REDCap

Research Electronic Data Capture

Avoid Common Pitfalls

THE UNIVERSITY OF CHICAGO MEDICINE & BIOLOGICAL SCIENCES

CENTER FOR RESEARCH INFORMATICS

Version 6.5.0 - © 2015 Vanderbilt University

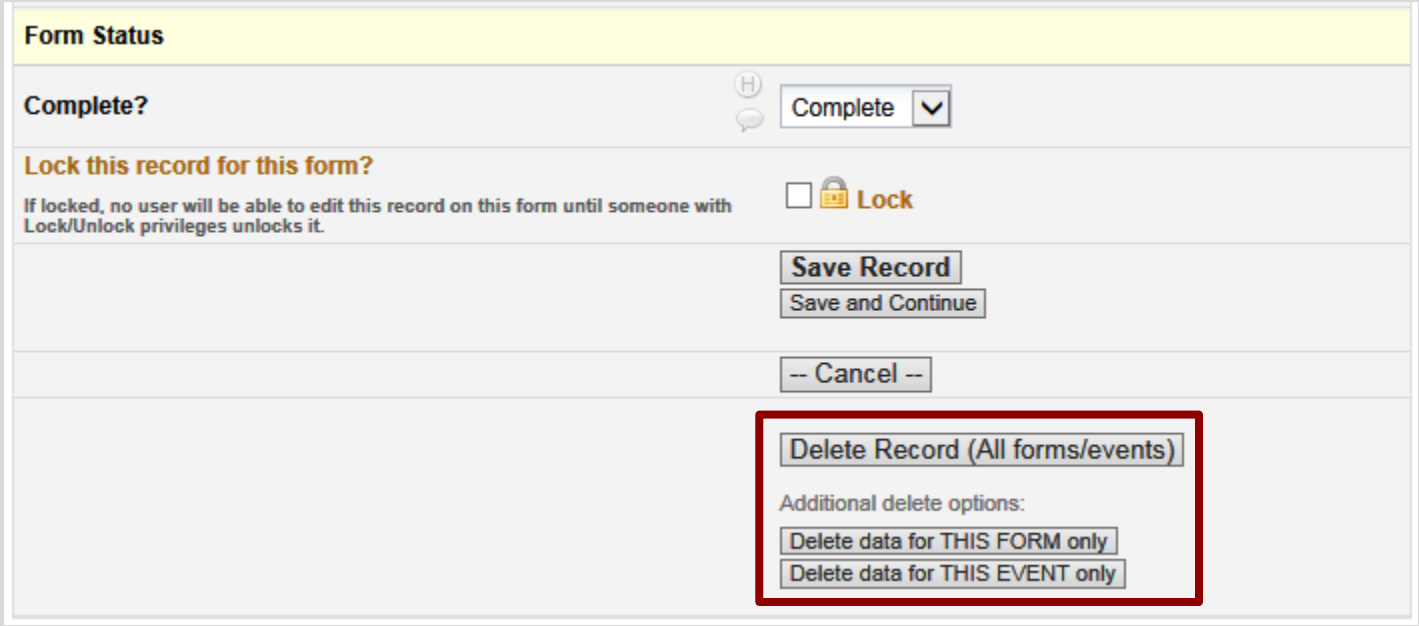
COMMON PITFALLS

1. Deleting Records
2. Deleting Events
3. Renaming Fields
4. Enumeration Changes
5. Not Testing your Project
6. Not Moving to Production Status
7. Branching Logic Error
8. Calculated Fields

1. DELETING RECORDS

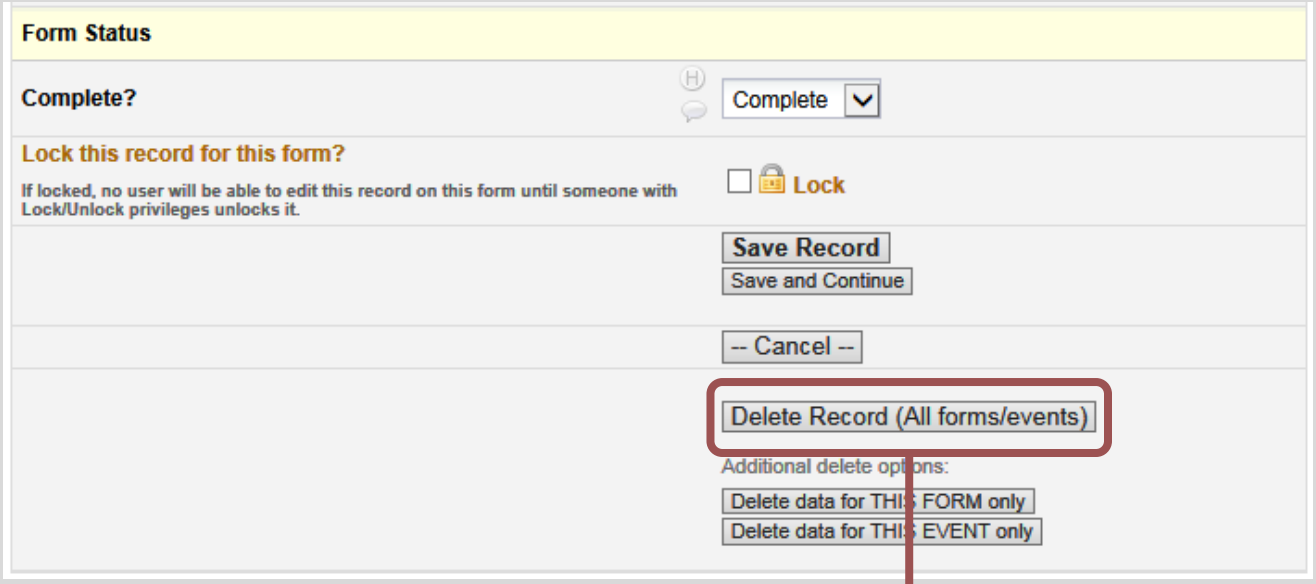
There are 3 options for deleting records:

