

Moved over to under Kobo ownership [here](#) as of Mar 25, 2025. See below the “archived” version of the document. This document is safe to delete.

Differences between Collect and Enketo

ODK Collect started out as the only ODK client and continues to be the reference implementation. Enketo was carefully developed by [Martijn van de Rijdt](#) to match Collect’s behavior. As part of that effort, Martijn wrote up [the more formal spec](#) we now all use and maintain. Still, there are some deviations between the two clients, some intentional and some not. ODK is now maintaining Enketo and would like to redress or more clearly document the deviations.

Differences that would be good to redress	2
(FIXED) Non-relevant values are used in Enketo expressions	2
Read-only is dynamic in Collect, static in Enketo	2
Nested repeats are not supported well in Enketo	2
Markdown, HTML support	2
Performance of secondary instance queries is different	2
Differences we will try to prioritize	2
Audit log is not supported in Enketo	2
Enketo is single-page by default, also has a grid theme	3
Repeat UX is very different	3
Value clearing UX is very different	3
Accuracy threshold attributes are not supported in Enketo	3
Relevance inheritance from model for Collect and body for Enketo	3
Enketo uses browser language to set form language, Collect doesn’t use device language	3
Differences there is no intention to address	3
search() appearance/function is only supported in Collect	3
“Fast external itemset” is only supported in Collect	4
To triage	4
When repeat_count goes down, Collect keeps extra instances	4
Functionality in Collect not Enketo	4
Widgets in Collect that are different in Enketo	4
Functionality in Enketo that is not in Collect	5
Guidance hint ux is different	5
Whitespace in CSV external secondary instances	5
Attempt to get single child node from nodeset	5
Integers	5

Differences that would be good to redress

(FIXED) Non-relevant values are used in Enketo expressions

- As of [EE 4.0.0](#), the [exclude non-relevant configuration](#) lets users opt into matching Collect's behavior. It will be turned on by default in Central v2022.3
- Collect: non-relevant values are treated as blanks
- This most impacts relevance expressions in decision trees
 - Relevance conditions for decision tree forms that work in Enketo have to be much more complex or use trigger to clear prior decisions
- Implementation complexity and performance concerns for Enketo
- <https://github.com/enketo/enketo-core/issues/849>

Read-only is dynamic in Collect, static in Enketo

Nested repeats are not supported well in Enketo

- Lots of various bugs
- It may not be practical to collect nested repeat data in Enketo but it's necessary for edits from Collect

Markdown, HTML support

- Both support markdown relatively well but with some quirks (e.g. [with links](#))
- Collect supports HTML which allows for much richer formatting
- Linebreaks in Enketo are a particular painpoint (can't use `
`)
 - Using spreadsheet newlines results in a single linebreak in Enketo and a paragraph in Collect
- Dynamic values can't be formatted in Enketo (e.g. if `*foo*` is in an external secondary instance it will be shown in bold by Collect but not Enketo)

Performance of secondary instance queries is different

- Now faster in Collect; similar caching could happen in Enketo

Differences we will try to prioritize

Audit log is not supported in Enketo

- Audit logging is most relevant in enumerator-mediated contexts
- In Enketo pages mode, it could be implemented the same way as Collect

- In Enketo single-page mode, it's not clear what it should log
- Diffs and reasons for change are [captured by Central](#)
- <https://github.com/enketo/enketo-core/issues/669>

Enketo is single-page by default, also has a grid theme

- Collect only has multi-page mode; some questions can be grouped in field-lists
- Lots of demand for Collect to eventually have something grid-like as screens get bigger

Repeat UX is very different

- Probably good that they are different because they're used in different contexts but there usability tradeoffs

Value clearing UX is very different

- Collect always allows clearing through long press (not discoverable)
- Enketo sometimes has (x), sometimes no special way to clear value

Accuracy threshold attributes are not supported in Enketo

- <https://github.com/enketo/enketo-core/issues/367>

Relevance inheritance from model for Collect and body for Enketo

- <https://github.com/getodk/javarosa/pull/659>

Enketo uses browser language to set form language, Collect doesn't use device language

No-calendar appearance is not available in Enketo

Differences there is no intention to address

search() appearance/function is only supported in Collect

- Not in ODK docs, not recommended for use
- Do not expect to ever support in Enketo

“Fast external itemset” is only supported in Collect

- Not in ODK docs, not recommended for use
- Do not expect to ever support in Enketo

To triage

Enketo sometimes writes NaN to the model whereas Collect always writes blank (e.g. “ * 30)

When repeat_count goes down, Collect keeps extra instances

- <https://github.com/getodk/javarosa/issues/45> – when H first joined ODK, heard about this as a deliberate choice to prevent data loss. Not so convinced anymore, maybe we should revisit.

Functionality in Collect not Enketo

- submission_url
- meta subscriberid
- meta simserial
- meta phonenumnumber
- meta email
- select_one_external
- background-audio
- barcode
- select_one/select_multiple quick
- select_one/select_multiple quickcompact
- text printer
- integer/decimal bearing
- text/integer/decimal thousands-sep
- range rating
- date/datetime no-calendar
- date/datetime ethiopian
- date/datetime islamic
- date/datetime coptic

Widgets in Collect that are different in Enketo

- date year
- geo* maps
- Geopoint – Collect has a managed experience
- Select multiple slight difference: Collect stores the selected options in the order they were selected, Enketo in the order that they were displayed

Functionality in Enketo that is not in Collect

- text multiline
- text distress
- geo* hide-input
- geo* streets, terrain, satellite

Guidance hint ux is different

Acknowledge text is different

Enketo dispatches xforms-value-changed events for calculations

Or at least for fields with a bound control that have a calculation

Whitespace in CSV external secondary instances

Enketo strips whitespace around tokens, Collect does not

Web Forms tests:

<https://github.com/getodk/web-forms/commit/6b1f1454749c6ab5226d3ba6654d15f254639645#diff-0ae498d66cc2114687268adaea646f23d17985f3209cc8596693a1e55251eeffR918>

Attempt to get single child node from nodeset

Collect crashes with error about using indexed-repeat. This is annoying and doesn't follow XPath rules but is generally more helpful than silently returning the child for the first node in the nodeset as Enketo does

Integers

XML defines int as being between 2147483647 and -2147483648 inclusive. Collect only allows 9 digits. Enketo allows any number of digits but at some point switches to using scientific notation.

More at <https://xlsform.org/en/ref-table/>