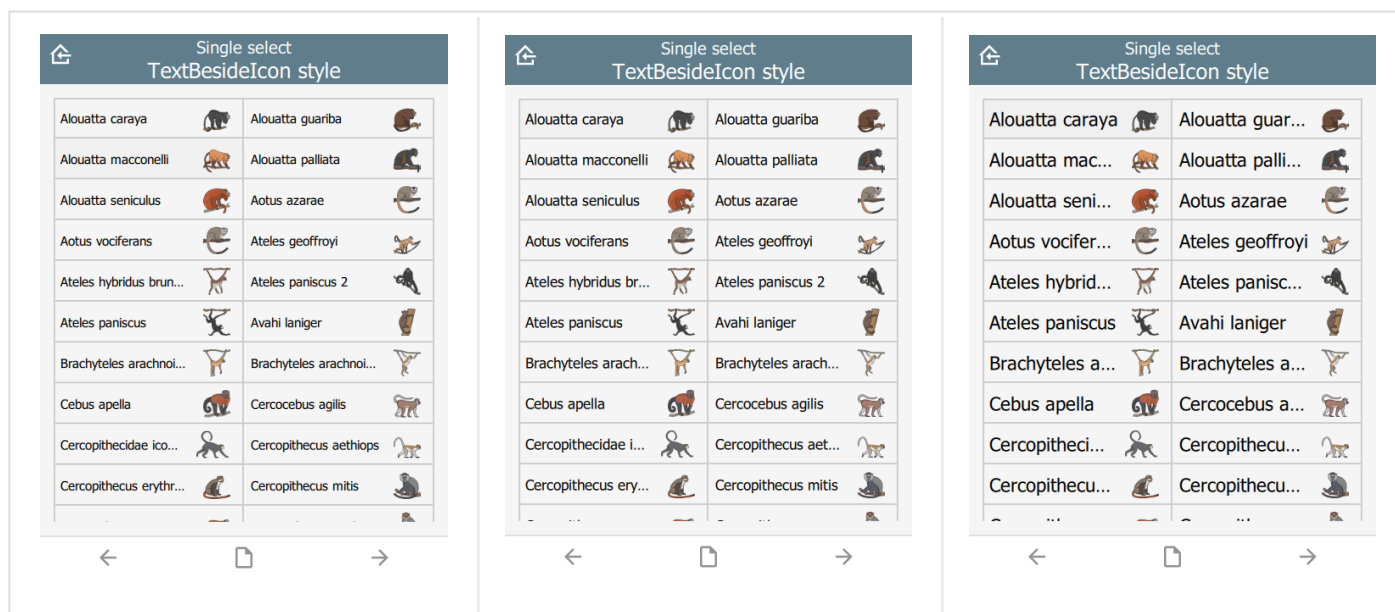
## fontSize

Size text font size. Requires `style` to be set. Defaults to 16.

type	name	bind::ct:content.fontSize
select_one animal	animal	14

◀ ▶
**survey**
choices
survey
○

`fontSize` values are 12, 14 and 16.



fontBold

Set font to bold. Requires `style` to be set. Defaults to false.

type	name	bind::ct:content.fontBold
select_one animal	animal	true
<div><div>◀▶</div><div>survey</div><div>choices</div><div>survey</div><div><input type="radio"/></div></div>		

itemHeight

Set height of individual items. Requires `style` to be set. Defaults to 48.

type	name	bind::ct:content.itemHeight
select_one animal	animal	48
<div><div>◀▶</div><div>survey</div><div>choices</div><div>survey</div><div><input type="radio"/></div></div>		

`itemHeight` values are `48`, `64` and `128`.

Single select  
Custom frame

◀

📄

→

Single select  
Custom frame

◀

📄

→

Single select  
Custom frame

◀

📄

→

qml

A **QML** fragment to use instead of the built-in content. See Developer section. For example:

type	name	bind::ct:content.qml
integer	animal_count	qml fragment



# Footer

The footer object supports custom control buttons, e.g. back, next, save, etc. If no footer object is specified, then the default control is used.

