Branching Logic

What is branching logic?

Branching Logic may be employed when fields in the database need to be hidden during certain circumstances. For instance, it may be best to hide fields related to pregnancy if the subject in the database is male. If you wish to make a field visible ONLY when the values of other fields meet certain conditions (and keep it invisible otherwise), you may provide these conditions in the Branching Logic section in the Online Designer (shown by the double green arrow icon), or the Branching Logic column in the Data Dictionary.

For basic branching, you can simply drag and drop field names as needed in the Branching Logic dialog box in the Online Designer. If your branching logic is more complex, or if you are working in the Data Dictionary, you will create equations using the syntax described below.

In the equation you must use the project variable names surrounded by **[ ]** brackets. You may use mathematical operators (=,<,>,<=,>=,<>), Boolean logic (and/or), and unary Boolean not (!). You may nest within many parenthetical levels for more complex logic.

You must **ALWAYS** put single or double quotes around the values in the equation UNLESS you are using > or < with numerical values.

The field for which you are constructing the Branching Logic will ONLY be displayed when its equation has been evaluated as TRUE. Please note that for items that are coded numerically, such as dropdowns and radio buttons, you will need to provide the coded numerical value in the equation (rather than the displayed text label). See the examples below.

[sex] = "0"	display question if sex = female; Female is coded as 0, Female
[sex] = "0" and [given_birth] = "1"	display question if sex = female and given birth = yes; Yes is coded as 1, Yes
([height] >= 170 or [weight] < 65) and [sex] = "1"	display question if (height is greater than or equal to 170 OR weight is less than 65) AND sex = male; Male is coded as 1, Male
[last_name] <> ""	display question if last name is not null (aka if last name field has data)

Can fields from different FORMS be used in branching logic?

Yes, branching logic may utilize fields either on the current data entry form OR on other forms. The equation format is the same, so no special formatting is required.

Can fields from different EVENTS be used in branching logic (longitudinal only)?

Yes, for longitudinal projects (i.e. with multiple events defined), branching logic may utilize fields from other events (i.e. visits, time-points). The branching logic format is somewhat different from the normal format because the unique event name must be specified in the logic for the target event. The unique event name must be prepended (in square brackets) to the beginning of the variable name (in square brackets), i.e. [unique_event_name][variable_name]. Unique event names can be found listed on the project's Define My Event's page on the right-hand side of the events table, in which the unique name is automatically generated from the event name that you have defined.

For example, if the first event in the project is named "Enrollment", in which the unique event name for it is "enrollment_arm_1", then we can set up the branching logic utilizing the "weight" field from the Enrollment event: [enrollment_arm_1][weight]/[visit_weight] > 1. Thus, presuming that this field exists on a form that is utilized on multiple events, it will always perform the branching logic using the value of weight from the Enrollment event while using the value of visit_weight for the current event the user is on.

Is branching logic for checkboxes different?

Yes, special formatting is needed for the branching logic syntax in 'checkbox' field types. For checkboxes, simply add the coded numerical value inside () parentheses after the variable name:

[variablename(code)]

To check the value of the checkboxes:

'1' = checked

'0' = unchecked

See the examples below, in which the 'race' field has two options coded as '2' (Asian) and '4' (Caucasian):

[race(2)] = "1"	display question if Asian is checked
[race(4)] = "0"	display question if Caucasian is unchecked
[height] >= 170 and ([race(2)] = "1" or [race(4)] = "1")	display question if height is greater than or equal to 170cm and Asian or Caucasian is checked

Can you utilize calculated field functions in branching logic?

Yes, see the list of functions that can be used in logic for Report filtering, Survey Queue, Data Quality Module, and Automated Survey Invitations.

How do I hide a calculation with branching logic without causing a notification from the web form?

If a calculated field is hidden by branching logic but will evaluate to a number, it causes the web form to notify the data enterer about a hidden field with data in it. To avoid receiving the notification, use an "if...else" statement--if(X, ValueIfTrue, ValueIfFalse)-where the value if false is "".

The statement should be written so if the calculated field is hidden, it will evaluate to false and provide the empty response, meaning there will be no data in the field. This will prevent the notification.

Can you program branching logic using dates?

Yes, see the list of functions that can be used in logic for Report filtering, Survey Queue, Data Quality Module, and Automated Survey Invitations.

Is it possible to use branching logic to skip an entire form or forms?

Branching logic will only hide questions, not entire data collection instruments. If you have a list of data collection instruments (DCIs) in a project (traditional) or event (longitudinal), you will see every form even if you hide all the fields with branching logic on that form. You'll have to click through the forms or "save and go to next form". A work around may be to add a descriptive text (reverse branching logic to show when all fields are hidden) that the form is not applicable to that specific record or just leave the form blank.

If using the Survey Queue for participants entering data, you can hide/display entire surveys.

Is it possible to use branching logic to skip an entire section?

Branching logic must be applied to each field. It cannot be applied at the form or section level. Section headers will be hidden *only* if all fields in that section are hidden.

My branching logic is not working when I preview my form. Why not?

Simply previewing a form within the Online Designer will display all questions. In order to test the functionality of your branching logic (and calculated fields), you must enter new records and enter test data directly into your forms.

Why does REDCap slow down or freeze and display a message about a javascript problem when I try to use branching logic syntax or Drag-N-Drop Logic builder in a longitudinal project with over 1000 fields?

You are encountering a limitation that stems from having a lot of fields especially multiple choice fields in your project. If a good number of your fields involve multiple choices then the number of choices that the Drag-N-Drop Logic Builder has to load into the pop-up is large. So having a lot of fields with several choices each can slow down the system. The performance is further affected because REDCap uses javascript (powered by the user's browser) to do the drag-n-drop and also to process the conversion of the advanced syntax to the drag-n-drop method (if you decide to switch methods within the pop-up).

The slower your computer and the slower your browser (Internet Explorer is the worst, especially versions 6 and 7), than the slower the drag-n-drop method will be. Chrome is much faster at handling Javascript than other browsers and is recommended. The only other option is to use the data dictionary for building your branching logic.