CyberTracker XlsForm Reference Manual

Overview

CyberTracker is a universal data collection application which runs on mobile devices. One of the supported formats is XlsForm.

XlsForm has an extension mechanism which allows users to activate CyberTracker behavior without affecting the semantics of the form.

This reference manual descibes the CyberTracker extensions. Note that XlsForms are simply an Excel files and the extensions are columns in the worksheets.

There are three kinds of customization:

- Views, e.g. grid styles for single and multi-select lists
- Behaviors, e.g. GPS track logs and save targets
- Developer code, e.g. a new widget

Backend

CyberTracker supports ODK Central, KoBoToolbox and Survey123.

The CyberTracker extensions do not affect the semantics of the form and are transparent to backends. It is possible to use the same form to collect data across platforms (web, ODKCollect, etc) with a single form. In this scenario, CyberTracker would be chosen as a way to meet the needs of specific field workers.

Limitations

While CyberTracker supports most of the commonly used XlsForm features, it is not as mature as the existing data collection tools like ODK Collect, Kobo Collect and Survey123. Users should prefer to use those tools for mission critical projects.

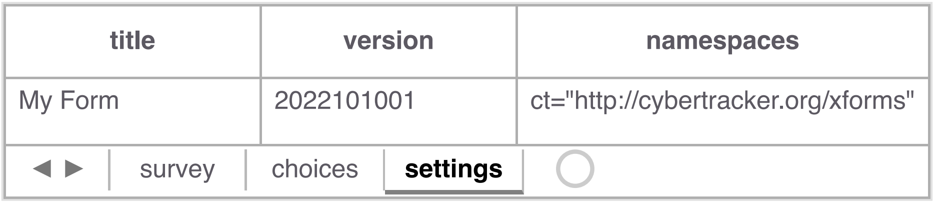
TABLE OF CONTENTS

Initial setup

The following 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
settings
○

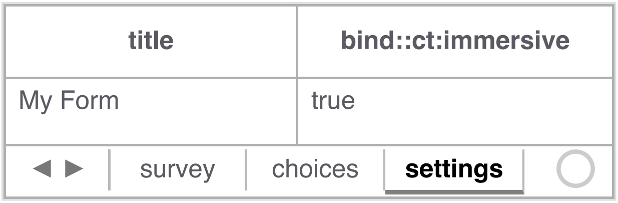
version (recommended)

The version field is used to track form versions over time. While not strictly required, it is a best practice to keep this field up to date. The XlsForm specification recommends the convention of ‘yyyymmddrr’. For example, 2022021501 is the 1st revision from Feb 15th, 2022.

Settings

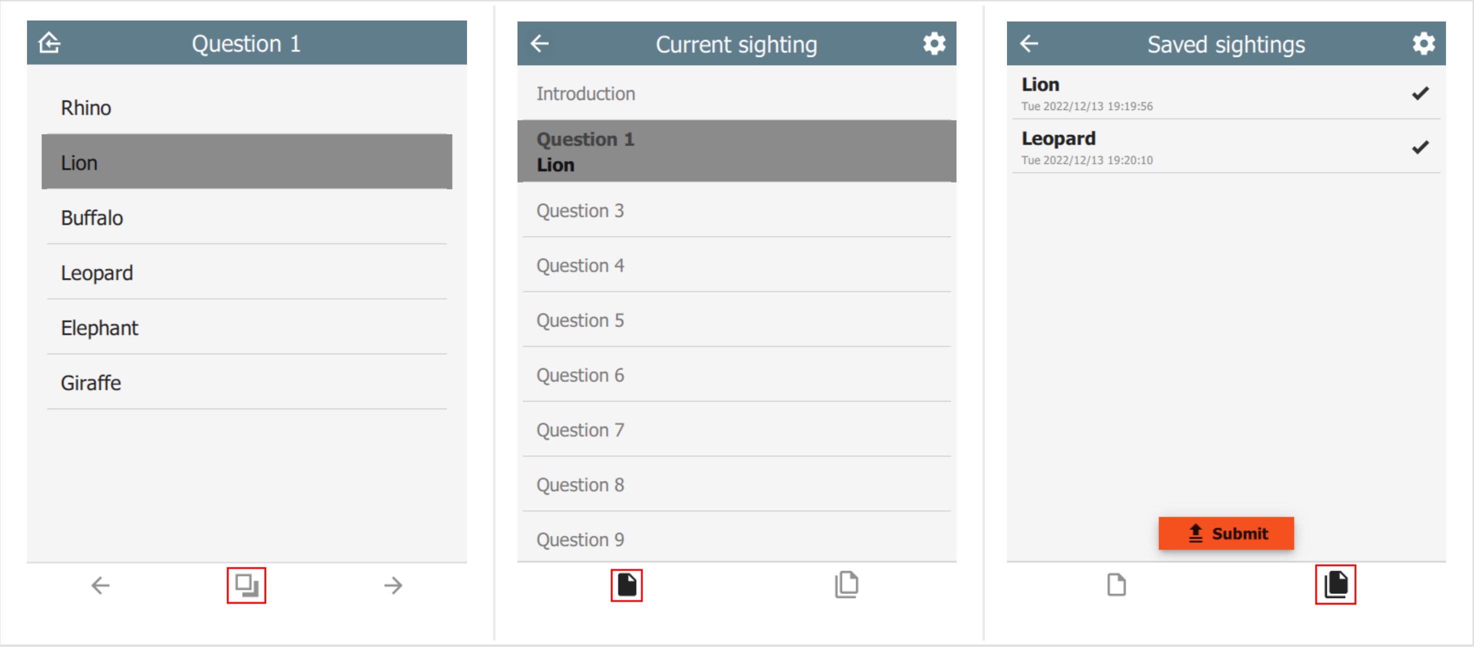
immersive

Setting this to `yes` causes the UI to use the wizard exclusively, i.e. there is no **Home** page. Default is `no`.



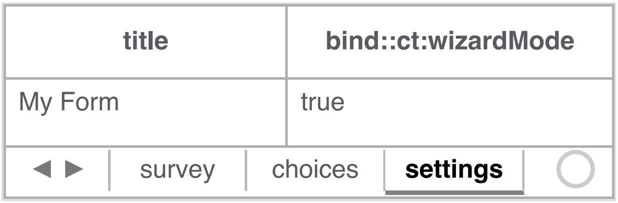
title	bind::ct:immersive
My Form	true

In the table below, the user context is always within a sighting and each page typically holds one question. Pressing the **options** button (highlighted in the first image) navigates to a new page which shows 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.



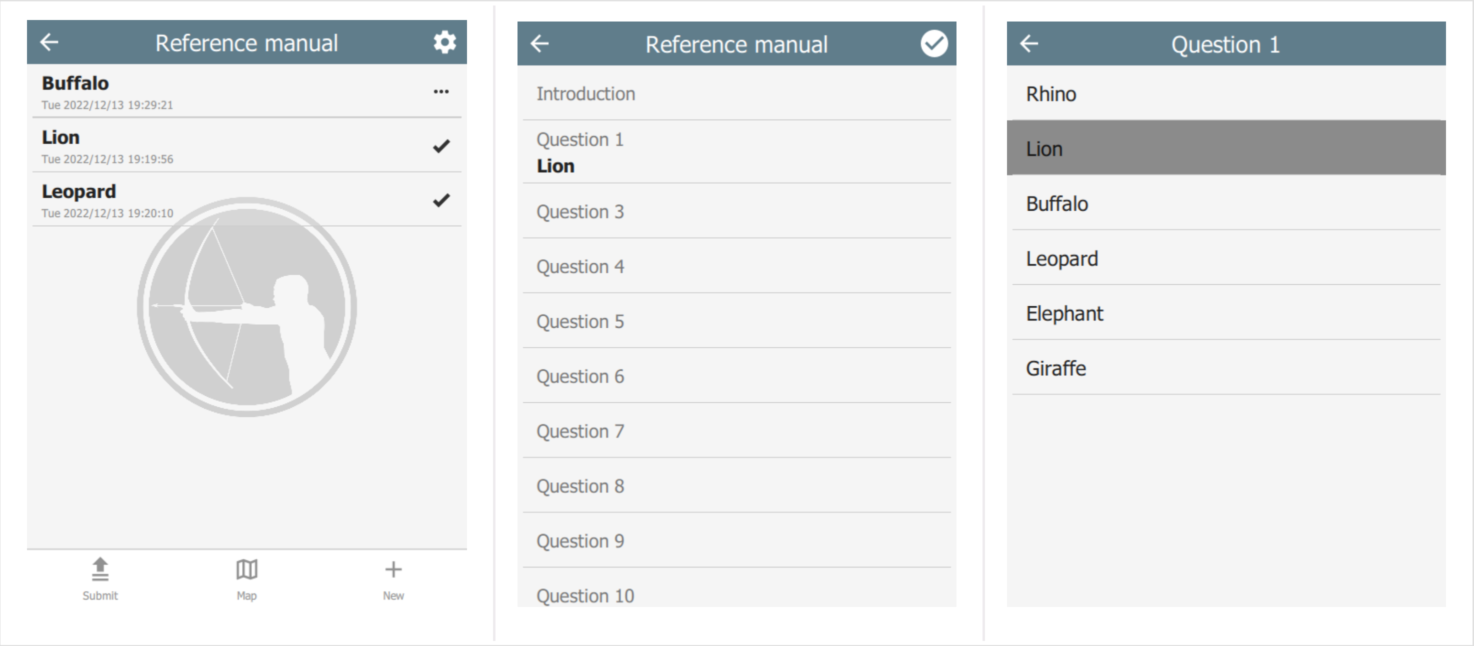
wizardMode

If the `immersive` column is missing or set to `no`, 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 page after saving a sighting.

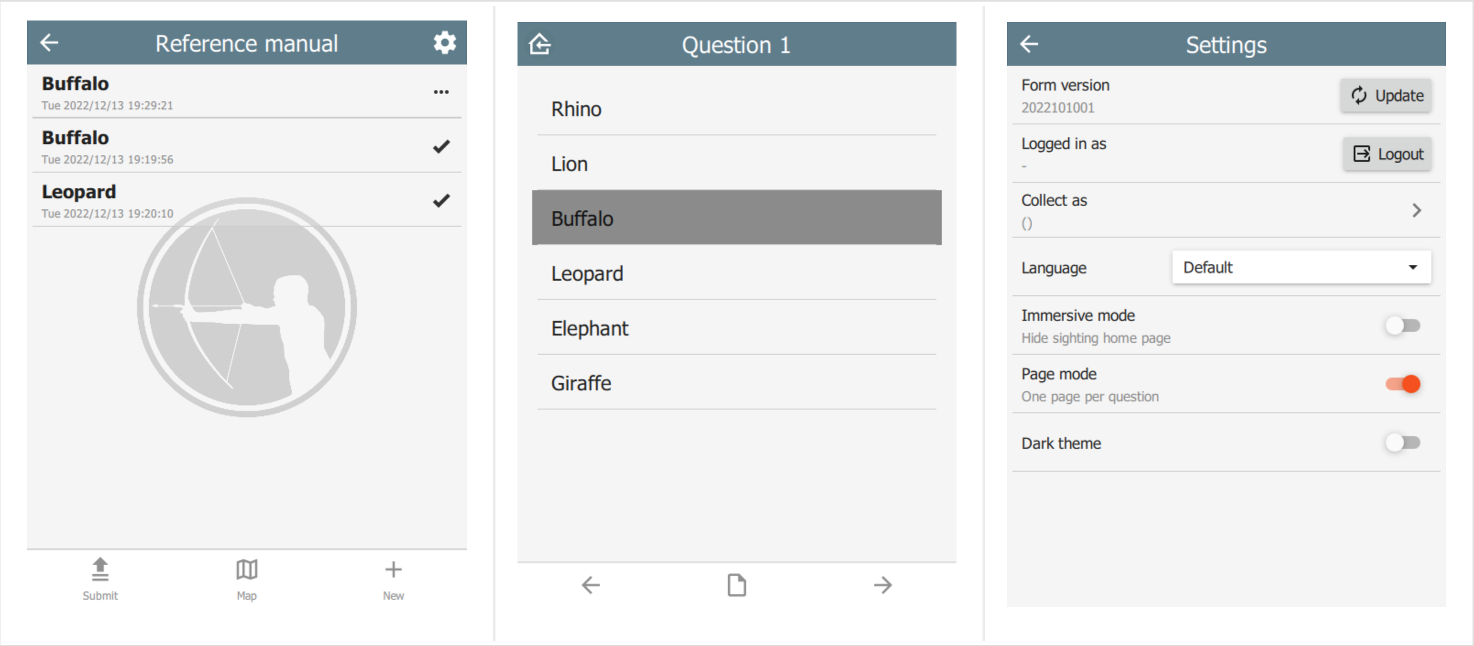


title	bind::ct:wizardMode
My Form	true

If `wizardMode` is set to `no`, 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 `yes`, 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**.