## buttons

`buttons` is text which specifies which buttons should be shown on the footer toolbar.

type	name	bind::ct:footer.buttons
select_one	animal	back next index save map
<div><div><div>◀▶</div><div><b>survey</b></div><div>choices</div><div>survey</div><div>○</div></div></div>		

🏠

Question

Rhino

Elephant

Lion

Leopard

Buffalo

Giraffe

←

→

📄

📥

📖

## back button

The `back` button navigates to the prior question on the form. If the wizard is at the start of the form, the back button is hidden.

## next button

The `next` button navigates to the next question on the form. If there is no next question, then the next button is hidden.



## save button

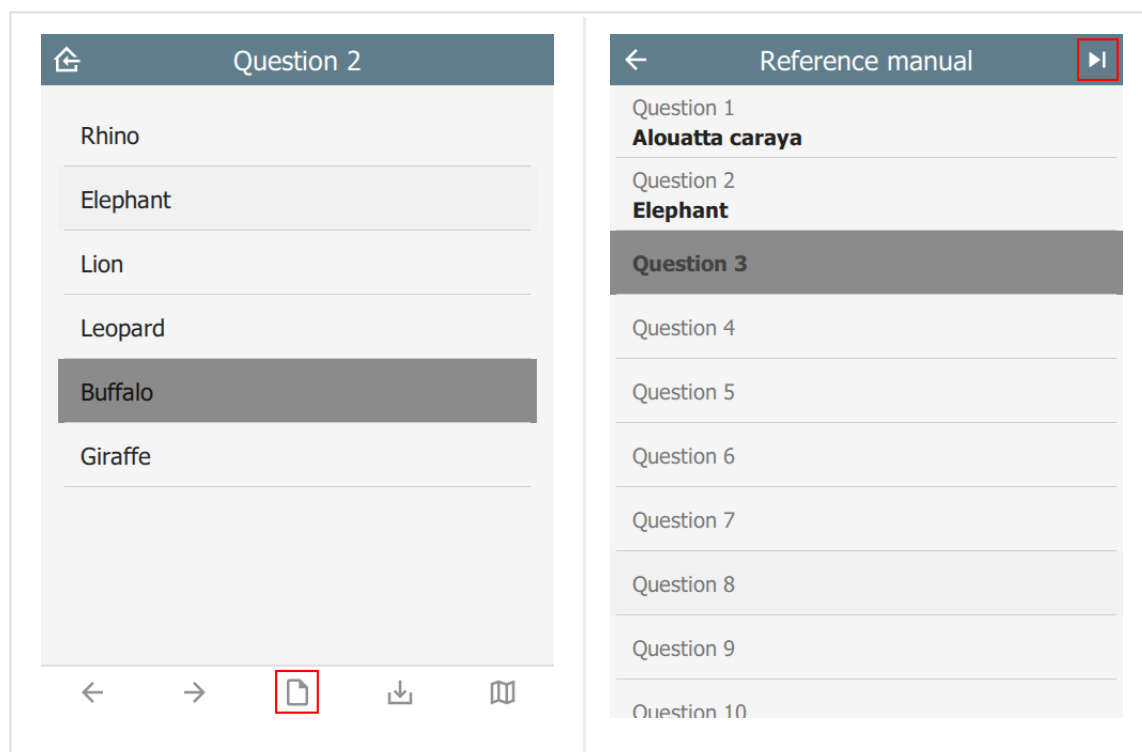
The `save` button will attempt to save the current sighting. If the sighting has invalid data, then the **Index page** will be shown with invalid fields highlighted.

## nextOrSave button

The `nextOrSave` button will show as a `next` button unless there are no more questions, in which case it will become a `save` button.

## index button

The `index` button displays a list of all the form questions. Selecting a question will navigate the wizard to it directly. A jump-to-last button on the top right of the header will jump to the next required question. If all required questions are filled in, then it jumps to the last question.

