

Using Repeat Groups in the Form Builder

Overview

In some forms you may want to repeat a set of questions. For example, when registering a mother you may want to ask the number of children then repeat a set of questions for those children.

1. Add a Repeat group to your form
2. The Repeat group has a logic property called *Repeat Count*. Set this value to the number of times you'd like the question to repeat. To do this, point to the value of another question that asks about the number of children a person has(ex. /data/num_children). If you'd like to calculate the number of times to repeat, add a hidden value that does the appropriate calculation and then reference that question in the Repeat Count

- [Overview](#)
- [Controlling the Number of Times a Group is Repeated](#)
- [Logic Within the Repeat Group](#)
 - [Using the current\(\) function](#)
 - [Example](#)
- [Model Iteration inside a Question List](#)
- [Logic Outside The Repeat Group](#)
- [Removing Repeat Groups when using counts](#)
- [Removing Repeat Groups](#)

Controlling the Number of Times a Group is Repeated

Repeat groups can be either be configured to repeat a fixed number of times or a user chosen number of times.

Option 1 - Fixed Number of Times

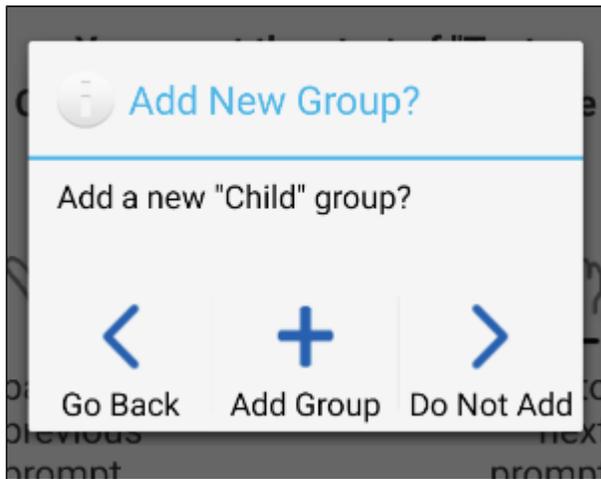
Add a question or hidden value that determines the number of times repeated. Then drag that question into the Repeat Count setting for the repeat group question. **Note:** you cannot just type a number into this box. You must drag a hidden value or question into it.

Note: If the repeat group is inside of a question list, you need to configure it to repeat a fixed number of times.

The screenshot shows the CommCare Form Builder interface. On the left, a question list is visible with the question "How many patients scheduled to visit?" and a sub-question "Repeat Group" containing "Patient Name", "Patient Age", and "Patient Address". The main panel shows the configuration for the "Repeat Group". The "Display Text" is "Repeat Group" and the "Question ID" is "repeat_group". Under the "Logic" section, the "Repeat Count" is set to "1 quantity_of_patients".

Option 2 - User Controlled

By default, CommCare will prompt the mobile worker asking if they would like to go through the repeat again after they have completed all the questions in the repeat group.



Logic Within the Repeat Group

Repeat groups can have multiple questions with the same path (ex. `/data/child_repeat/name`). For this reason, special functions are required when writing logic within the repeat group.

