Resource – Tips & Tricks

Below is a collection of helpful hints to give users different perspectives on solutions or ways of using REDCap that they may not have previously considered. They are interesting anecdotal items that can get those creative problem solving gears turning!

Hide a field from specific users

@IF([user-role-label]!="Administrator",@HIDDEN,'')

This little bit of code can be dropped into the Action Tag section of any field in order to make that field hidden to any user in your project that is not in the "Administrator" User Role (set in the User Rights page). You can use any Role you have in your project, and you can swap out the @HIDDEN tag with @READONLY or another Action Tag to change the behavior of the field.

Limit the # of Repeated Instances for an Instrument

[last-instance] <= "n"

Use this code in your Form Display Logic for the instrument you wish to limit. It must be an instrument that is enabled as a repeating instrument and the number you specify should be -1 your target max number. This will make the instrument inaccessible for any record that have n+1 repeated instances saved , and hence new instances cannot be added.

Live value confirmation (email address, phone number, any field!)

When you need to double-check that a user has entered key data points correctly, you may want to ask them to enter it twice to be sure it is accurate. To do so, follow these steps:

1) Create the two identical fields, say [email1] and [email2], labeling them accordingly (call the second one a "Confirmation ..."). Make them both required.

2) Create a basic calculation field, and add the calculation if([email1]=[email2],1,''). Make it required and add @HIDDEN-SURVEY to the Action Tag section.

3) Create 2 descriptive text fields and add the text "Values do not match!" and "Values Match." in red and green, respectively.

4) In the branching logic for the non-matching descriptive text field, use [email1] != [email2] and in the branching logic for the matching descriptive text field, use [email1] = [email2].

This prevents a participant from submitting a survey unless the two fields match exactly. The calculation field acts as a submission block because it will only carry a value if the two fields match, and being required, must carry a value to submit the instrument.

Dynamic Color-coding for Record IDs based on record data conditions

[cc_calc] Text Field: @CALCTEXT(if([somefield1]<>"" AND [somefield2]<>"" AND [somefield3]<>"","some_6digit_hexcode","some_other_6digit_hexcode")) @HIDDEN

Custom Record Label: - ([somefield])

This set of code portions will allow you to change the color of your Record ID text anywhere that it appears in your project (RSD, Record Homepage, top of a data entry instrument, etc.) This example creates a Text field using the @CALCTEXT action tag that references the conditions of three fields and results in a single if/then color indication (e.g., red or green). You could take this as far and as complex as you like however with nested if() statements. Note: @CALCTEXT is needed instead of a normal Calc field type because the hex color codes often include letters.

Once the text/calc field is built, you then add some html (<span tag>) including the piped value of your text/calc field in the label to make the label dynamically change. Then, whatever field you are actually piping between the span tag will show up and be affected by the color code, which was piped from your text/calc field.

You can even use multiple calc fields for setups that utilize HSL values to get extra fancy.

Using country code if a phone number seems to not be recognized for Twilio use

If when using Twilio you come across a phone number that seems to not be recognized by REDCap, it could be due to the area code being problematic. IF this is the case, the issue can be resolved by adding 1 in front of the phone number. This is operationally the same number, just including the US country code. In doing so, the number should be usable.

Some previously known problematic area codes have been: **743, 463, 854**