Form version
2022101001

Logged in as
-

Collect as
()

Language: Default

Immersive mode
Hide sighting home page

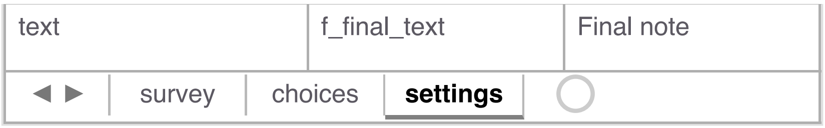
Page mode
One page per question

Dark theme

summary

The summary attribute specifies which fields to use as the summary of a sighting on the **Home** page. For example:

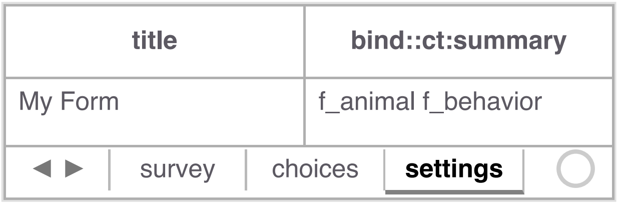
type	name	label
text	f_initial_text	Initial note
select_one animal	f_animal	Animal
select_multiple behavior	f_behavior	Behavior



text	f_final_text	Final note

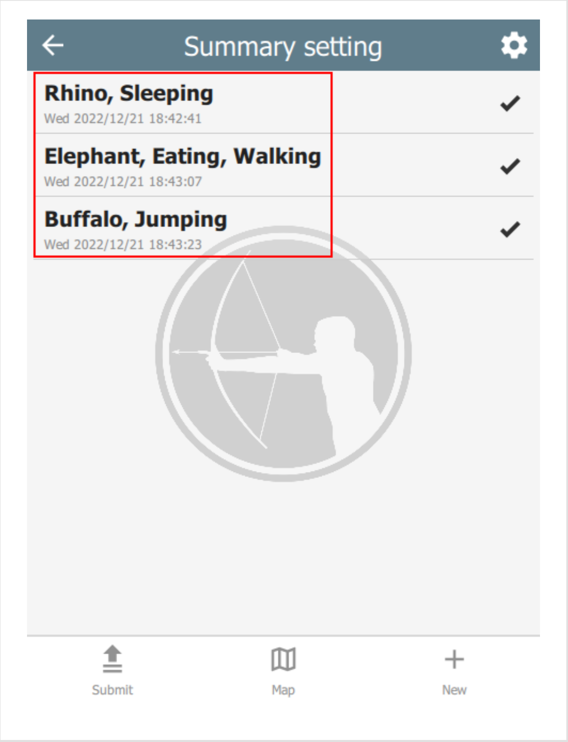
◀ ▶
survey
choices
settings

The following setting will ensure that the summary only uses the `f_animal` and `f_behavior` questions.



title	bind::ct:summary
My Form	f_animal f_behavior

◀ ▶
survey
choices
settings



Summary setting

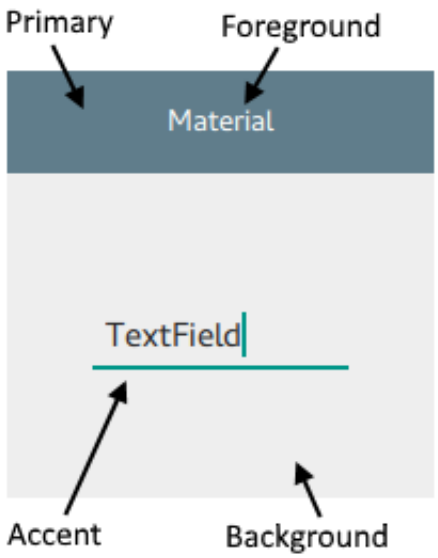
Rhino, Sleeping ✓
Wed 2022/12/21 18:42:41

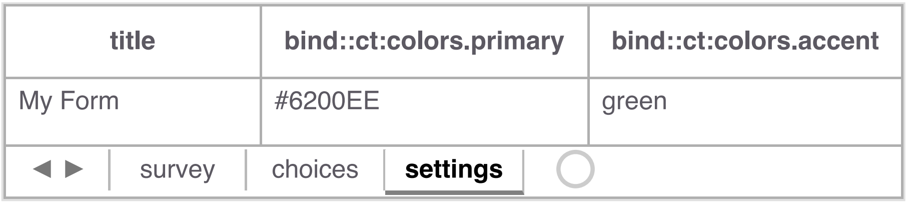
Elephant, Eating, Walking ✓
Wed 2022/12/21 18:43:07

Buffalo, Jumping ✓
Wed 2022/12/21 18:43:23

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

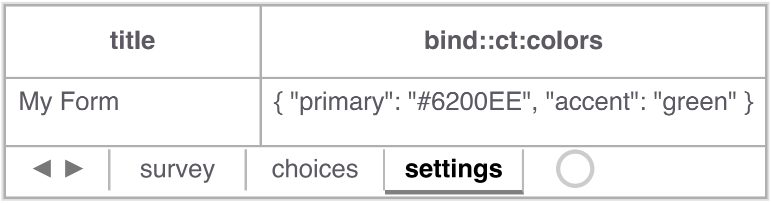
◀ ▶ survey choices **settings** ○

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:

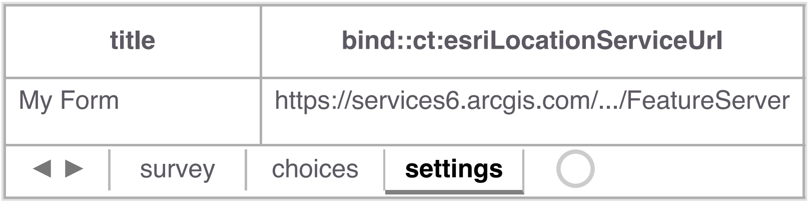


title	bind::ct:colors
My Form	{ "primary": "#6200EE", "accent": "green" }

◀ ▶ survey choices **settings** ○

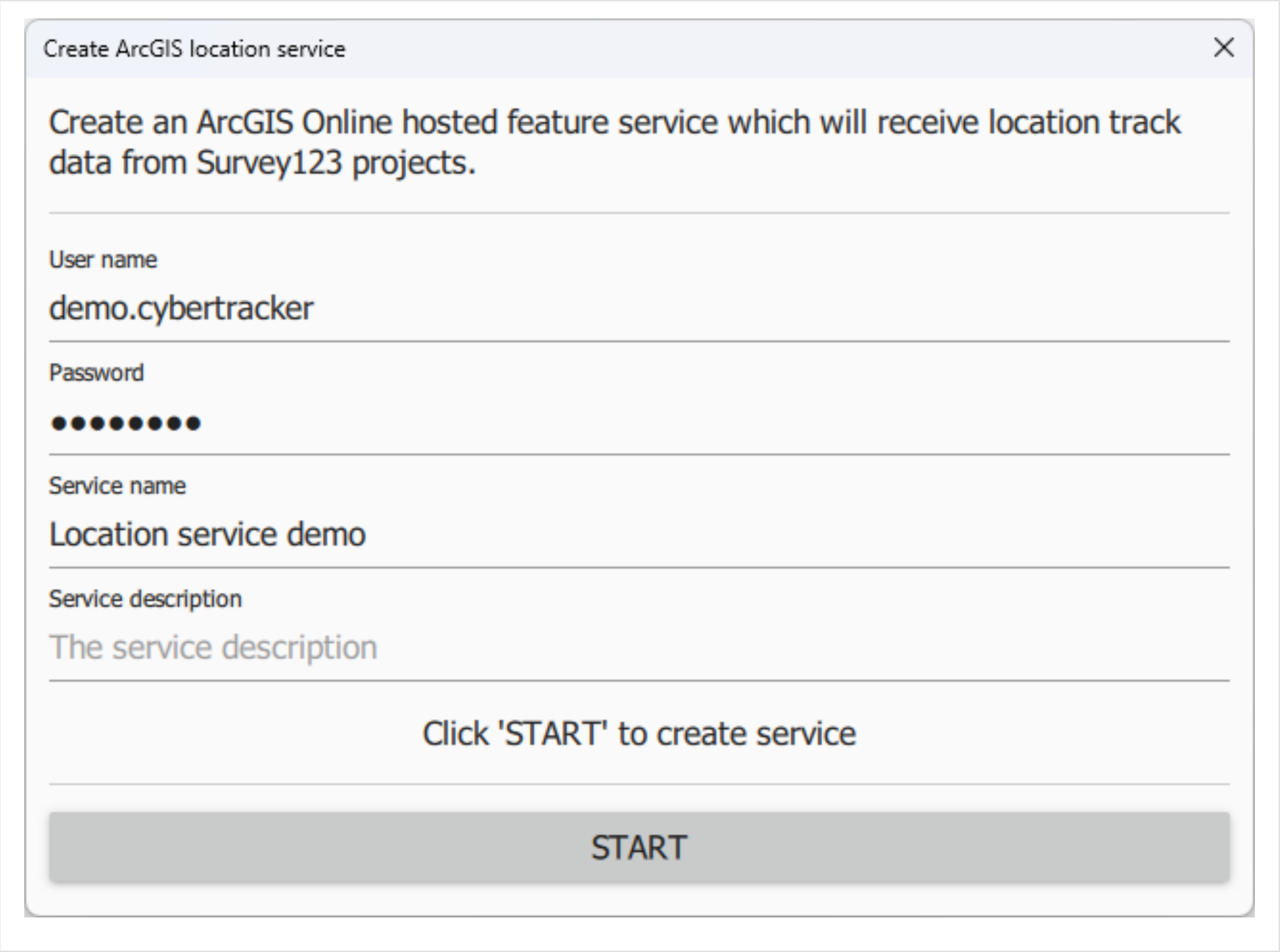
esriLocationServiceUrl

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



title	bind::ct:esriLocationServiceUrl
My Form	https://services6.arcgis.com/.../FeatureServer

The feature service should be created using the CyberTracker Desktop Simulator (see Download page). There is an option off the **Tools** menu called **Create ArcGIS location service**. This tool will automatically create and configure a hosted service which is compatible with CyberTracker:



Create ArcGIS location service

Create an ArcGIS Online hosted feature service which will receive location track data from Survey123 projects.

User name
demo.cybertracker

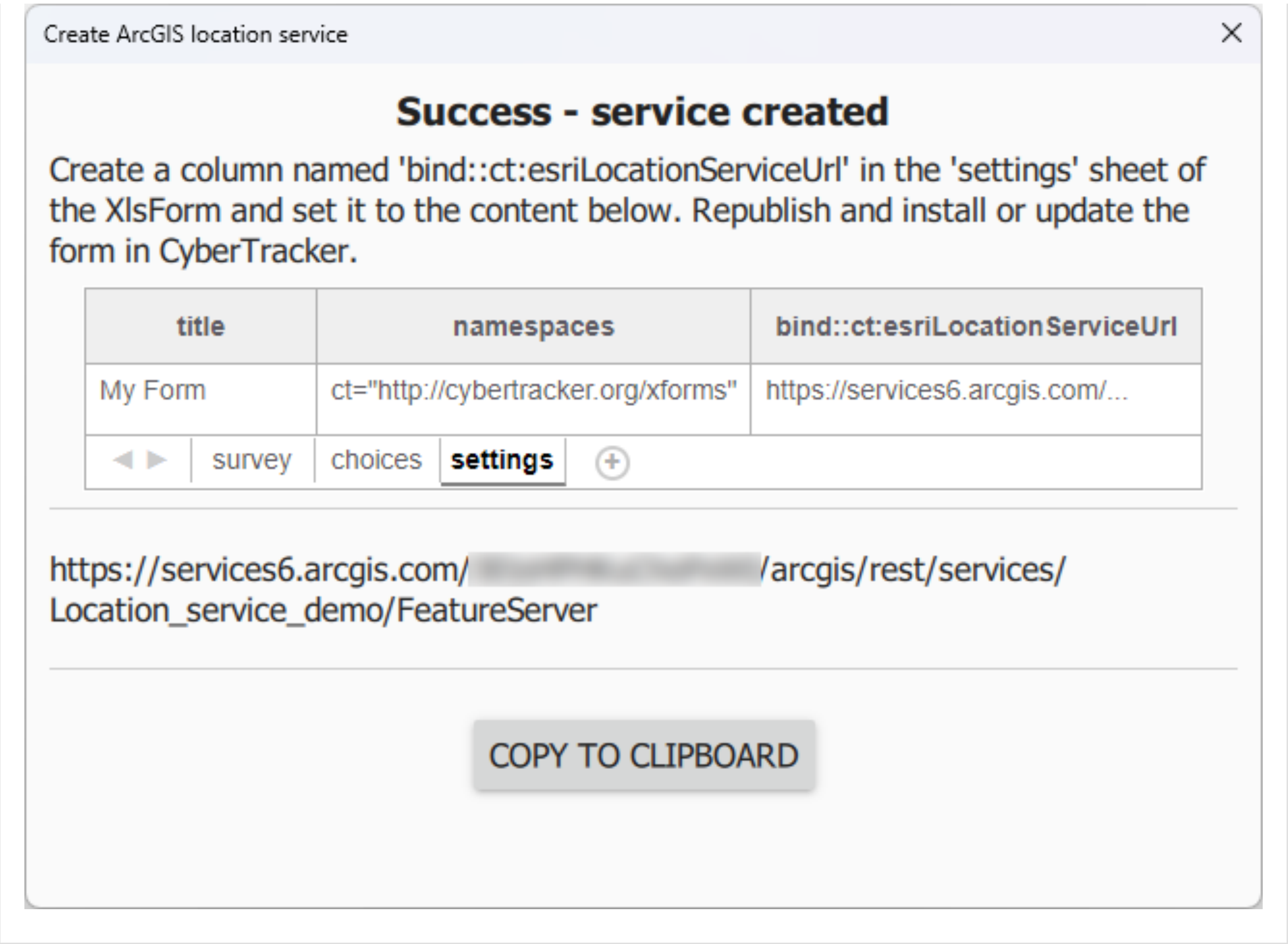
Password
●●●●●●●●

Service name
Location service demo

Service description
The service description

Click 'START' to create service

After clicking **Start**, the tool will display the following:



Success - service created

Create a column named 'bind::ct:esriLocationServiceUrl' in the 'settings' sheet of the XlsForm and set it to the content below. Republish and install or update the form in CyberTracker.

title	namespaces	bind::ct:esriLocationServiceUrl
My Form	ct="http://cybertracker.org/xforms"	https://services6.arcgis.com/...

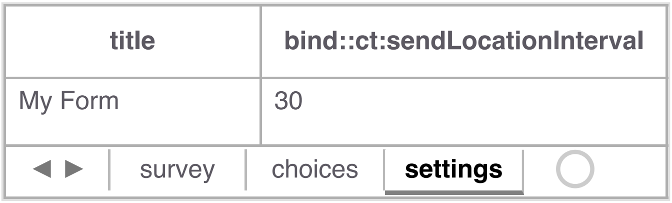
https://services6.arcgis.com/[redacted]/arcgis/rest/services/Location_service_demo/FeatureServer

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. See Tracks.

sendLocationInterval

If using Survey123, CyberTracker can send the current location at regular intervals - separately from tracks. The value is in seconds and is user configurable via the form Settings menu on the device. This specifies the default value. `bind::ct:esriLocationServiceUrl` must be configured.



title	bind::ct:sendLocationInterval
My Form	30

Header

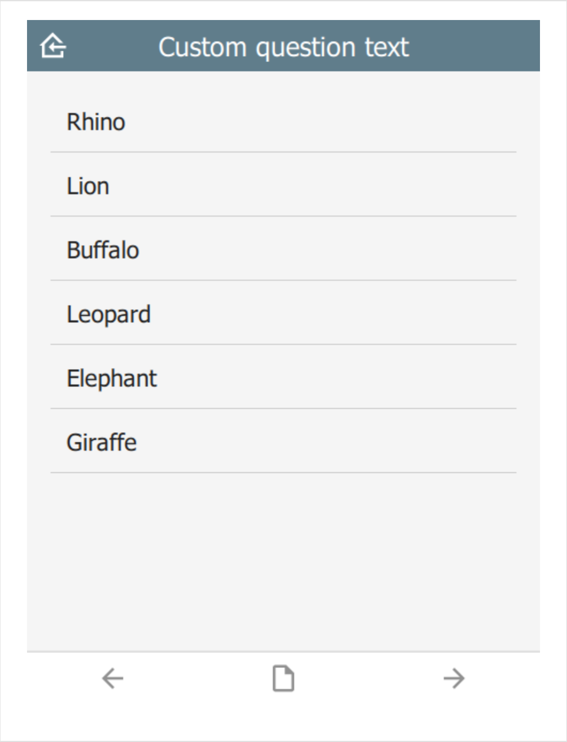
The header object supports custom header attributes. If no header object is specified, then the default header is used. By default, the header title is taken from the question label.

