Form Audit Log

Collect can log the behavior of enumerators as they navigate through a form. This log has many uses including discovering:

- questions that take a long time to answer
- how enumerators typically navigate through a form
- enumerators who take a particularly long or short time to answer
- if enumerators were at the correct location when filling out a form
- when, why and who changed answers

This information can inform form design and training or feed into data validation processes.

> ℹ️ **See also**
>
> [Lightweight timestamping](#)

> ⚠️ **Warning**
>
> If using Aggregate, Aggregate 1.5.0+ required
>
> If a version of Aggregate lower than 1.5.0 is used, **audit files will not be saved on the server**.

Enabling audit logging

To enable logging for a form, add a row of `type` `audit` and `name` `audit` in an XLSForm:

survey	
type	**name**
audit	audit

A form may contain at most one row of `type` `audit`.

Location tracking

You may add the location of events to the log. To do this, add the following parameters to the XLSForm. All three parameters are required.

`location-priority`

> *high-accuracy*: The most accurate location provided by the device, regardless of power use.
>
> *balanced*: Block level accuracy (~100 meters). Uses less power than `high-accuracy`.
>
> *low-power*: City level accuracy (~10 kilometers). Uses less power than `balanced`.
>
> *no-power*: No locations will be returned unless another application on the device has requested location updates. Uses no additional power.

`location-min-interval`

> The desired minimum time, in seconds, at which location updates will be fetched by the device.

`location-max-age`

> The maximum time, in seconds, locations will be considered valid by the device. Must be greater than or equal to `location-min-interval`.

survey		
type	**name**	**parameters**
audit	audit	location-priority=balanced location-min-interval=60 location-max-age=120

When location tracking is enabled, ODK Collect requests location updates from Android periodically, with an interval determined by `location-min-interval`. The requests are sent with `location-priority` to ensure Android does not use

When Collect receives the location updates, it stores the locations in a timestamped cache. At the time of an event, Collect checks the cache for locations stored over the last `location-max-age` and returns the most accurate location in the cache.

For the most accurate locations, set `location-priority` to *high-accuracy*. For the most recent locations, use low numbers for `location-min-interval` and `location-max-age`.

> ✏️ 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.

> ⚠️ Warning
>
> Location tracking can be an invasion of privacy. Users of ODK Collect will be informed that their location is being tracked when they open a form with this feature enabled.
>
> Users can control their privacy by disabling location providers in Android, refusing to grant Collect location permissions, or by disabling location tracking of specific forms in Collect.
>
> Disabling location tracking will not prevent users from filling out forms, but these changes are logged as events in the log.

Change tracking

You can enable change tracking so that old answers and new answers will be added to the question events. To do this, add the following parameter to the XLSForm: `track-changes=true`.

survey

type	name	parameters
audit	audit	track-changes=true

Reason for changes

New in version 1.25: ODK Collect v1.25.0

You can add to `track-changes-reasons=on-form-edit` to prompt enumerators to enter a reason before they save changes to a form:

survey

type	name	parameters
audit	audit	track-changes-reasons=on-form-edit

This will prevent filled out forms being edited without a reason being given. If a reason is given the form will be saved normally and the audit log will include a `change reason` event with the reason recorded in the `change-reason` column.

Enumerator identification

New in version 1.25:
ODK Collect v1.25.0

If your form needs a record of the identity of the enumerator you can use `identify-user=true`.

survey

type	name	parameters
audit	audit	identify-user=true

This will cause Collect to prompt the enumerator for their identity before filling out or editing a form instance. In the audit log, a `user` column will be included that will be populated for each event. The enumerator will not be able to fill in or edit the form without entering a non-blank identity.

> ℹ️ Tip
>
> `identify-user` is useful for data collection workflows where devices might be passed between multiple enumerators for data verification or completion.
>
> In cases where a device will only ever used by a single enumerator, it might make more sense to use username metadata. This will write the username to each submission instead of to the audit log.

Viewing audit logs

Central will export a CSV with audits from all submissions if an export is requested for a form with an audit.

Log structure

If a form includes an audit, Collect will create an `audit.csv` file as the form is filled out. The `audit.csv` file has the following structure:

audit.csv

event	node	start	end
question	/data/name	1523403169208	1523403170733

Values in the `event` column represent a particular user action such as opening a form, saving a form, or displaying a question. Possible event types are described in the Event types section.

Values in the `node` column represent the node in the form that the event refers to, if applicable.

Values in the `start` and `end` columns are timestamps represented as the number of milliseconds since midnight, January 1, 1970 UTC. This is known as epoch time and provides a standard way of representing date/time even across timezones. The Timestamps section contains more information about timestamps.

