

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](#)
 - [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](#)
 - [options](#)
 - [map](#)
 - [save](#)
 - [next](#)
 - [hidden](#)
 - [qml](#)
 - [qmlBase64](#)
 - [qmlFile](#)
- [Save](#)
 - [snapLocation](#)
 - [targets](#)
 - [track](#)
- [Miscellaneous](#)
 - [fixCount](#)
 - [trackFileFormat](#)
 - [hidden](#)
- [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	settings	○

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

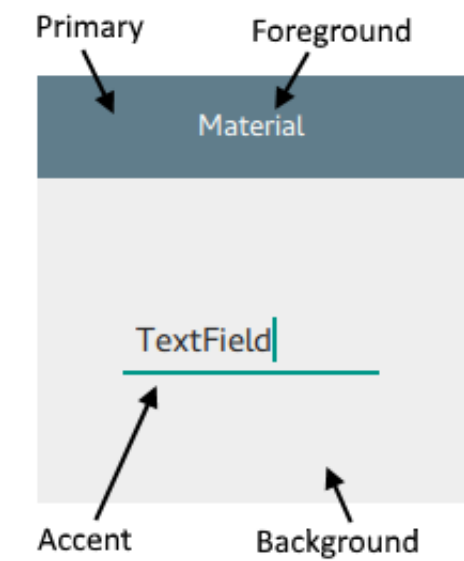
immersive

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
<div><div>◀ ▶</div><div>survey</div><div>choices</div><div>settings</div><div>○</div></div>	

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
<div><div>◀ ▶</div><div>survey</div><div>choices</div><div>settings</div><div>○</div></div>		

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 esriLocationServiceUrl column:

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

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	settings <input type="radio"/>

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
integer	animal_count	Custom title
◀ ▶	survey	choices survey ○

topText

Custom smaller text above maintitle

type	name	bind::ct:header.topText
integer	animal_count	Custom top header
◀ ▶	survey	choices survey ○

button

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

- `track` - the current state of the GPS track system
- `battery` - the current state and level of the battery

type	name	bind::ct:header.button
integer	animal_count	track
◀ ▶	survey	choices survey ○

hidden

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

type	name	bind::ct:header.hidden
integer	animal_count	false
◀ ▶	survey	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
integer	animal_count	qml fragment
◀ ▶	survey	choices survey ○

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
integer	animal_count	16
◀ ▶	<u>survey</u> choices survey	○

style

The visual appearance of the question.

type	name	bind::ct:content.style
select_one animal	animal	IconOnly
◀ ▶	<u>survey</u> choices survey	○

For **select_one** questions

- IconOnly
- TextOnly
- TextBesideIcon
- TextUnderIcon

For **select_multiple** questions:

- IconInlay
- IconOnly
- TextOnly

- TextBesideIcon

For **range** questions:

- Grid

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	4
◀ ▶ survey	choices survey	<input type="radio"/>

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
◀ ▶ survey	choices survey	<input type="radio"/>

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
◀ ▶ survey	choices survey	<input type="radio"/>

border

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

type	name	bind::ct:content.border

select_one animal	animal	true
◀ ▶	survey	choices survey <input type="radio"/>

borderWidth

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

type	name	bind::ct:content.borderWidth
select_one animal	animal	3
◀ ▶	survey	choices survey <input type="radio"/>

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 <input type="radio"/>

fontBold

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

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

itemHeight

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

type	name	bind::ct:content.itemHeight
select_one animal	animal	64
◀ ▶	survey	choices survey <input type="radio"/>

qml

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

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

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

```
import QtQuick 2.15

Rectangle {
    color: "blue"
}
```

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. back, next, save, etc. If no footer object is specified, then the default control is used.

options

Show **options** button in immersive mode. Show **index** button in non-immersive mode. Default is true.

type	name	bind::ct:footer.options
integer	animal_count	true
<div><div><div>◀ ▶</div><div>survey</div><div>choices</div><div>survey</div><div><input type="radio"/></div></div></div>		

map

Show **map** button. Default is true.

type	name	bind::ct:footer.map
integer	animal_count	false
<div><div><div>◀ ▶</div><div>survey</div><div>choices</div><div>survey</div><div><input type="radio"/></div></div></div>		

save

Show save button. Default is false.

type	name	bind::ct:footer.save
integer	animal_count	true
<div><div><div>◀ ▶</div><div>survey</div><div>choices</div><div>survey</div><div><input type="radio"/></div></div></div>		

next

Show next button. Default is true, if there is a next page.

type	name	bind::ct:footer.next
integer	animal_count	false

◀ ▶	survey	choices	survey	○
-----	---------------	---------	--------	---

hidden

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

type	name	bind::ct:footer.hidden
integer	animal_count	false
◀ ▶	survey	choices survey ○

qml

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

type	name	bind::ct:header.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 the `hidden` column on this question, so that the user is not asked for the location beforehand.

type	name	bind::ct:save.snapLocation		
integer	animal_count	true		
◀ ▶	survey	choices	survey	○

targets

In the example below, the user will be presented with a popup containing the choices **Restart** or **Continue**. Depending on which is selected, a new sighting will be created starting at the targetted question.

`restart` and `continue` 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 ignored.

type	name	bind::ct:save.targets		
integer	animal_count	[{ "choice": "restart", "question": "animal"}, { "choice": "continue", "question": "count"}]		
◀ ▶	survey	choices	survey	○

list_name	name	label
saveTarget	restart	Restart
saveTarget	continue	Continue
◀ ▶	choices	choices survey ○

track

In the example below, there is a `select_one` question called `start_stop` with choices `start` and `stop`. 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:save.track
integer	animal_count	[{ "condition": "\${start_stop}='start'", "updateIntervalSeconds": 5, "distanceFilterMeters": 120 }, { "condition": "\${start_stop}='stop'", "updateIntervalSeconds": 0, "snapTrack": true }]
<div><div>◀ ▶</div><div>survey</div><div>choices survey <input type="radio"/></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 the Survey123 provider 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
◀ ▶	survey	choices survey <input type="radio"/>

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	bind::ct:trackFormat	bind::ct:hidden
file	my_track_file	kmz	true
◀ ▶	survey	choices survey <input type="radio"/>	

hidden

The question will be marked as hidden and will not be shown. This is useful for fields which CyberTracker fills in during the save process. For example, the `geopoint` question (see `snapLocation`) and `file` (see `snapTrack`) questions.

type	name	bind::ct:hidden
geopoint	my_location	true
◀ ▶	survey	choices survey <input type="radio"/>

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.