text

Custom header text.

type	name	bind::ct:header.text
select_one animal	Animal	Custom question text

◀ ▶ **survey** choices settings ○


Custom question text

Rhino

Lion

Buffalo

Leopard

Elephant

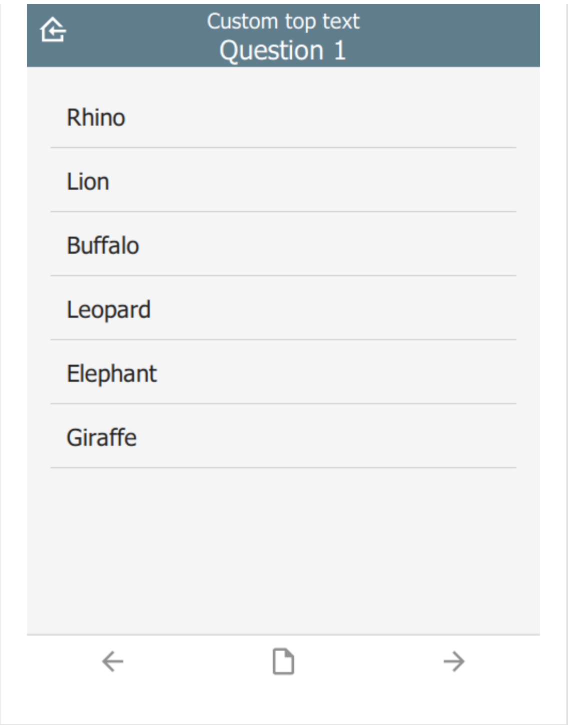
Giraffe

topText

Custom smaller text above main title.

type	name	bind::ct:header.topText
select_one animal	animal	Custom top text

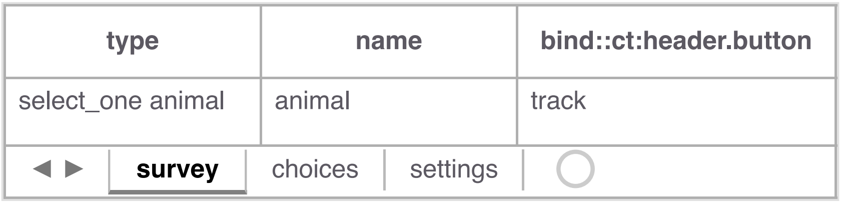
◀ ▶ **survey** choices settings ○



button

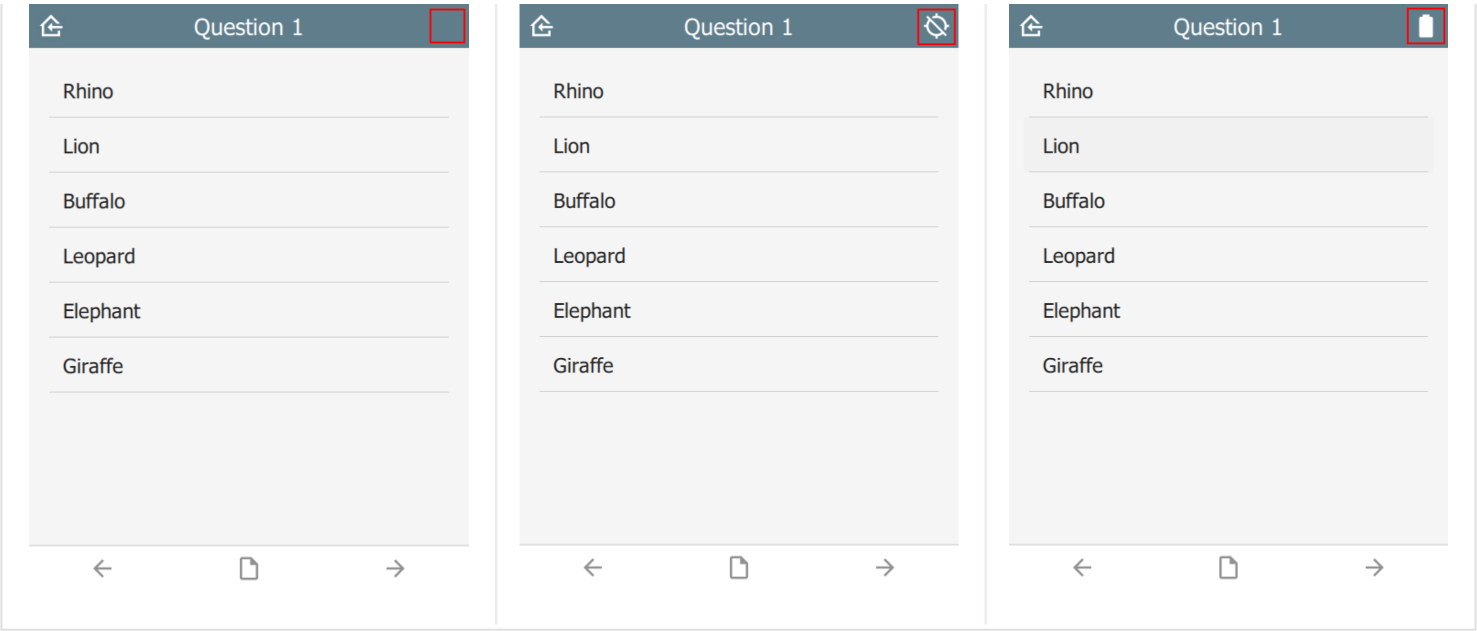
Type of the button in the top-right corner. Valid values are:

- 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



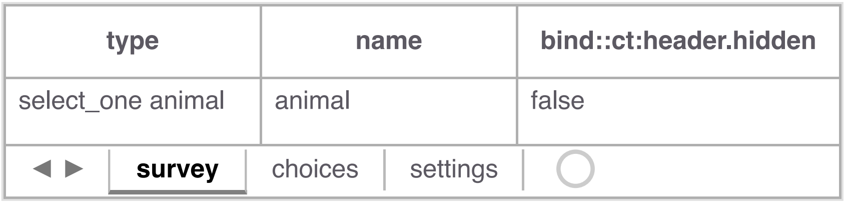
type	name	bind::ct:header.button
select_one animal	animal	track

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



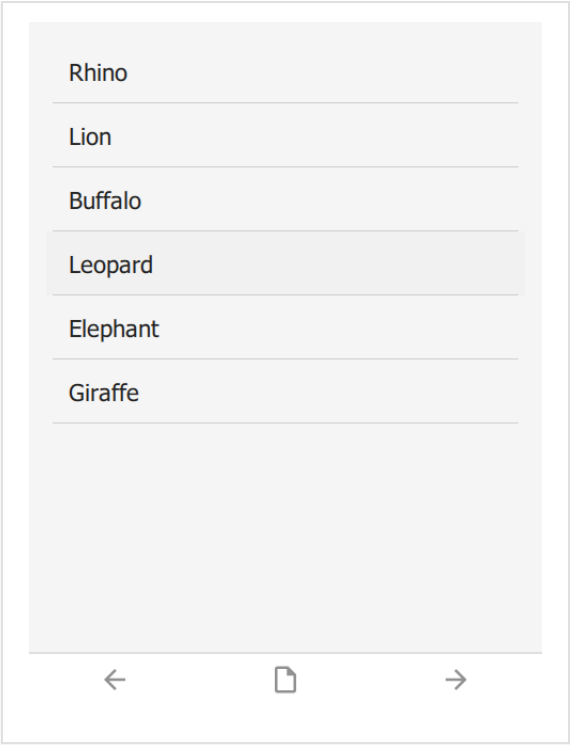
hidden

If `yes` then the header is hidden. Default is `no`.



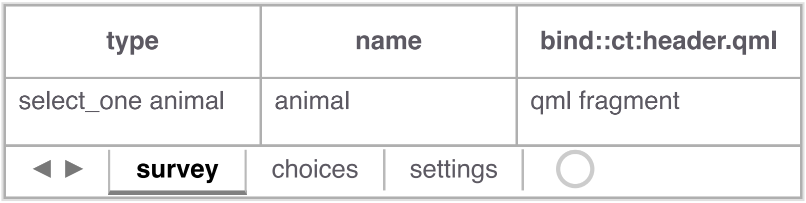
type	name	bind::ct:header.hidden
select_one animal	animal	false

◀ ▶ survey
choices
settings



qml

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



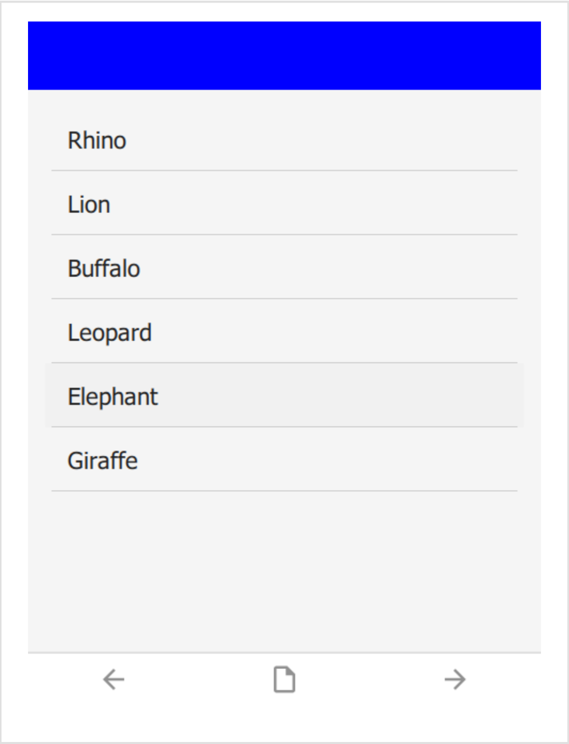
type	name	bind::ct:header.qml
select_one animal	animal	qml fragment

◀ ▶
survey
choices
settings
○

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

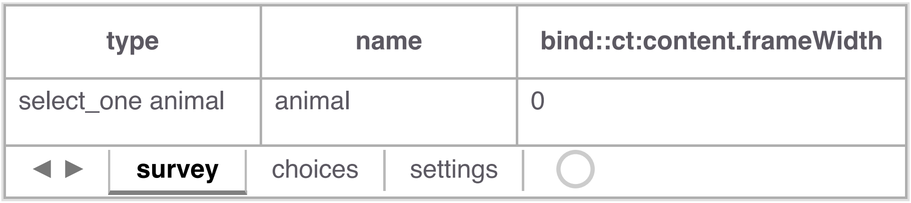
Name of a QML file which exists alongside other project files. This is not supported on ODK or KoBoToolbox, but can be used in Survey123.

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.

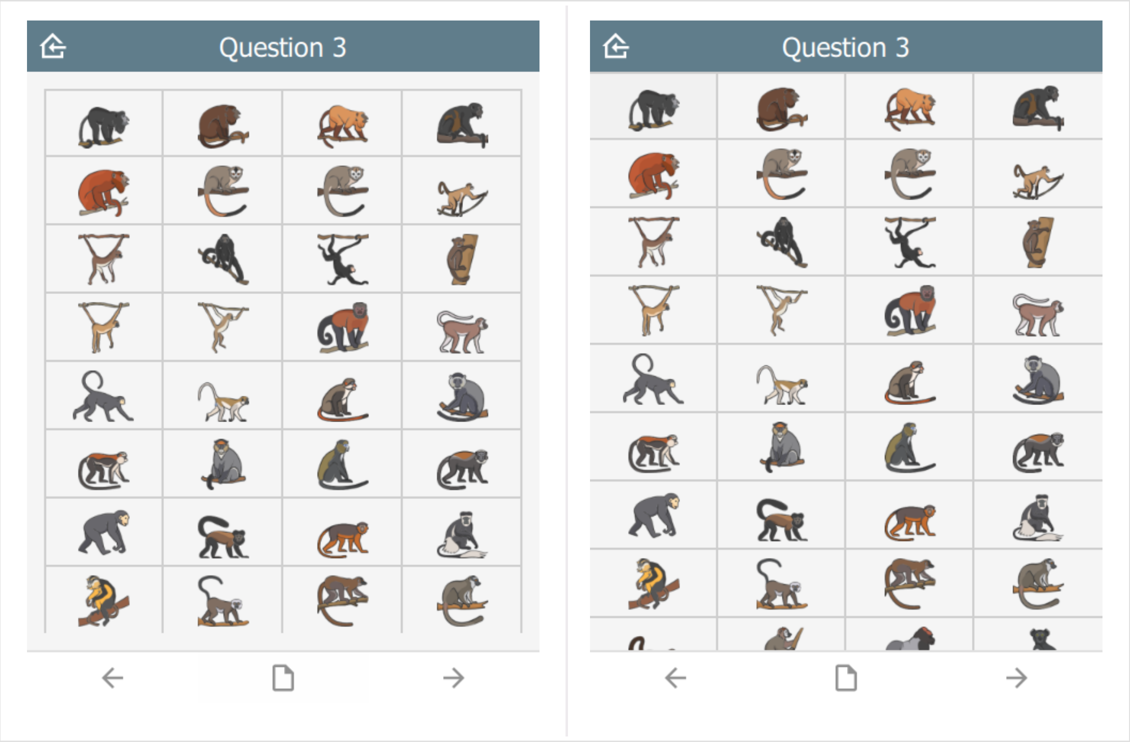
frameWidth

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



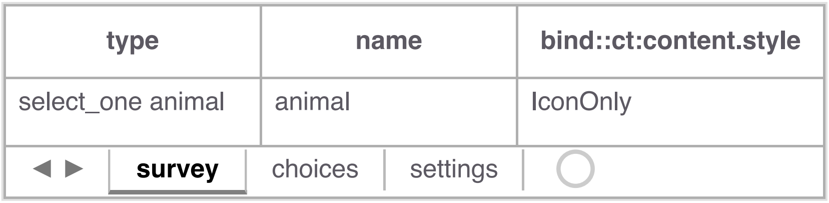
type	name	bind::ct:content.frameWidth
select_one animal	animal	0