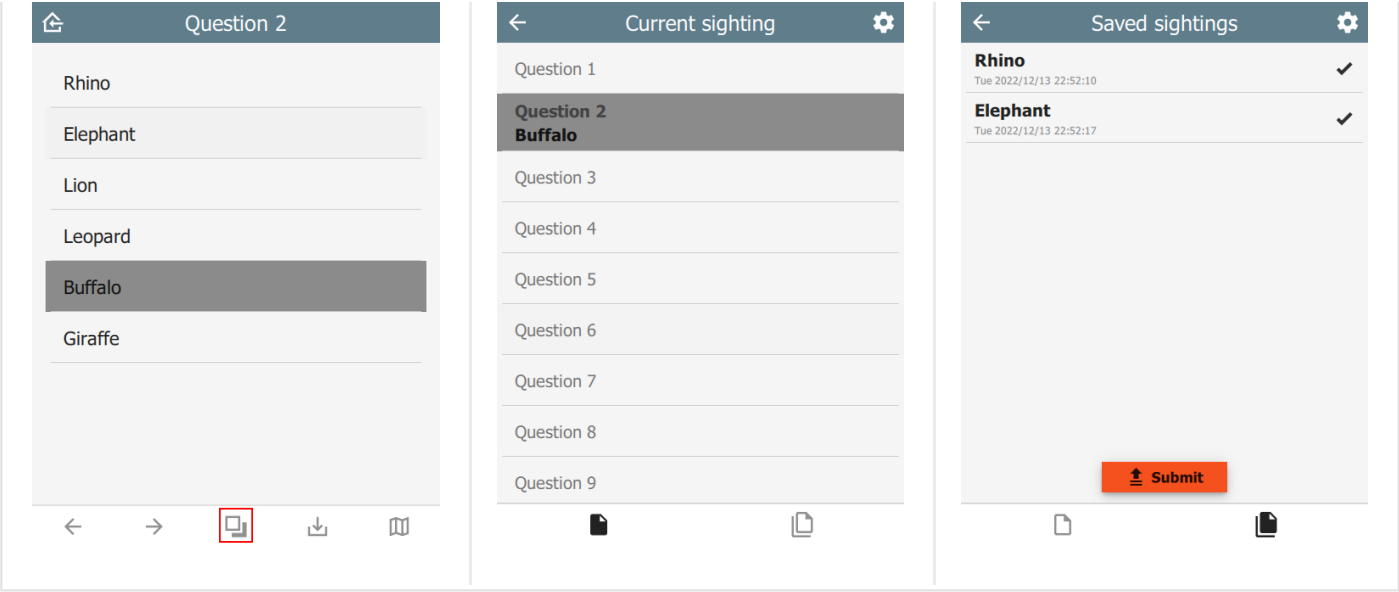


## options button

The `options` button is only available in `immersive` mode. In non-immersive mode, it becomes the `index` button (see above).

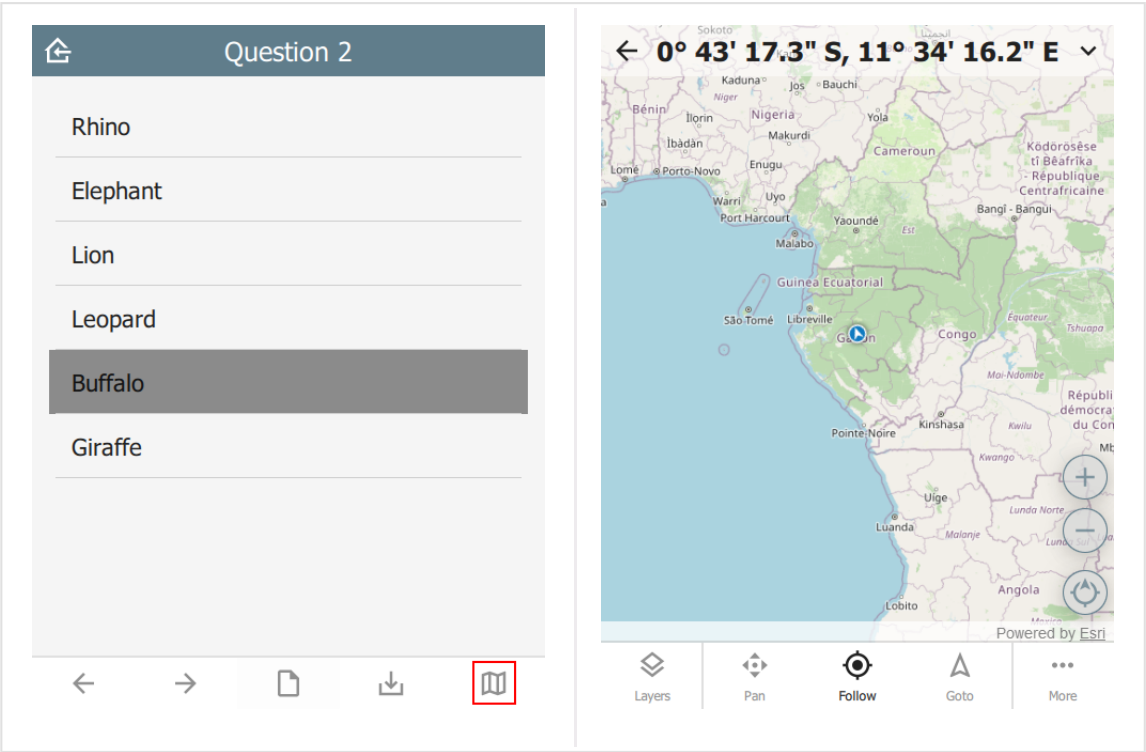
The `options` button shows an options page with two tabs: current sighting and saved sightings:

