Beiwe Research Platform: Survey Branching

Last updated: March 26, 2023

Survey Branching

This document is intended to explain how to implement survey branching in your Beiwe studies, if desired.
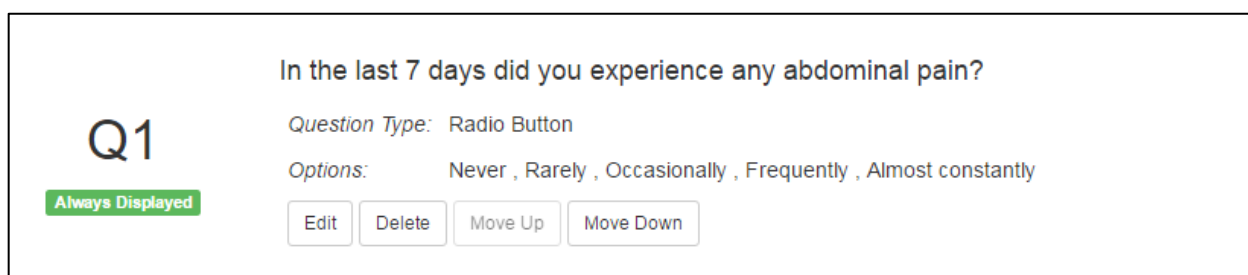
Setting up a survey with branching

The Beiwe backend user interface (studies.beiwe.org) is easy to use and helps guide you in setting up your conditional questions. Start with one or more questions that will always be shown and whose answers can serve as the basis of future survey branching logic that you'll configure. **Any questions that you would like to use to determine branching logic must have a definitive answer (Boolean output)**, thus you *can* use any of the following question types:

- slider questions,
- radio button questions, and
- numeric open response questions.

Beiwe's remaining question types, such as information or checkbox, do not result in a unique outcome and thus *cannot* be used for branching decisions. However, they can be used in the normal flow of your surveys without being referred to later. Set up your questions as you always have:

- Enter your question text
- Choose a question type (as noted above, depending on how you're using the question)
- Add your answer choices.
- Press "**Add Question**" to save

Questions in surveys are designated as "Always Displayed" or "Conditionally Displayed."

As you add your answer choices to a radio button survey question, note that a number is now noted to the left of your list of the choices, as shown in the screenshot below. If you'll be using this question for survey branching, take note of these numbers to use later on when you set up your branching logic.

Edit Question   ✕

Question   In the last 7 days, how much did your Abdominal Pain interfere with your usual or daily activities?
Text:

Question   Radio Button   ▼
Type:

Answer   Add new answer choice
Choices:

        0:   Not at all   ✕

        1:   A little bit   ✕

        2:   Somewhat   ✕

        3:   Quite a bit   ✕

        4:   Very much   ✕

If you're using a slider or numeric open response question type, the specific number entered on either can be used for your subsequent conditional questions.

"Always Displayed" and "Conditionally Displayed" questions can be used as the basis for survey branching. To set up the conditions for displaying a "Conditionally Displayed" question, the following four operator blocks are available:

- Add "NOT" block
- Add "OR" block
- Add "AND" block
- Add "Conditional"

Beiwe Research Platform – Survey Branching Release Notes

One or more of the operators can be configured to implement your branching logic. Click on one or more of these new buttons in the **Edit Question** screen to get started. "**Add Conditional**" is a single operator as noted when used and as can be seen in the following screenshot.

The NOT, OR and AND blocks can be configured with multiple conditional statements and can be nested within each other. See an example in the following screenshot. Remember to click "**Add Question**" to save your questions and "**Save and Deploy**" to save the survey when you're done and as you're adding new questions. Add your survey delivery times as normal before your final **Save and Deploy**.

Beiwe Research Platform – Survey Branching Release Notes

Questions that use the conditional operators and thus are displayed conditionally can use any question type and therefore can be used to ask additional questions. In the example below, if Yes is answered to the "**Always Displayed**" question, a follow-on question is asked for quantity, while if No is answered, an informational box with advice is provided. Additional conditional questions could be set up using the answers from Questions 1 and/or 2 since either result in a unique answer.