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



style

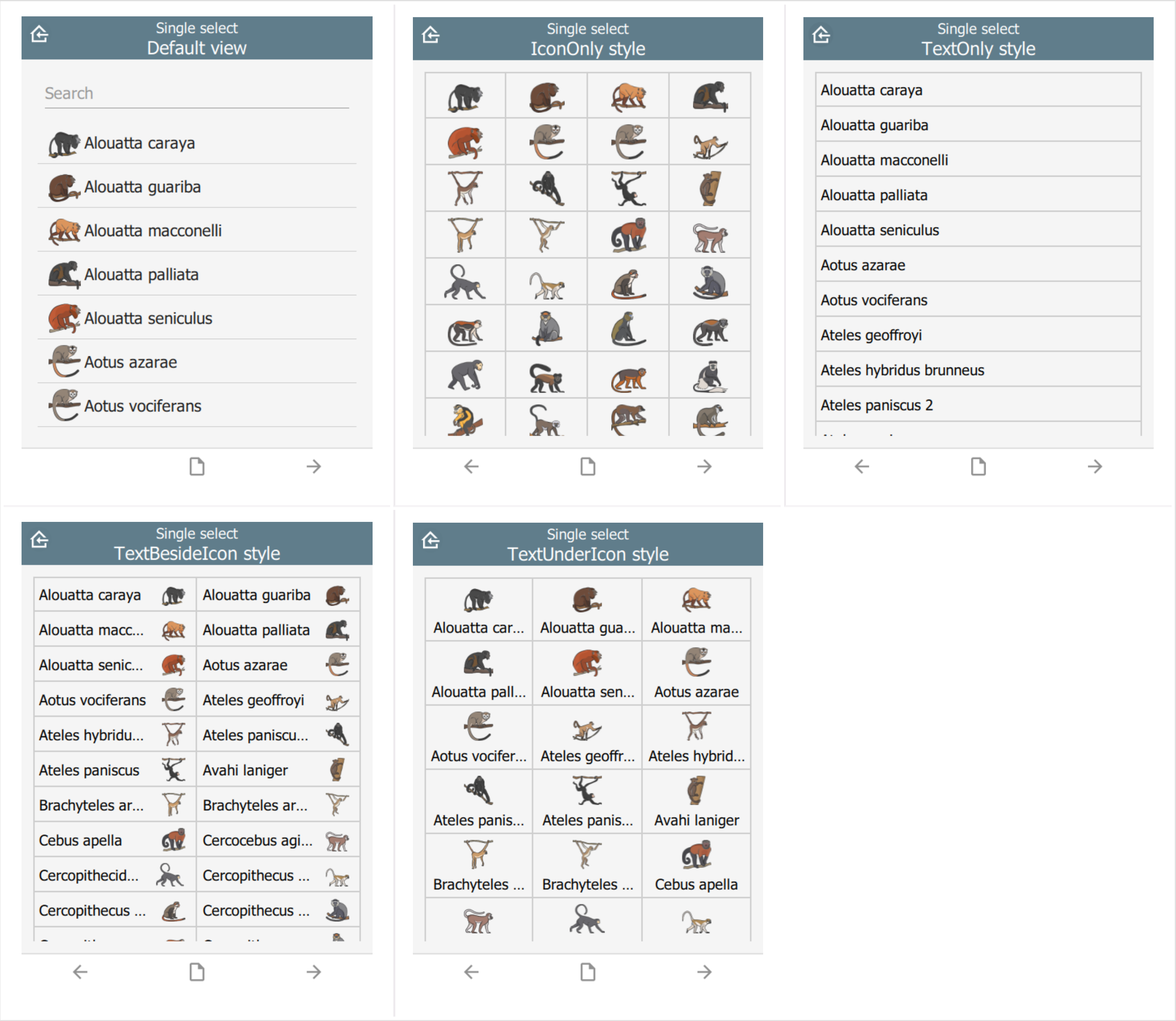
The visual appearance of the question.



type	name	bind::ct:content.style
select_one animal	animal	IconOnly

For **select_one** questions

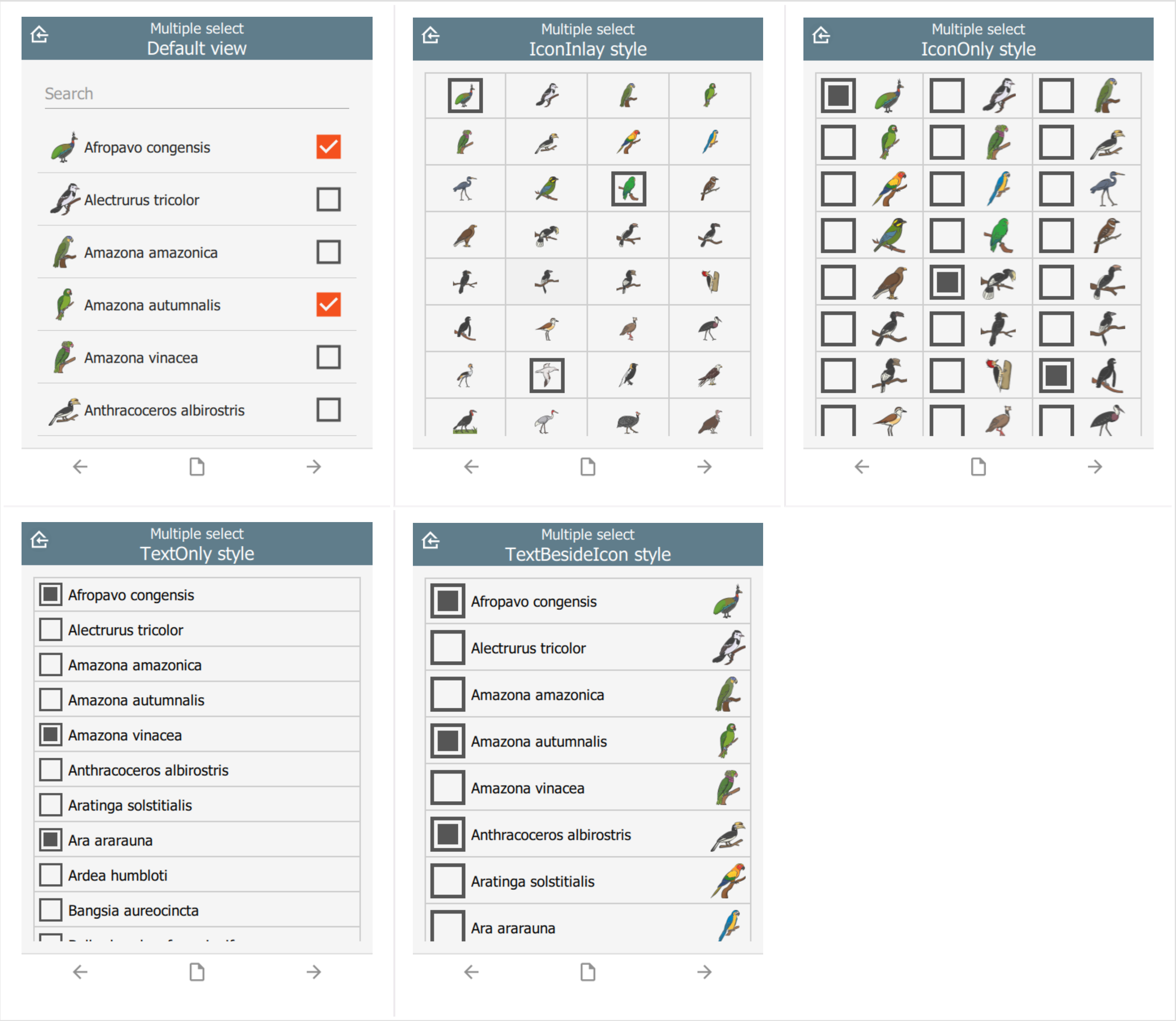
- (not specified)
- IconOnly
- TextOnly
- TextBesideIcon
- TextUnderIcon



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

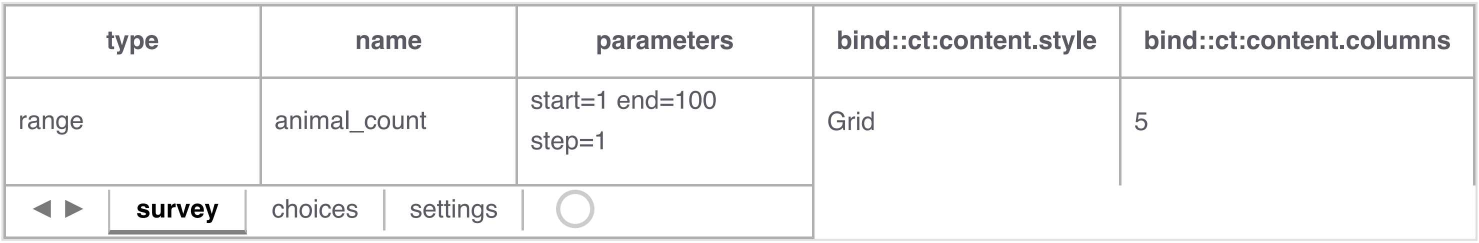
Multiple select
TextOnly style

- ■ Afropavo congensis
- □ Alectrurus tricolor
- □ Amazona amazonica
- □ Amazona autumnalis
- ■ Amazona vinacea
- □ Anthracoceros albirostris
- □ Aratinga solstitialis
- ■ Ara ararauna
- □ Ardea humbloti
- □ Bangsia aureocincta

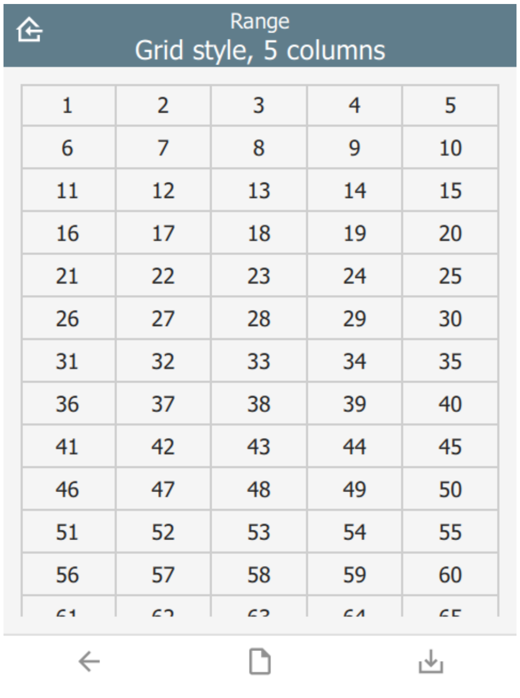
Multiple select
TextBesideIcon style

- ■ 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.columns
range	animal_count	start=1 end=100 step=1	Grid	5

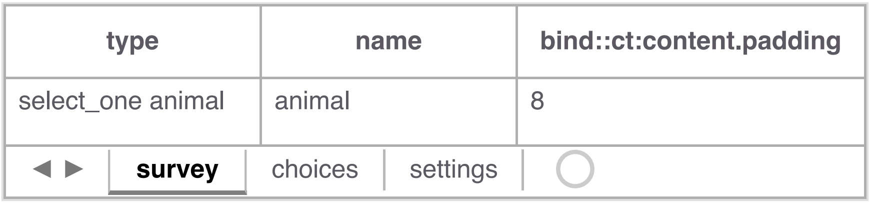


1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50
51	52	53	54	55
56	57	58	59	60
61	62	63	64	65

Ignored for other question types.

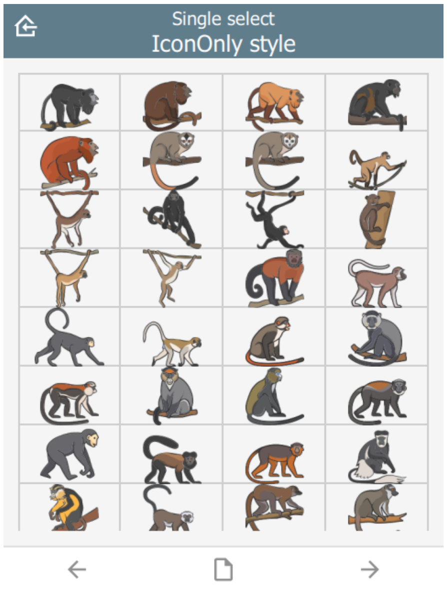
padding

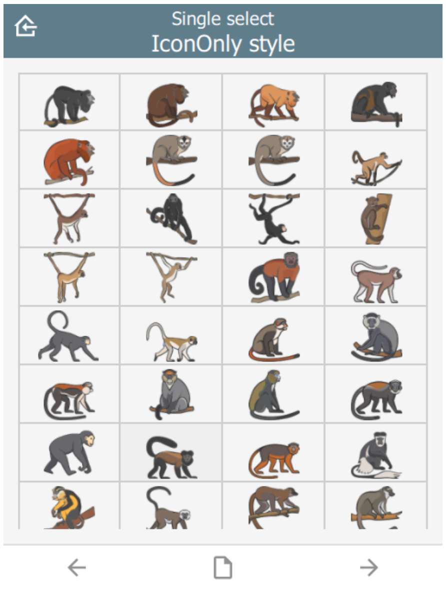
The padding space between grid items. Requires `style` to be set.