--	--	--



map button

The `map` button opens the map dialog.



hidden

If `true` then the footer is hidden. Default is `false`.

type	name	bind::ct:footer.hidden
select_one animal	animal	false

qml

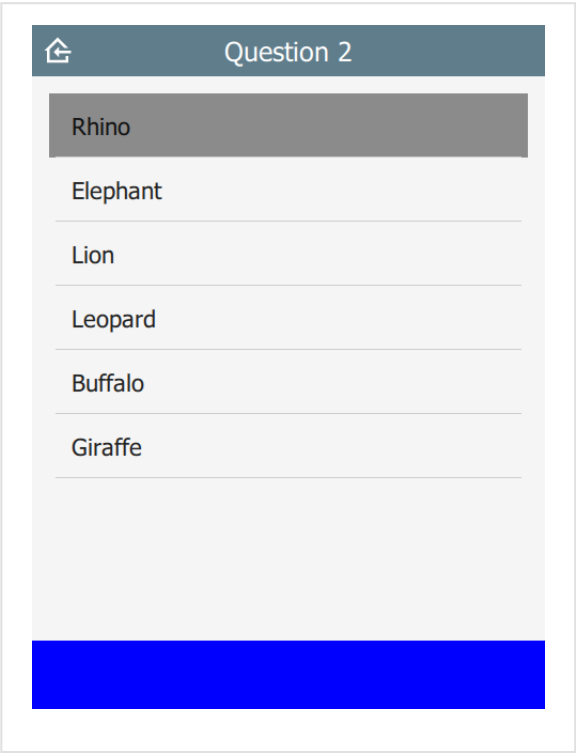
A **QML** fragment to use instead of the built-in footer. See Developer section. For example:

type	name	bind::ct:footer.qml
integer	animal_count	qml fragment
◀▶	survey	choices survey ○

To set the footer to a blue rectangle, replace `qml fragment` above with the following:

```
import QtQuick 2.15

Rectangle {
    color: "blue"
    height: 64
}
```



qmlBase64

Base64 encoded QML (see **qml** above).

qmlFile

Name of a QML file which exists alongside other project files. This is not supported on ODK or KoBoToolbox, but can be used in Survey123.

# Save

When the user presses the **Save** button, this triggers the save behavior.