If both location tracking and change tracking are enabled in the log, the CSV will look like this:

audit.csv

event	node	start	end	latitude	longitude	accuracy	ol val
form start		1550615022663					
location tracking enabled		1550615022671					
question	/data/name	1550615022682	1550615097082	37.4229983	-122.084	14.086999893188477	
location permissions granted		1550615068610					
location providers enabled		1550615068665					
location tracking disabled		1550615095914		37.4229983	-122.084	14.086999893188477	
question	/data/age	1550615097082	1550615097655	37.4229983	-122.084	14.086999893188477	
question	/data/name	1550615097656	1550615102351	37.4229983	-122.084	14.086999893188477	Joh
location tracking enabled		1550615099271		37.4229983	-122.084	14.086999893188477	
question	/data/age	1550615102351	1550615107630	37.4229983	-122.084	14.086999893188477	
end screen		1550615107631	1550615109199	37.4229983	-122.084	14.086999893188477	
form save		1550615109199		37.4229983	-122.084	14.086999893188477	
form exit		1550615109199		37.4229983	-122.084	14.086999893188477	
form finalize		1550615109199		37.4229983	-122.084	14.086999893188477	

Values in the `latitude` and `longitude` columns represent the latitude and longitude in decimal degrees. Values in the

`accuracy` column represents accuracy in seconds.

> ✏️ **Note**
>
> Locations will often be repeated in the log. This is because locations are not captured at the time of the event, but rather retrieved from a cache of the most accurate points captured over the last `location-max-age`.

> ✏️ **Note**
>
> Answers will be recorded only if they differ (if the new answer is different than the old one), otherwise, cells should be empty. Answers which contain commas will be surrounded by double quotes.

Event types

The event column of the audit log can have the following values:

Event	Description	Node?	Timestamps?	Coordinates?	Answers?
form start	Start filling in the form	No	`start` only	If enabled and available	No
question	View a question	Yes	Yes	If enabled and available	If enabled
group questions	View multiple questions on one screen (`field-list`)	Yes	Yes	If enabled and available	No
jump	View the jump screen	No	`start` only	If enabled and available	No
add repeat	Add a repeat	Yes	Yes	If enabled and available	No
delete repeat	Delete a repeat	Yes	Yes	If enabled and available	No
end screen	View the end screen	No	Yes	If enabled and available	No
form save	Save the form	No	`start` only	If enabled and available	No
form exit	Exit the form	No	`start` only	If enabled and available	No
form resume	Resume the form	No	`start` only	If enabled and available	No
form finalize	Finalize the form	No	`start` only	If enabled and available	No
save error	Error trying to save	No	`start` only	If enabled and available	No
finalize error	Error trying to finalize the form (probably encryption related)	No	`start` only	If enabled and available	No
constraint error	Constraint or required error on finalize	No	`start` only	If enabled and available	No
location tracking enabled/disabled	Toggle location tracking in Collect	No	Yes	If enabled and available	No
location providers enabled/disabled	Toggle location providers in Android	No	Yes	If enabled and available	No
location permissions granted/not granted	Toggle location permission in Android	No	Yes	If enabled and available	No

Timestamps

If we relied entirely on the time reported by the device for timestamps, users or the network could change the device time and manipulate the correctness of the audit log. For this reason, we only use device time for the form start timestamp. All subsequent event timestamps are the result of elapsed time, which users cannot change, added to the form start timestamp. This means that while the timestamps themselves may potentially be inaccurate, the time elapsed within and between the timestamps are always accurate within one form editing session.

Using epoch time makes it easy to compute elapsed time by subtracting start from end. For example, given the following log:

audit.csv

event	node	start	end
form start		1488761807863	
question	/data/name	1488761807868	1488761809157

The enumerator spent `1488761809157 - 1488761807868 = 1289` milliseconds on the screen showing the `/data/name` question. This corresponds to `1289 / 1000 = 1.289` seconds.

To convert from epoch time to time in UTC in most common spreadsheet programs, divide the epoch time by 86400000 ms per day and add 25569 days between January 1, 1900 (what spreadsheet programs use as "day zero") and January 1, 1970. For example, to convert the timestamp `1488761807868`:

```
(1488761807868 / 86400000) + 25569 = 42800.03944
```

When the cell is set to type `date time` in common spreadsheet programs, it will show `3/6/2017 0:56:48 UTC`. A common workflow if device time is needed in a human-readable format will be to add a column for the calculation above and change that column's type to `date time`.

Known limitations

- If the device is turned off while a form is being filled, Collect will not record a log entry for the screen that was shown at the time of device shutdown. Events before and after the shutdown will be logged.
- Editing a saved form that was saved using different audit log options can result in a corrupt audit. It might take place when a user saves a form then updates a form definition (changing audit log options) and tries to edit the saved form.