- `..`: Using two dots (`..`) will allow you to access the *parent* of the current question. For example, if you were writing logic for the question *name* in the example above, typing `..` would let you access the `/data/child_repeat` in logic. This can then be used to access another question within the same repeat. For example, `../age` will let you access the question *age* within the repeat.
 - Use `..` when writing logic between questions in the repeat. For example, you may only want to ask a child's date of birth if its known. If so, your display condition may be something like `../child_dob_known = 'yes'`
 - Use `..` if you'd like to display the value of another question in the label. So instead of having `<output value="/data/child_repeat/name" />`, instead use `<output value="../name" />`
 - You may need to repeat `..` multiple times if you have groups within your repeat. For example, if you writing logic from a question within your group, you may need to use `../age`
- `position()`: The position function placed in a calculate condition of a hidden value will return the position in the repeat group that the question is in. It will tell you whether you're on the first repeat, second repeat, etc. You must use the position function on the `repeat_group` question. For example, you could do something like this in a label (assuming the label is not within another group in the repeat): *Please enter the name for Child* `<output value="position(..)"/>`
 - The `position()` function is *zero-indexed* - this means that the first item will be 0, the second item will be 1, etc.
 - For example, if I am on the first "child" in 4 children `position(..)` would return 0.
 - If you want to start with the number "1" or "2" you would input `position(..) + 1`, and `position(..) + 2`, respectively
 - **Warning:** Using the position function in this way is non-standard and has tricky behavior with nested elements and "conditional" usage. In almost all cases, **you can avoid issue by using the position() function in the default value of a question, instead of in the calculate condition.** This will not change the behavior of the function in any meaningful way for the use cases suggested here.
- `current()`: This will return the question. Its needed if you're writing logic in the repeat for something like a lookup table.
- **Note on an important limitation of Repeat Groups:** It is currently not possible for CommCare to delete a node within a repeat group once it has been created. This is important for implementation best practices in the following way: For questions where the repeat count is controlled by a user-input question, if the user enters an initial value and then changes it to something smaller (so from 2 to 1 for example), 2 repeat nodes will still show up. In order to prevent this inconsistency, you can do the following:
 - Add a group question type immediately within your repeat group, and place all of the inner contents of the repeat group inside that inner group.
 - Put this display condition on the inner group: `position(..) < /data/repeat_count` (where `data/repeat_count` is the integer question that defines the repeat count)

Using the current() function

When you are referencing values inside of a repeat by default the references you make will refer to the "relative" questions inside of the group, not in the questions in the group before or after the current one.

When using lookup tables or other constructs that make use of "predicate filters" (IE: expressions with brackets like `locations[@id = "home"]`), the use of the `current()` function is necessary to reference the questions in the current group. This is because inside of the `[]`s, a relative reference (like `./sibling`) is relative to the element which is being filtered, not the question.

The `current()` function translates to essentially mean the same thing as the question itself, after which two dots can be used to make a relative reference to that question.

Example

Inside of a repeat named **registration** you have answered a question **currently_enrolled**, and want to use that answer to filter a lookup table question named **facility** within the repeat. The lookup table question is a "sibling" of the **currently_enrolled** question, meaning they are both inside of the **registration** repeat directly, and not inside of another group. The lookup table will filter facilities based on whether a field on the lookup table (**is_ currently_enrolled**) is set to 'yes'.

In the lookup table's filter, you would set the value to be

is_ currently_enrolled = current()../currently_enrolled

this is needed because within the filter, if we had tried to reference **currently_enrolled**, it would refer to a field in the lookup table with that name, and not to a question in the form.

the **current()** function starts a reference which refers to the current **facility** question, so the **../** step moves the reference back to the active **registrati on** repeat group, and the **currently_enrolled** step moves the reference to the sibling node.

Model Iteration inside a Question List

(This section assumes you are familiar with [using Repeat Groups for Model Iteration.](#))

As mentioned above "If the repeat group is inside of a question list, you need to configure it to repeat a fixed number of times."

This makes model iteration inside a question list a little more complicated to set up, because you cannot use a "Model Iteration ID Query". You need a hidden value outside the question list to store the number of models to iterate, a hidden value inside the repeat group to store the current position, and a question to fetch each item from the models. Let us step through that.

Imagine we had a lookup table whose items we wanted to iterate. Here is a lookup table called "bunnies", with items "Flopsy", "Mopsy", "Cottontail" and "Peter".