## snapLocation

Setting `snapLocation` to true will create a popup to acquire the GPS location. The location will be set in the first `geopoint` type question on the form. It is recommended to set appearance to `hidden` on this question, so that the user is not asked for the location beforehand.

type	name	label	appearance	bind::ct:save.snapLocat
geopoint	my_location	Location	hidden	
select_one animal	animal	Animal		
text	note	Field note	multiline	true
<div><div>◀ ▶</div><div>survey</div><div>choices</div><div>survey</div><div>○</div></div>				

In this example, the user flow will be:

Animal

Rhino

Elephant

Lion

Leopard

Buffalo

→

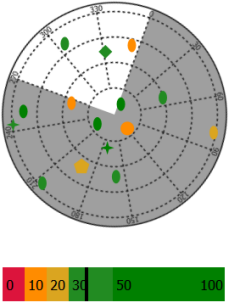
Note

Test note

←

↓

←



## targets

In the example below, the user will be presented with a popup containing the choices **Restart** or **Another**. After the sighting is saved, a new sighting will be created starting at

the targetted question. All prior question data will be replicated into the new sighting.

restart and another are taken from the choices sheet in the saveTargets list name.

question is the name of the targetted question in the survey table. If the question is not relevant, then this choice will be hidden.

type	name	bind::ct:save.targets
integer	animal_count	[[{"choice": "restart", "question": "animal"}, {"choice": "another", "question": "behavior"}]]
<div><div>◀▶</div><div>survey</div><div>choices</div><div>survey</div><div>○</div></div>		

list_name	name	label
saveTarget	restart	Restart
saveTarget	another	Another
<div><div>◀▶</div><div>survey</div><div>choices</div><div>survey</div><div>○</div></div>		

Animal

Rhino

Elephant

Lion

Leopard

Buffalo

→

Behavior

Eating

Sleeping

Walking

Running

Jumping

←

→

Note

Field note

Download

Restart

Another

←

→

track

In the example below, there is a select\_one question called track with choices start, stop and nochange. When the user presses **Save**, the track timer is adjusted depending on which choice was selected. The values in updateIntervalSeconds and distanceFilterMeters are the new track settings.

type		name		bind::ct:trackFileFormat	bind::ct:save.track
file		trackFile		geojson	
select_one track		track			
text		Note			[{ "condition": "selected(\${track}, 'start')", "updateIntervalSeconds": 5, "distanceFilterMeters": 10 }, { "condition": "selected(\${track}, 'stop')", "updateIntervalSeconds": 0, "snapTrack": true }]
<div><div>◀ ▶</div><div>survey</div></div>	choices	survey	<div><div></div></div>		

list_name	name	label
track	start	Start
track	stop	Stop
track	nochange	No change
<div>◀ ▶</div> <div>survey</div> <div>choices</div> <div>survey</div> <div>○</div>		