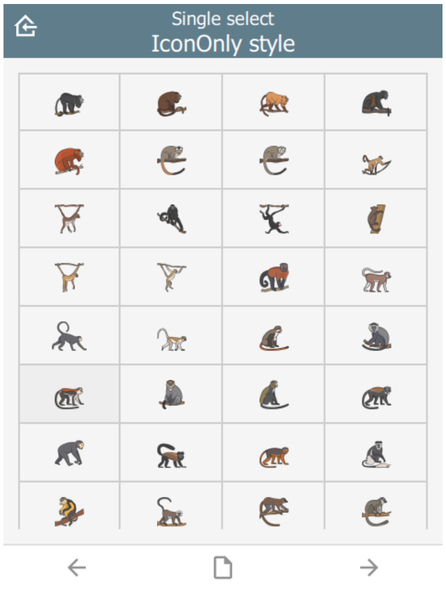


type	name	bind::ct:content.padding
select_one animal	animal	8

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

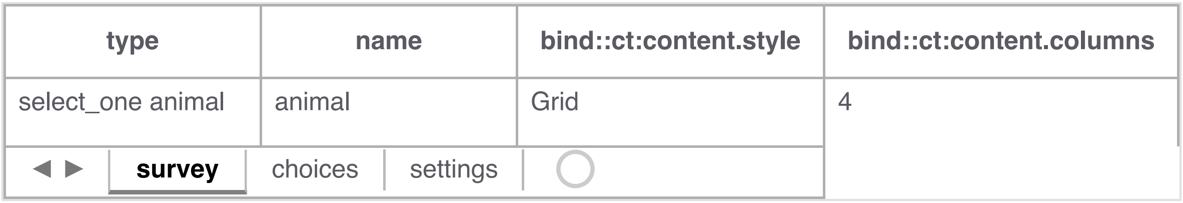






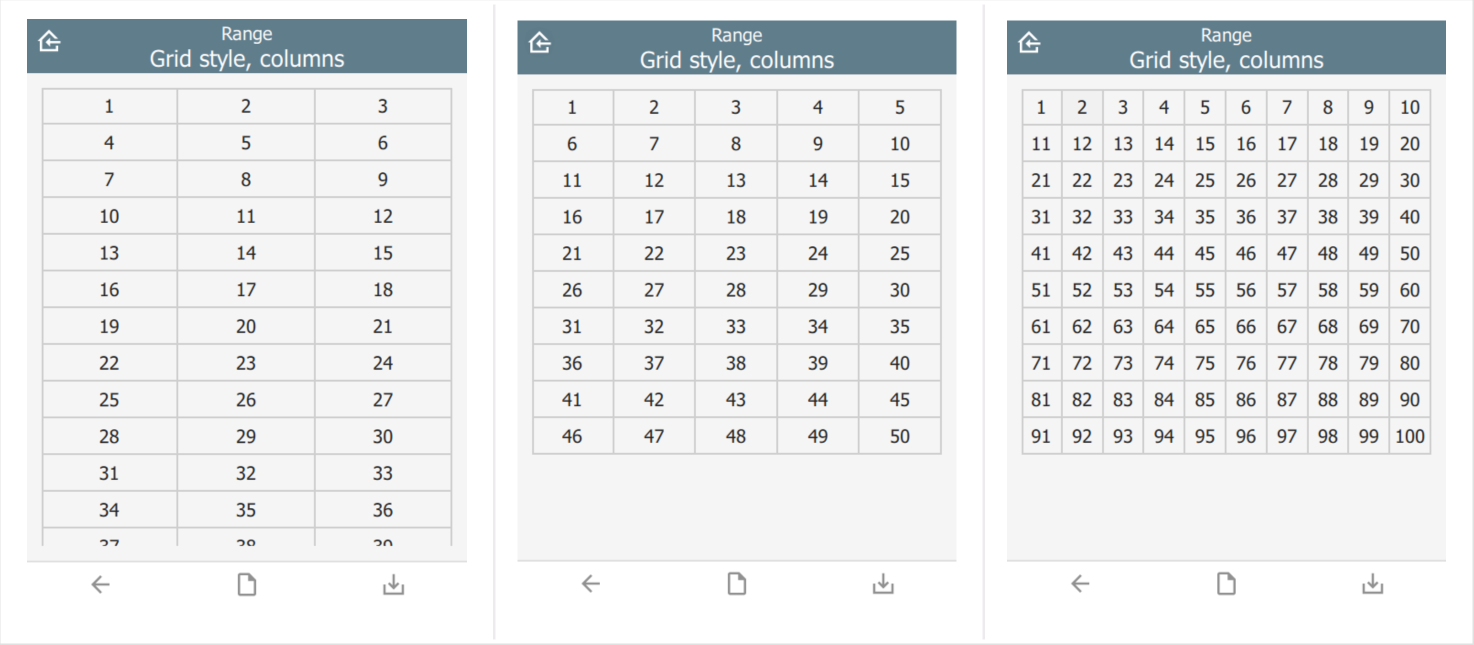
columns

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



type	name	bind::ct:content.style	bind::ct:content.columns
select_one animal	animal	Grid	4

For example, column values below are `3`, `5` and `10`.



Range
Grid style, 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

Range
Grid style, columns

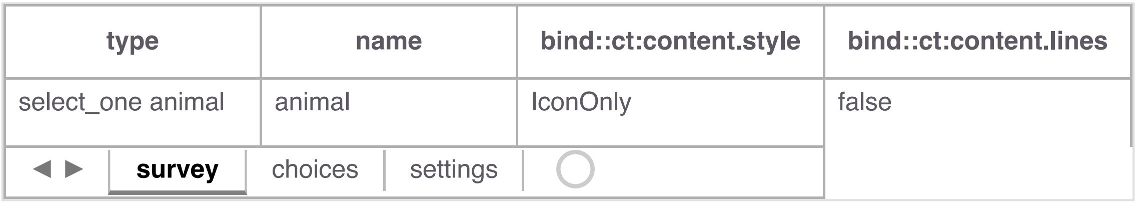
1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Range
Grid style, 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	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

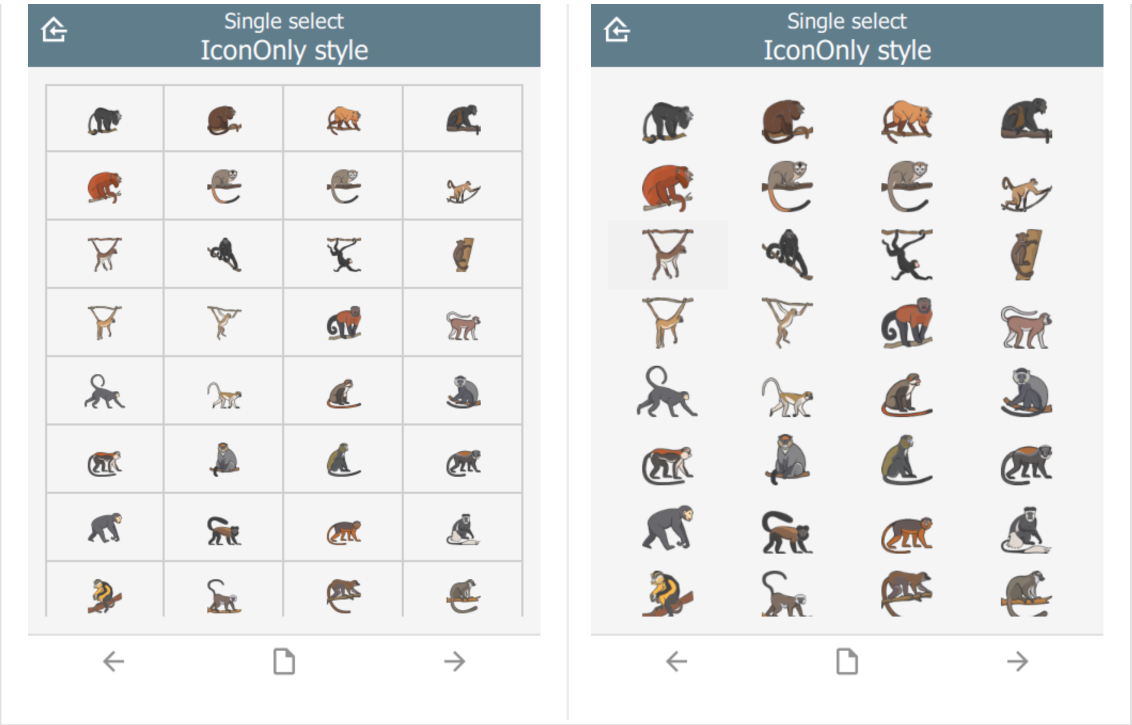
lines

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



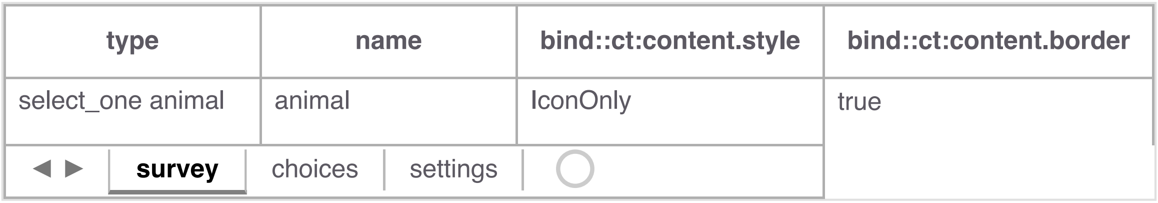
type	name	bind::ct:content.style	bind::ct:content.lines
select_one animal	animal	IconOnly	false

`lines` value below is `yes` and `no`.



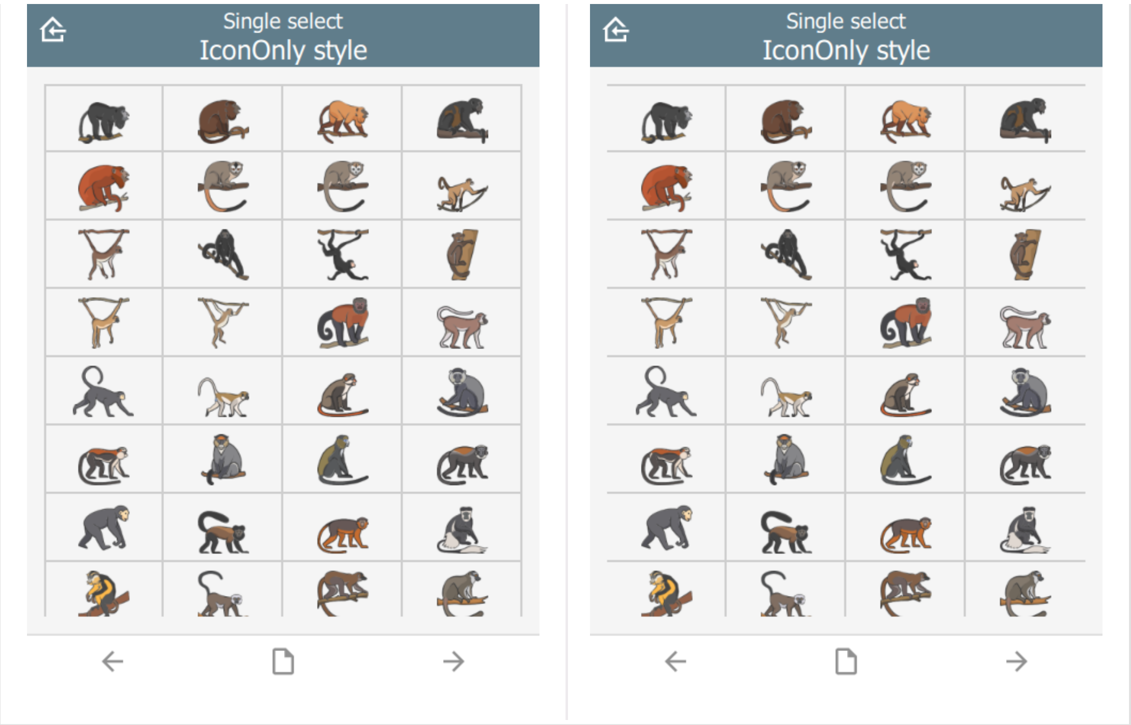
border

Show border around the outside of a grid. Requires `style` to be set. Defaults to `no` if frameWidth is 0, `yes` otherwise.



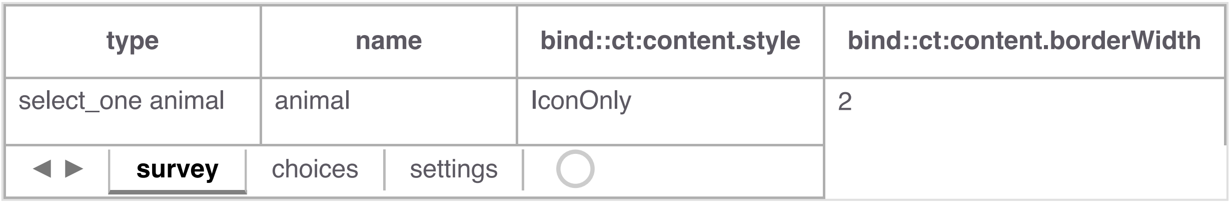
type	name	bind::ct:content.style	bind::ct:content.border
select_one animal	animal	IconOnly	true

Border value is `yes` and `no`.



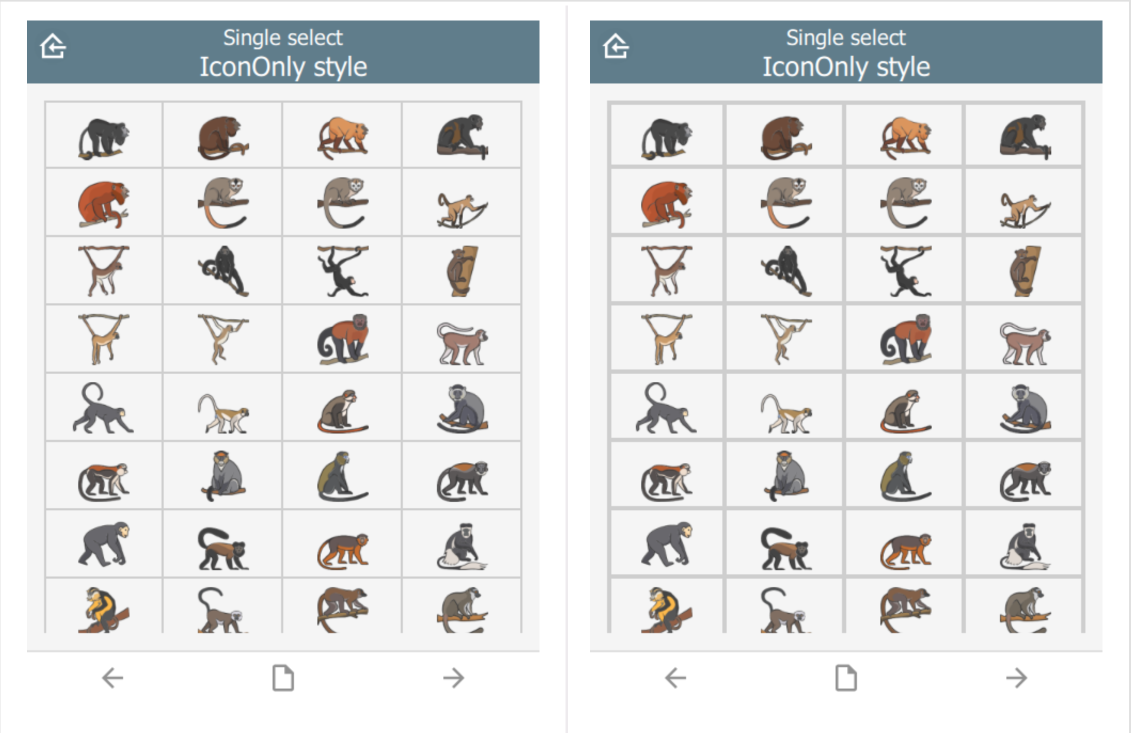
borderWidth

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



type	name	bind::ct:content.style	bind::ct:content.borderWidth
select_one animal	animal	IconOnly	2

borderWidth value is `2` and `4`.