When will I get an error when entering branching logic?

You won't get errors when you save contradictory logic, like the expression "*((Q1 == 0) AND (Q1 == 1))*". You will have to correct logic errors like this on your own when you notice a survey is not giving you the results you're expecting. Errors only appear when Beiwe can't parse the logic, for example, if you're referencing a question that doesn't exist anymore (e.g., if you referenced Q5 and then deleted it, even if there's a new Q5) or if you're referencing a question that doesn't have a numeric answer (like a checkbox question or alphanumeric response). If the logical expression can be parsed, even if the logical expression always evaluates to *False*, it won't show any errors when you click "**Save and Deploy.**"

An example of an error that can occur if there is a problem with branching logic when you click "**Save and Deploy**":

If Question 3 refers to Question 2:

Errors:
- Question 3: contains a reference to a question that is not of a numeric question type.

If Question 2 then gets deleted:

Errors:
- Question 2: contains a reference to a question that does not exist.

What happens to questions that aren't displayed?

If any comparative operator (<, <=, ==, >=, or >) references a question that wasn't answered, the expression should evaluate as False. For example, if you're evaluating (Q3 < 1) AND (4 > 3) and Q3 wasn't answered, it should evaluate as (False) AND (4 > 3). If you're evaluating NOT(Q3 == 1) and Q3 wasn't answered, that should evaluate as NOT(False), which equals True.

If Q3 never displayed, it cannot have been answered. If Q3 displayed, and the participant answered it, and then the participant went back in the survey and changed a previous answer so that Q3 shouldn't have displayed, the answer to Q3 is still there to be operated on. Thus, if you want Q4 to never display if Q3 didn't display either, you could add a copy of the Q3 display logic to Q4's display logic inside an AND block.

Entering long text responses: How to write responses so the app shows a new line

You need to use two newlines when entering your text in the "**Question Text**" box in order to get a paragraph break to display in the app. A single newline is ignored.

Beiwe randomize feature

Beiwe's randomize feature does not work with survey branching. If you click the "**Randomize**" box when configuring your survey, it will disable any branching questions in the survey.

Frequently Asked Questions

What version of iOS is required for Beiwe?
Beiwe will run on iOS 12 or higher.

What version of Android is required for Beiwe?
Beiwe will run on an Android operating system release 8.0 or higher.

Will questions be displayed to patients in order (e.g., Question 1 then Question 2, etc.)
Yes, questions will be displayed in numerical order unless the checkbox to randomize questions is selected.

What if I don't want to use survey branching?
You do not have to use survey branching. Simply ignore the conditional branching operators and enter your survey questions as you always have.

Can I use survey branching in an existing study or survey?
Yes.

What does == mean?
It means equals. == is the JavaScript programming language convention for equals and our developers see the world in JavaScript programming.

Which questions can use survey branching?
Survey branching can only operate on slider questions, radio button questions, and numeric open response questions. Survey branching does not work with checkbox questions or the two other types of open response questions, single-line and multi-line text. A radio button question can only have one answer selected at a time, so its numeric answer is clear: it's just the index of the selected answer. Because a checkbox question can have multiple answers selected at once, the numeric value of a checkbox question's answer is not clear; thus, branching cannot be used.

Where can I get a tutorial on Boolean logic?

Here's a good "cheat sheet", compliments of http://www.i-programmer.info/babbages-bag/235-logic-logic-everything-is-logic.html:

Truth Tables

The rules for combining expressions are usually written down as tables listing all of the possible outcomes. These are called truth tables and for the three fundamental operators these are:

P	Q	P AND Q
F	F	F
F	T	F
T	F	F
T	T	T

P	Q	P OR Q
F	F	F
F	T	T
T	F	T
T	T	T

P	NOT P
F	T
T	F

Questions, Comments or Feedback?

If you have any questions about the information in this document, please reach out to the Onnela Lab Beiwe Administrator. Hassan Dawood, Senior Research Operations Manager: hdawood@hsph.harvard.edu