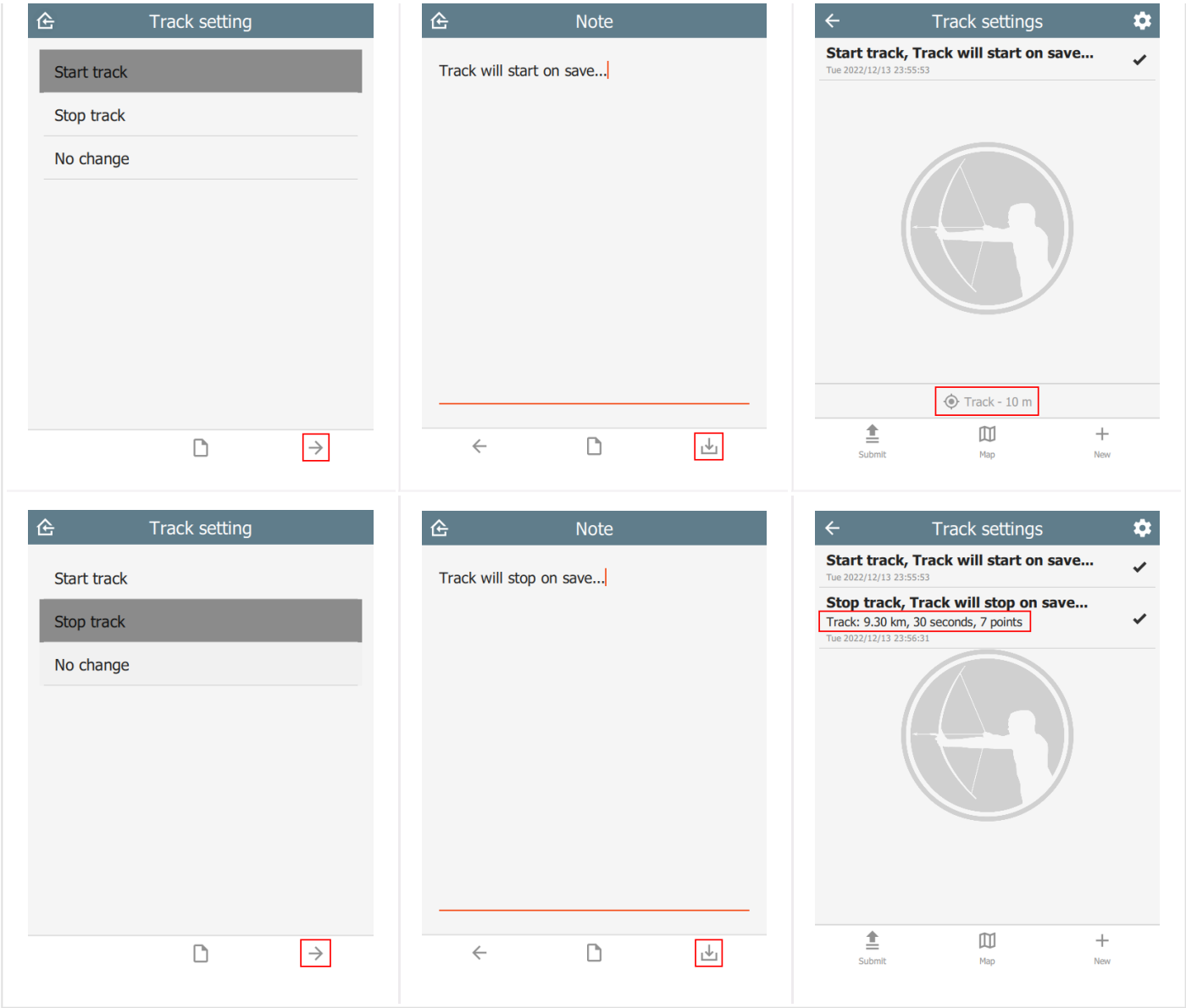
`condition` is an XlsForm expression which activates this option if matched, e.g. `${start_stop}='start'`. Check out the [ODK Form Logic documentation](#).

`updateIntervalSeconds` is the number of seconds between GPS readings. Set to 0 to disable the track timer.

`distanceFilterMeters` is the minimum distance between readings in meters. This is optional and by default no distance filter is used.

`snapTrack` causes the system to snapshot all the track points (since prior snap). If using Survey123 and `bind::ct:esriLocationServiceUrl` is specified, then the track data will be sent to the feature service. Otherwise, a track file will be created and added to a `file` type question in the form. The question selected must be of type `file` and should have a `trackFileFormat` column specified.

--	--	--





## Miscellaneous

### fixCount

For `geopoint` question types, the number of skipped readings before a fix is taken. The default value is 4.

type	name	bind::ct:fixCount
geopoint	my_location	4
◀ ▶	<b>survey</b>	choices   survey   ○

### trackFileFormat

When the user presses **Save** and creates a track file, it is stored in a `file` field which has the `trackFileFormat` value set.

Supported values are `kmz` (not supported on Survey123) and `geojson` (inside a zip).

Survey123 users should prefer to use a location service - see `esriLocationServiceUrl` in the `settings` sheet. If a location service is specified, this question should be removed.

type	name	appearance	bind::ct:trackFormat
file	my_track_file	hidden	kmz
◀ ▶	<b>survey</b>	choices   survey   ○	

# Frequently Asked Questions

## Which backends support XmlForm?

CyberTracker can connect to ODK Central, KoBoToolbox and Survey123 (ArcGIS online).

## Are CyberTracker extensions visible to other tools?

The XlsForm extension system supports custom columns by using the `namespaces` value in the `settings` sheet. Unless these columns are specifically understood, they will be ignored. Columns prefixed with `bind::ct:` add behavior only in CyberTracker.