Viewing Table 'bunnies'

Select Table to View

field: name	field: sex	field: year_of_birth
Cottontail	F	1902
Flopsy	F	1902
Mopsy	F	1902
Peter	M	1902

Showing 1 to 4 of 4 entries 10 per page < Previous 1 Next >

And here is a form that iterates them inside a question list. The relevant fields and respective values needed for this set up can be seen in the table below:



Display Text	Question ID	Question Type	Calculate Condition	Default Value	Repeat Count
repeat_count	repeat_count	hidden value		count(instance('bunnies')/bunnies_list/bunnies)	
Bunnies!	question_list	question list			
repeat_group	Repeat Group	repeat group			#form /repeat_count
position	position	hidden value	position(..) + 1		
bunny	bunny	hidden value	instance('bunnies')/bunnies_list /bunnies[current()/../position]/name		
<output value="current()/../position" /> - <output value="current()/../bunny" />	bunny_label	label			

The first question in the form is a hidden value with the self-descriptive name **repeat_count**, whose Default Value is set to "**count(instance('bunnies')/bunnies_list/bunnies)**".

The second question is our question list. Inside the question list is the repeat group. Its Repeat Count is set to the first question, **"/data/repeat_count"**.

Inside the repeat group is a hidden value named **position**, with a Calculate Condition set to "**position(..) + 1**".

So the next question in the repeat group is another hidden value called **bunny**, with a Calculate Condition of "**instance('bunnies')/bunnies_list /bunnies[current()/../position]/name**".

The last question is a label to show that we have achieved what we set out to do. Its Display Text is "**<output value="current()/../position" /> - <output value="current()/../bunny" />**".

Logic Outside The Repeat Group

As a given path could point to multiple questions, we can use special functions to access a specific repeated value. Using something like **/data/child_repeat/name** outside of a repeat group will lead to an error.

XPath nodeset has more than one node [...] Cannot convert multiple nodes to a raw value. Refine path expression to match only one node.

To resolve, you must modify the expression so it gives a single value. Here are some common approaches:

- **Access A Question By Count:** You can access a specific question (ex. the second) using square brackets. For example, **/data/child_repeat [1]/name** will access the name of the first repeated child (numbering starts at 1). Note that this does not work with **easy references**, only with the text-based **/data/child_repeat[1]/name** style of references.
- **Join all values into one:** You can combine the values into one string using the **join** function. For example, **join(", ", /data/child_repeat/name)** will join together each name, separated by a comma and a space.
- **count(question_path):** The count function will give you the number of times a particular question was repeated. For example, **count(/data /child_repeat)** will give you the number of times the repeat group was repeated.

- Filtering By Question Answer:** Its also to access a specific question based on an answer provided in the repeat. This is done by using a filter in square brackets. For example, to count the number of children who are female, you can use the following: `count(/data/child_repeat[gender = 'female'])`. You can replace *gender* with the question you'd like to check. Its also possible to use multiple square brackets to filter. For example, `/data/child_repeat[gender = 'female'][age > 3]/name[0]` will provide the name of the first female child who is over 3 years old. (Note, that this will cause an error if there is no female child who is over 3 years old).

Generally if a given question path can point to multiple results, you can use square brackets to filter that set of results or choose a specific answer by count.

Removing Repeat Groups when using counts

When using repeat counts, it's possible to configure the form to "delete" repeat groups added if the count changes. (IE: Repeat count was originally 5 and is now 2)

For an example: Upload this XForm to CommCareHQ

[Simple Repeat _Delete_ Example.xml](#)

Removing Repeat Groups

In apps with user controlled repeat counts, sometimes a user will accidentally create a repeat group which is not needed.

Physically deleting a repeat group can create confusing or ambiguous behavior in a form, and potentially error prone, since the action can't be undone. By default the mobile app prevents users from creating repeat groups.

Instead, if that behavior is needed, it can be introduced by placing the questions inside of a repeat into a **group**, and using the group's **Display Condition** to eliminate the questions inside of the repeat. This lets users effectively cancel the repeat while also making it clear how to undo that action, and allowing the form itself to decide how to handle any awkwardness around removing the repeat data.