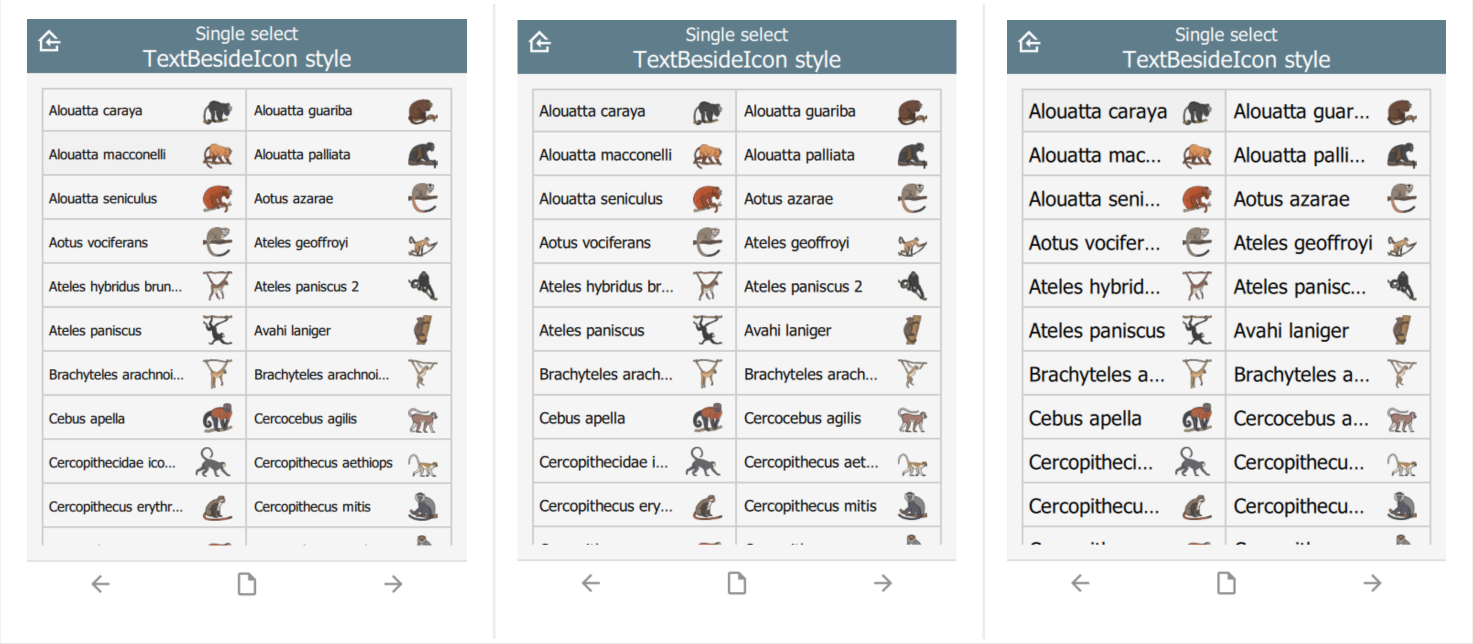
fontSize

Size text font size. Requires `style` to be set. Defaults to 16. Note that the font is subject to scaling according to the **Font size** in the main Settings page.

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

◀ ▶ **survey** choices settings ○

`fontSize` values are `12`, `14` and `16`.



Single select
TextBesideIcon style

Alouatta caraya	Alouatta guariba
Alouatta macconelli	Alouatta palliata
Alouatta seniculus	Aotus azarae
Aotus vociferans	Ateles geoffroyi
Ateles hybridus brun...	Ateles paniscus 2
Ateles paniscus	Avahi laniger
Brachyteles arachnoi...	Brachyteles arachnoi...
Cebus apella	Cercocebus agilis
Cercopithecidae ico...	Cercopithecus aethiops
Cercopithecus erythr...	Cercopithecus mitis

← 🗋 →

Single select
TextBesideIcon style

Alouatta caraya	Alouatta guariba
Alouatta macconelli	Alouatta palliata
Alouatta seniculus	Aotus azarae
Aotus vociferans	Ateles geoffroyi
Ateles hybridus br...	Ateles paniscus 2
Ateles paniscus	Avahi laniger
Brachyteles arach...	Brachyteles arach...
Cebus apella	Cercocebus agilis
Cercopithecidae i...	Cercopithecus aet...
Cercopithecus ery...	Cercopithecus mitis

← 🗋 →

Single select
TextBesideIcon style

Alouatta caraya	Alouatta guar...
Alouatta mac...	Alouatta palli...
Alouatta seni...	Aotus azarae
Aotus vocifer...	Ateles geoffroyi
Ateles hybrid...	Ateles panisc...
Ateles paniscus	Avahi laniger
Brachyteles a...	Brachyteles a...
Cebus apella	Cercocebus a...
Cercopitheci...	Cercopithecu...
Cercopithecu...	Cercopithecu...

← 🗋 →

fontBold

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

type	name	bind::ct:content.style	bind::ct:content.fontBold
select_one animal	animal	IconOnly	true

◀ ▶ **survey** choices settings ○

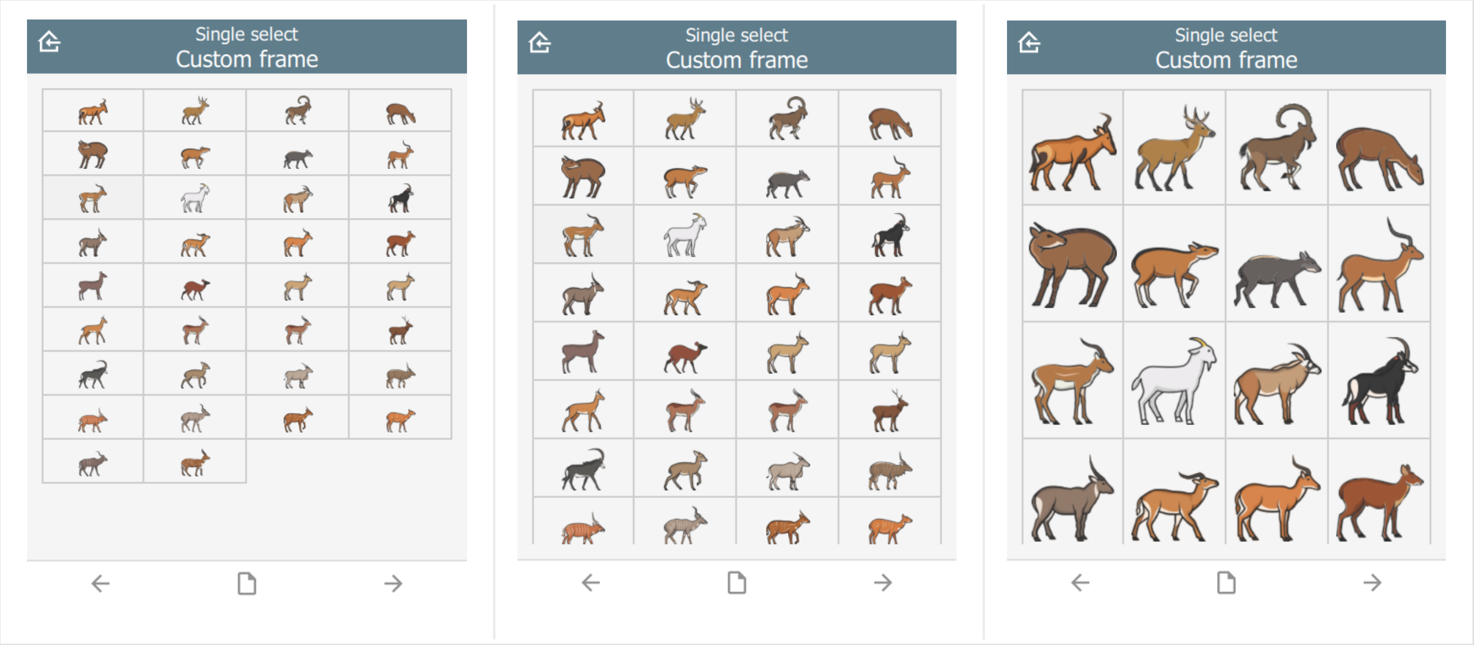
itemHeight

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

type	name	bind::ct:content.style	bind::ct:content.itemHeight
select_one animal	animal	IconOnly	48

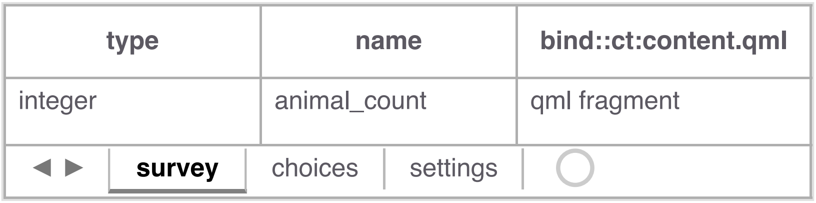
◀ ▶ **survey** choices settings ○

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



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

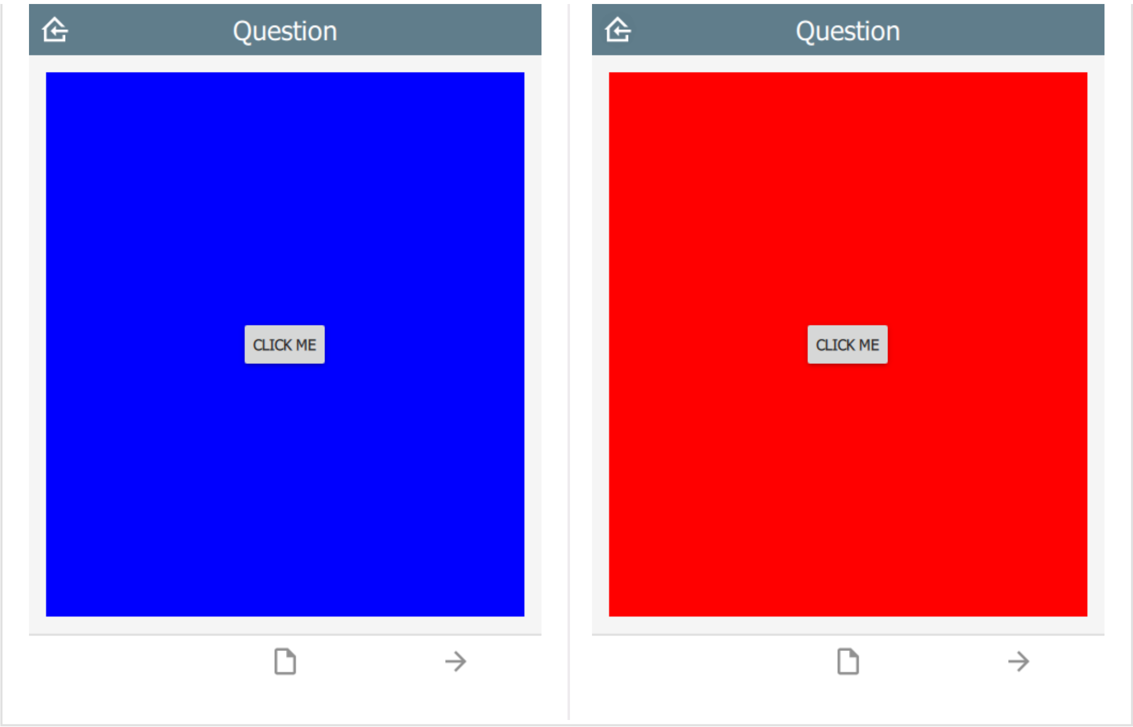
◀ ▶ survey
choices
settings
○

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

```
import QtQuick 2.15
import QtQuick.Controls 2.15

Rectangle {
    color: "blue"
    Button {
        anchors.centerIn: parent
        text: "Click me"
        onClicked: parent.color = "red"
    }
}
```

In the example, the content is blue, then changes to red when the button is clicked.



qmlBase64

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

qmlFile

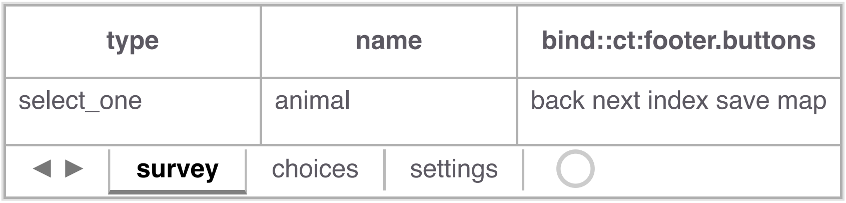
Name of a QML file which exists alongside other project files. This is not supported on ODK or KoBoToolbox, but can be used in Survey123.

Footer

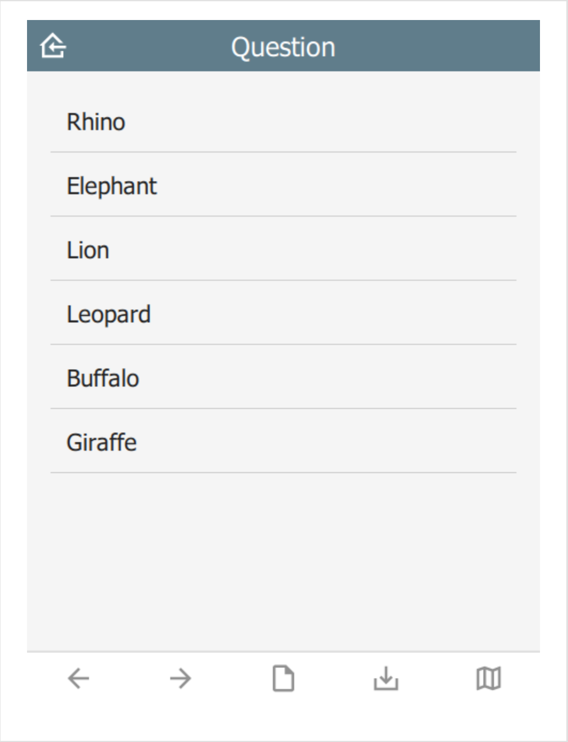
The footer object supports custom control buttons, e.g. home, back, next, save, etc. If no footer object is specified, then the default control is used.

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



home button

The `home` button returns to the **Home** page. In `immersive` mode, this returns to the Projects page, otherwise it returns to the project home page.

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

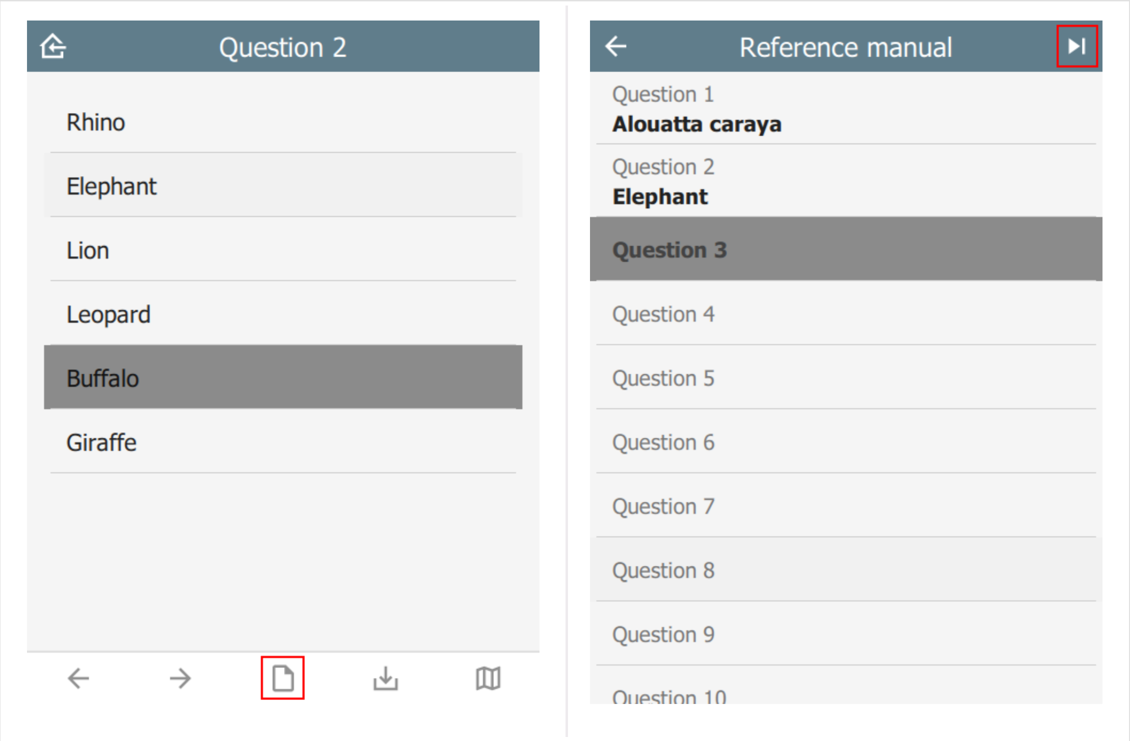
The `save` button will attempt to save the current sighting. If the sighting has invalid data, then the **Index page** will be shown with invalid fields highlighted.

