

XLSForm

XLSForm is a standard for building forms in Excel. XLSForms are simple to get started with and can represent complex forms.

XLSForms can be created and edited by any application that works with `.xlsx` documents. This means that they are portable and can leverage many useful features commonly available in spreadsheet applications such as formulas, comments and document versioning.

Many users choose to use Google Sheets or Excel for the web so that they can collaborate on or publicly share their forms.

One way to get started is by downloading or making a copy of the form definition template below.

- [🔗 Form definition template](#)

You can hover over columns to see guidance about how to use them. The [All Widgets form definition](#) shows examples of all of the different user-visible question types.

The ODK documentation shows all form design examples as XLSForms and describes how XLSForms are used by ODK tools. There is also a general tutorial for building XLSForms on the [XLSForm website](#).

Once your form has been designed, you can [upload the XLSForm directly to Central](#).

Tip

If your ODK server does not have the latest XLSForm features or you need to temporarily preview a form in a browser, try [XLSForm Online](#).

If you need to design XLSForms offline or your form has sensitive data that you'd rather not upload into XLSForm Online, use the [pyxform](#) command line tool.

The survey sheet

At minimum, an XLSForm has a sheet named **survey** to describe the types and order of fields in a form. It must have these three columns:

- `type`: the type of the field represented by each row. Supported types and how they are displayed are described [here](#).
- `name`: the name of the field represented by each row. This name will be used in your data results. It may not contain spaces and must start with a letter or underscore. Use a short and descriptive name. For example: `date_of_birth`.
- `label`: the user-visible question text for the field represented by each row. For example: `When was ${first_name} born?`
This text can [reference other fields](#) or [have translations](#).

The survey sheet can have many other columns to represent different [question types](#) and [form logic](#). You can see the most commonly-used columns in [this template](#).

The settings sheet

You should also include a **settings** sheet to uniquely identify your form definition and its current version. We recommend specifying at least the following columns:

- `form_title`: The title that will be displayed by tools that list this form.
- `form_id`: The unique ID that identifies this form to tools that use it. It may not contain spaces and must start with a letter or underscore. Use a short and descriptive name. For example: `bench_inventory_2021`.
- `version`: The unique version code that identifies the current state of the form. A common convention is to use a format like `yyyymmddrr`. For example, `2017021501` is the 1st revision from Feb 15th, 2017.
- `instance_name`: An [expression](#) that will be used to represent a specific filled form created from this form definition. For example, `concat(${first_name}, ${age})`. [Learn more](#).

The **settings** sheet is also useful when using [multi-language forms](#) or when defining a form with [encryption](#).

Naming filled forms

Looking to save time? Try [ODK Cloud](#), the official managed hosting service from the creators of ODK.

form's contents in its name. This name will be shown in several places to help guide data collection and analysis. You should pick a name that uniquely identifies the filled form and the data it had captured. For example:

- If a single filled form represents data about a real-world thing like a person or park bench, your `instance_name` expression should include some information to uniquely identify the thing like the person's name or the park bench's location and current status.
- If a single filled form represents data about an observation, consider including the date and time of the observation in the `instance_name` expression.
- If your form definition includes a repeat, consider including the repeat count in the `instance_name` expression.

Filled form names in Collect

Each filled form is identified by its `instance_name` value in Collect's [Edit Saved Form](#), [Send Finalized Form](#) and [View Sent Form](#) lists.

In workflows where forms have to be filled in multiple different steps, a useful `instance_name` expression will make it much easier to find which filled form to edit. If forms only have to be edited under certain conditions (e.g. not all household members were available), you can include this status in the `instance_name`.

In the [View Sent Form](#) list, `instance_name` can be helpful to identify which data collection tasks have been completed. For example, if a data collector needs to interview 25 specific people and the `instance_name` for each filled form identifies the respondent, they can go to [View Sent Form](#) to verify which subset of interviews they have already completed.

A sent form's `instance_name` is maintained after it is deleted. This makes it possible to confirm what work has been completed even if submissions are configured to [delete after send](#). However, it does mean sensitive data should be avoided in `instance_name`.

The `instance_name` is also used to identify filled forms in Collect's [filled form map](#).

Filled form names in Central

Each submission in Central has its own [detail page](#) which provides basic information about the submission, an activity history of action and discussion on that submission.

The title at the top is pulled from the `instance_name` and it makes navigation much easier to have friendly names at the top of the page and in the web browser title and tab.

The choices sheet

If you have [multiple choice questions](#), you will also need a **choices** sheet to specify choices for those questions. It must have these three columns:

- `list_name`: The unique ID that identifies a group of choices. It may not contain spaces and must start with a letter or underscore. Use a short and descriptive name. For example: `yes_no_maybe`.
- `name`: the name of the field represented by each row. This name will be used in your data results. It may not contain spaces and must start with a letter or underscore. For example you might use `1` or `y` for Yes and `-1` or `n` for No.
- `label`: the user-visible text for the choice represented by each row. For example: `Yes`, `No`, and `Maybe`. This text can [reference other fields](#) or [have translations](#).

Choices with the same list name are considered part of a related set of choices and will appear together for a question. This also allows a set of choices to be reused for multiple questions (for example, yes/no questions).