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

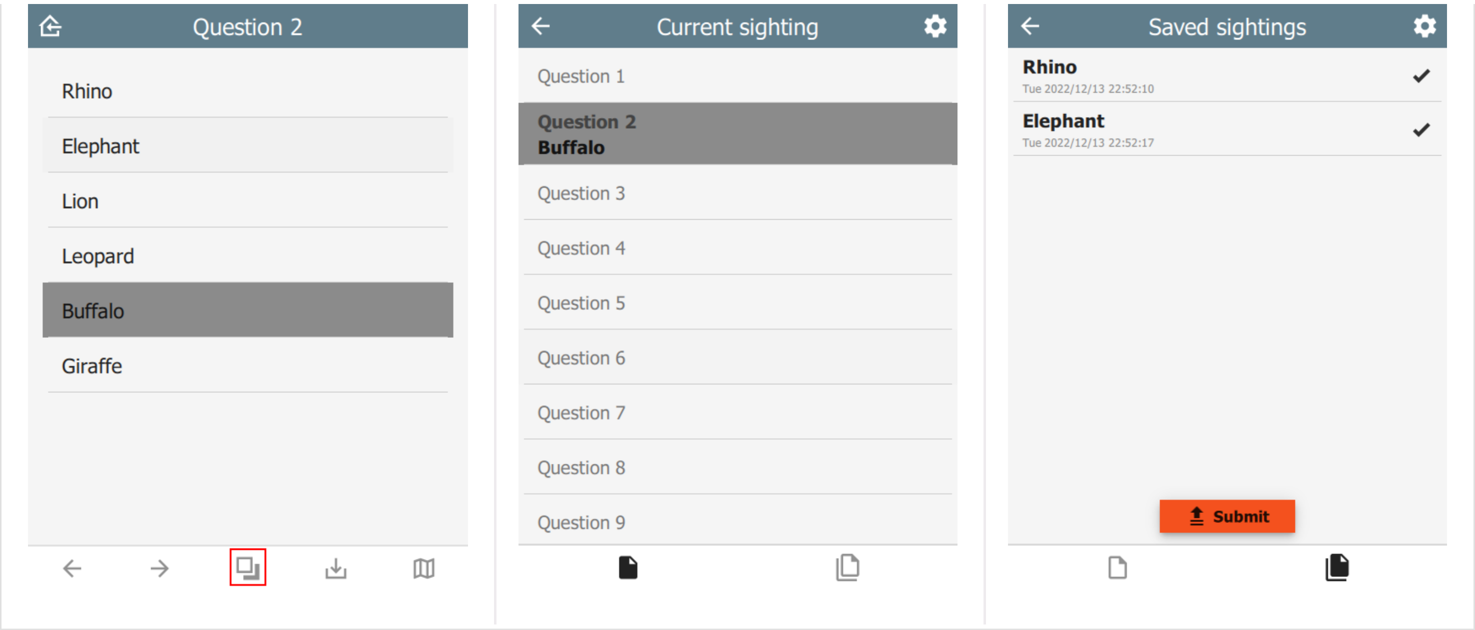
The `index` button displays a list of all the form questions. Selecting a question will navigate the wizard to it directly. A jump-to-last button on the top right of the header will jump to the next required question. If all required questions are filled in, then it jumps to the last question.



options button

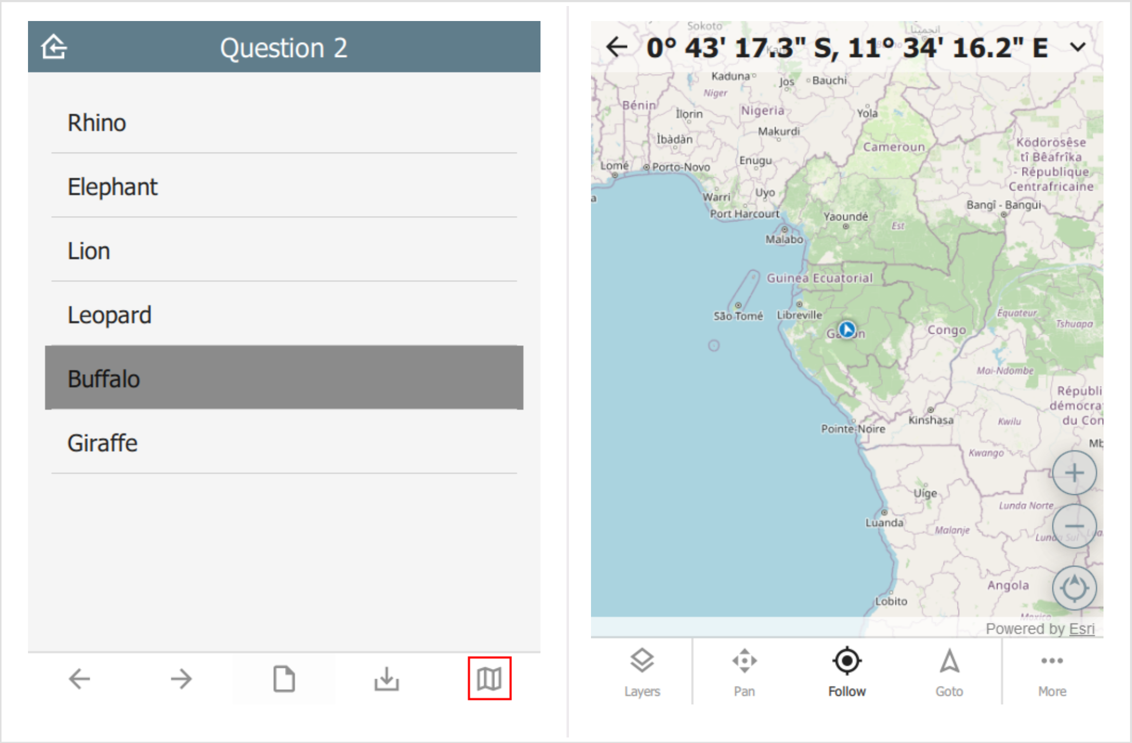
The `options` button is only available in `immersive` mode. In non-immersive mode, it becomes the `index` button (see above).

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



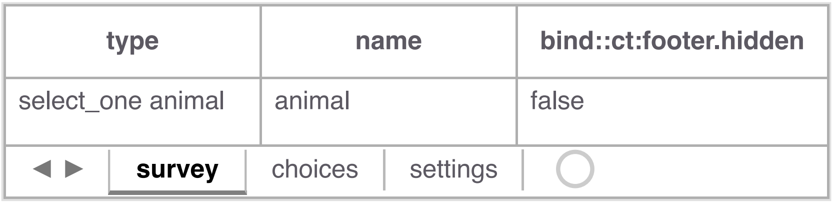
map button

The `map` button opens the map dialog.



hidden

If `yes` then the footer is hidden. Default is `no`.

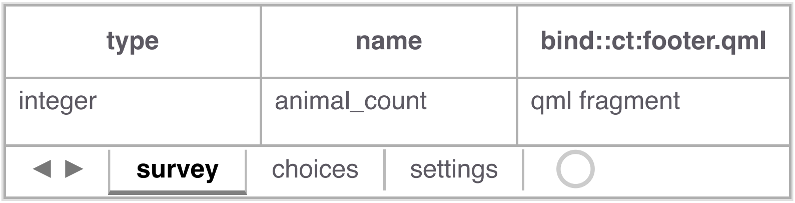


type	name	bind::ct:footer.hidden
select_one animal	animal	false

survey
choices
settings

qml

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



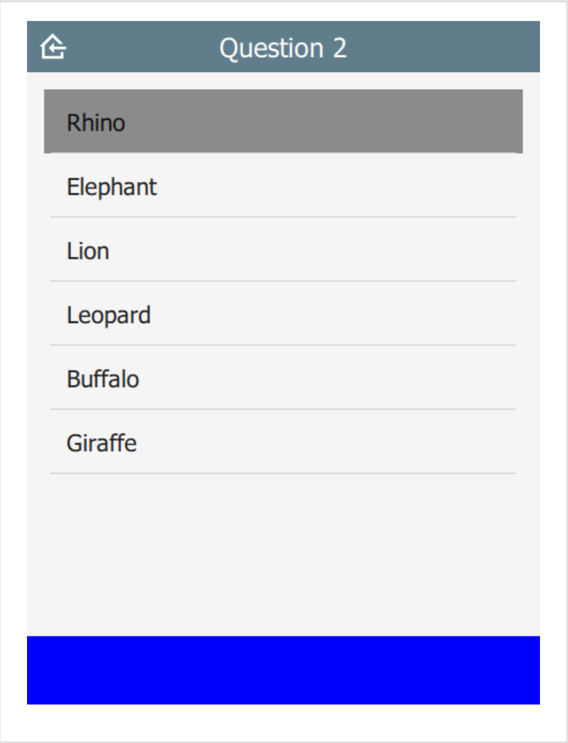
type	name	bind::ct:footer.qml
integer	animal_count	qml fragment

◀ ▶
survey
choices
settings

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.

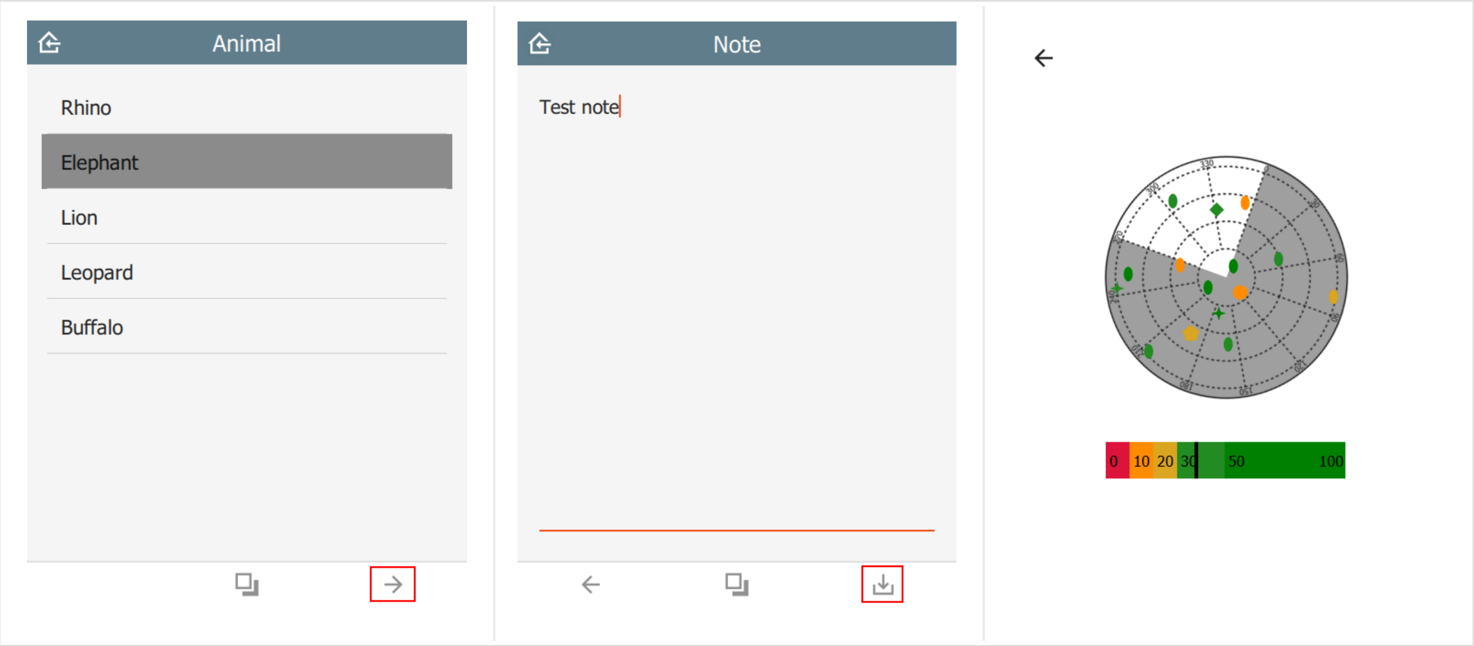
snapLocation

Setting `snapLocation` to the name of a `geopoint` question will create a popup to acquire the GPS location. This feature is only active when `wizardMode` is enabled.

type	name	label
geopoint	f_location	Location
select_one animal	f_animal	Animal
text	f_note	Note
◀ ▶ **survey**	choices	settings ○

title	bind::ct:save.snapLocation
My form	f_location
◀ ▶ **survey**	choices settings ○

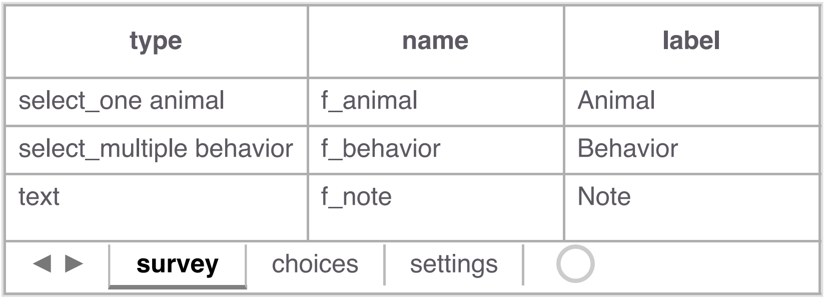
In this example, the user flow will be:



targets

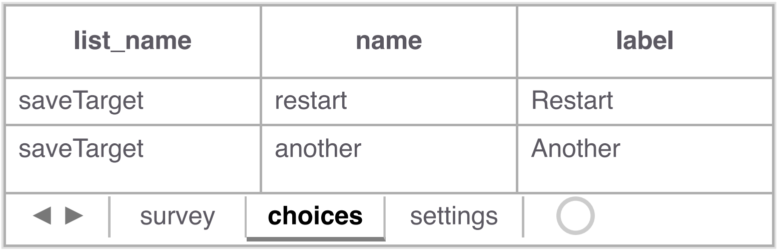
In the example below, the user will be presented with a popup containing the choices **Restart** or **Another**. After the sighting is saved, a new sighting will be created starting at the targeted question. All prior question data will be replicated into the new sighting.

This value must be a valid JSON array.



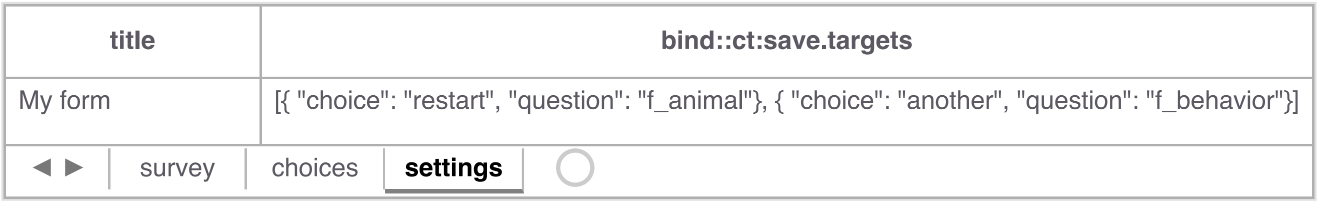
type	name	label
select_one animal	f_animal	Animal
select_multiple behavior	f_behavior	Behavior
text	f_note	Note

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



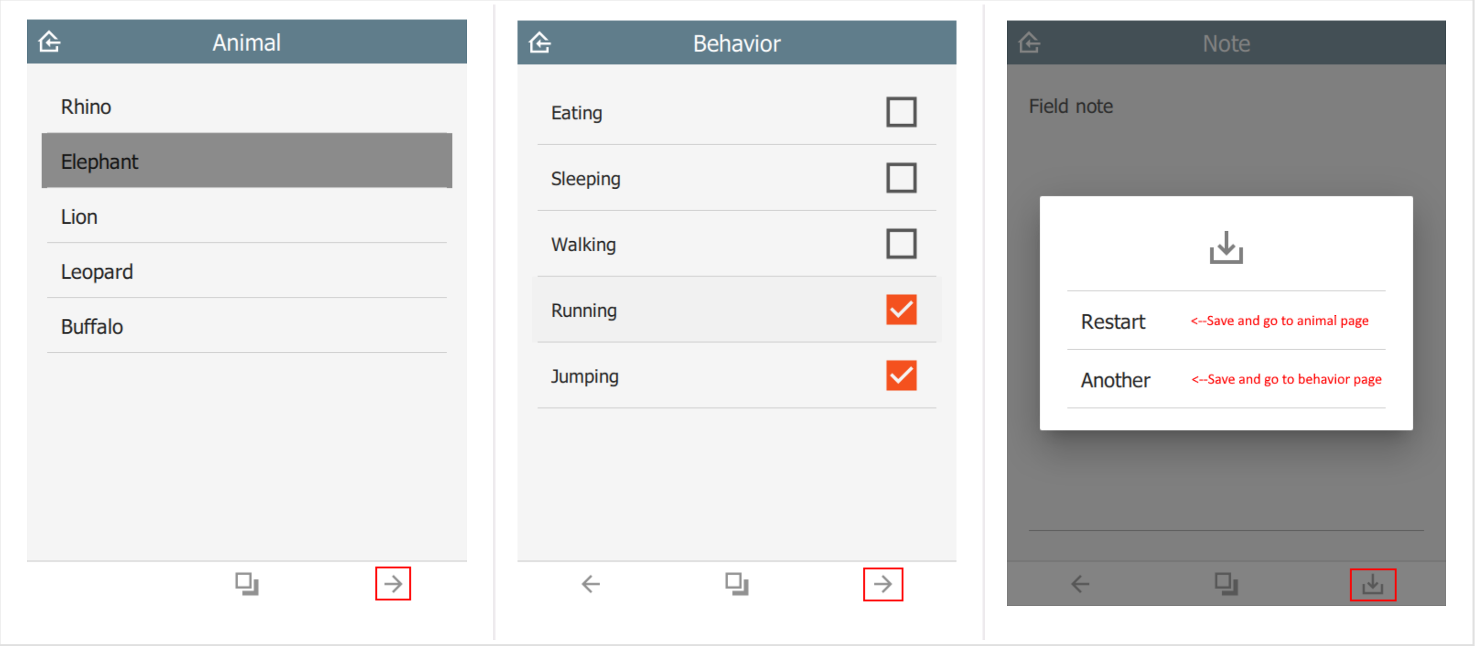
list_name	name	label
saveTarget	restart	Restart
saveTarget	another	Another

Note that `question` is the name of the targeted question in the `survey` table. If the question is not relevant, then this choice will be hidden.



title	bind::ct:save.targets
My form	[{ "choice": "restart", "question": "f_animal"}, { "choice": "another", "question": "f_behavior"}]

Note that if `immersive` is set to false, then the target list will automatically contain the **home** button. This option will save and return to the **Home** page without automatically creating a new sighting.



<--Save and go to animal page

<--Save and go to behavior page

track

In the example below, there is a `select_one` question called `f_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.

This value must be a valid JSON array.

type	name	label
file	f_track_file	
select_one track_items	f_track	Configure track
text	f_note	Note

◀ ▶ **survey** choices settings ○

list_name	name	label
track_items	start	Start
track_items	stop	Stop
track_items	nochange	No change

◀ ▶ survey **choices** settings ○

title	bind::ct:save.trackFile	bind::ct:save.track
My form	f_track_file	[{ "condition": "selected(${f_track}, 'start')", "updateIntervalSeconds": 5, "distanceFilterMeters": 10 }, { "condition": "selected(${f_track},'stop')", "updateIntervalSeconds": 0, "snapTrack": true }]

◀ ▶ survey choices **settings** ○

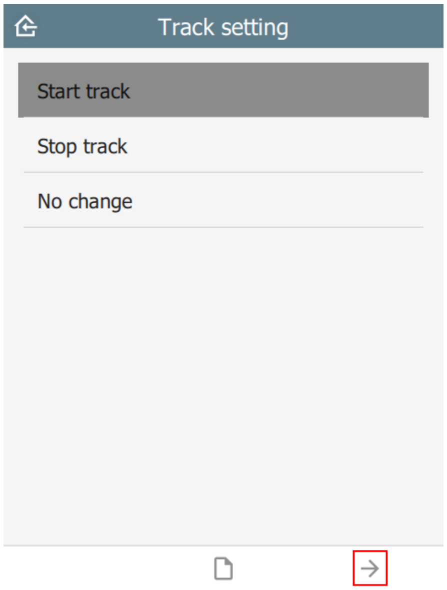
`condition` is an XlsForm expression which activates this option if matched, e.g. ${start_stop}='start'. Check out the ODK Form Logic documentation.

`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 `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.

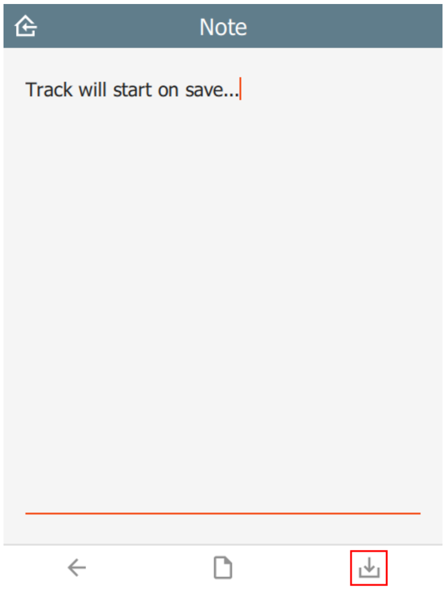


Track setting

Start track

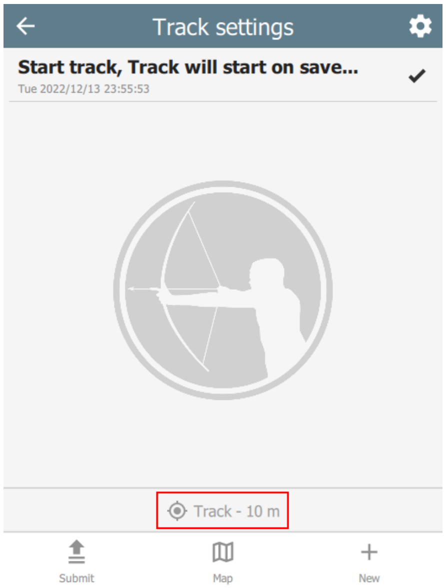
Stop track

No change



Note

Track will start on save...

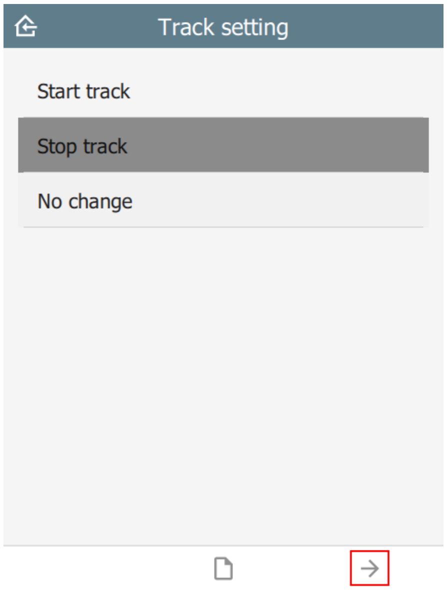


Track settings

Start track, Track will start on save... ✓

Tue 2022/12/13 23:55:53

Track - 10 m

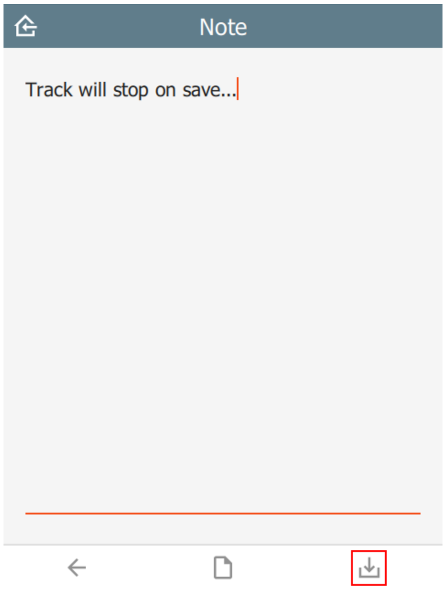


Track setting

Start track

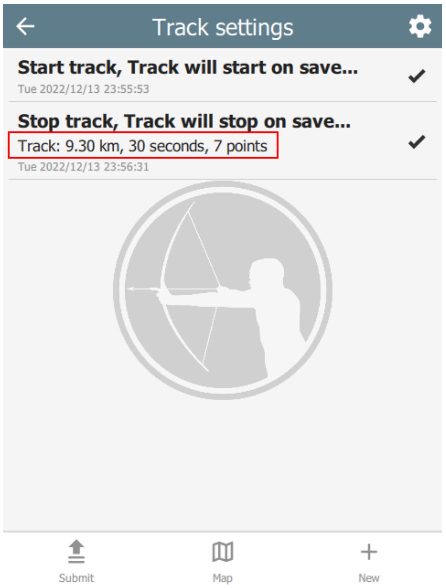
Stop track

No change



Note

Track will stop on save...



Track settings

Start track, Track will start on save... ✓

Tue 2022/12/13 23:55:53

Stop track, Track will stop on save... ✓

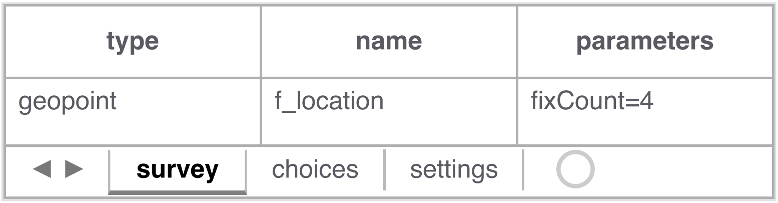
Track: 9.30 km, 30 seconds, 7 points

Tue 2022/12/13 23:56:31

Miscellaneous

fixCount

For `geopoint` question types, the number of skipped readings before a fix is taken. The default value is 4. Some GPS devices return old readings before real readings. To overcome this, setting the `fixCount` will cause the system to require several readings before the final location is taken.



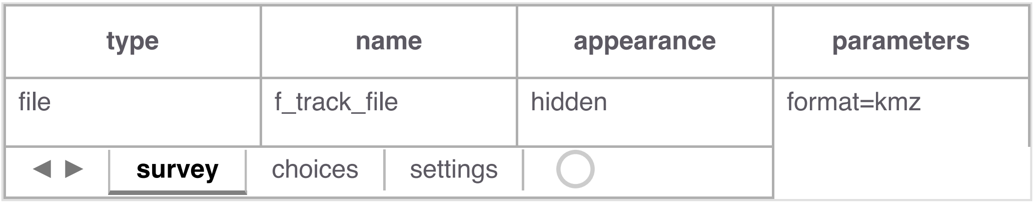
type	name	parameters
geopoint	f_location	fixCount=4

track file format

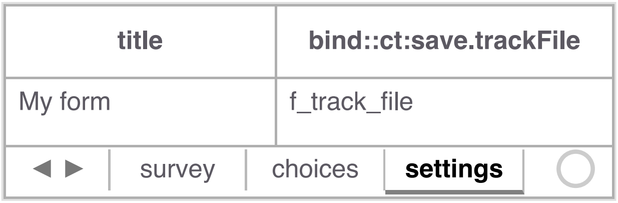
When the user presses **Save** and creates a track file, it is stored in a `file` field specified in the `settings` sheet in the `bind::ct:save.trackFile` column.

By default the format of the track is zipped geojson, but this can be changed by using the `format` parameter of the question itself. Supported values are `geojson` and `kmz` (not supported on Survey123).

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



type	name	appearance	parameters
file	f_track_file	hidden	format=kmz



title	bind::ct:save.trackFile
My form	f_track_file

Frequently Asked Questions

Which backends support XlsForm?

CyberTracker supports ODK Central, KoBoToolbox and Survey123.

Are CyberTracker extensions visible to other tools?

XlsForm extensions support custom columns by using the `namespaces` value in the `settings` sheet. Columns prefixed with `bind::ct:` are only used by CyberTracker and are ignored (but preserved) by other tools.