- Delete Record – all forms and events
- Delete Record – specific form
- Delete Record – specific event



1. DELETING RECORDS – OPTION 1

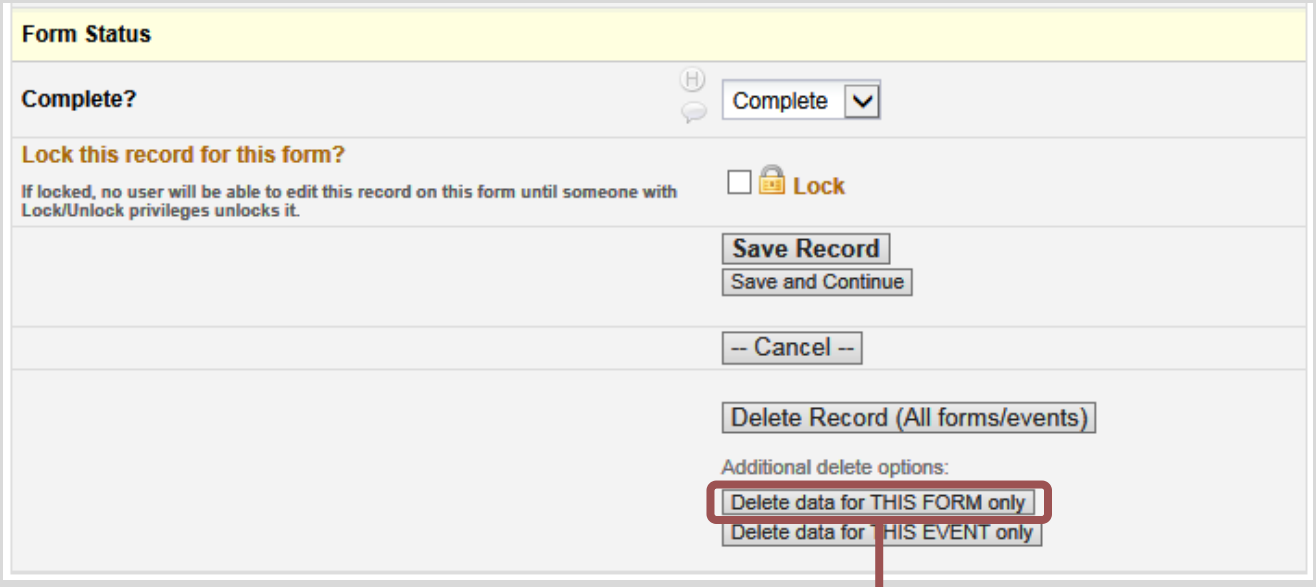
OPTION 1 will delete the record ENTIRELY from ALL forms (and events in longitudinal projects):

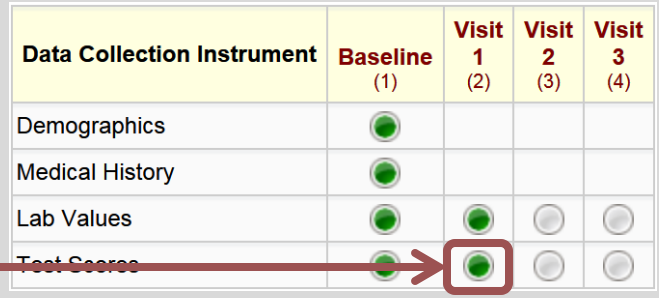


Data Collection Instrument	Baseline (1)	Visit 1 (2)	Visit 2 (3)	Visit 3 (4)
Demographics	●			
Medical History	●			
Lab Values	●	●	○	○
Test Scores	●	●	○	○

1. DELETING RECORDS – OPTION 2

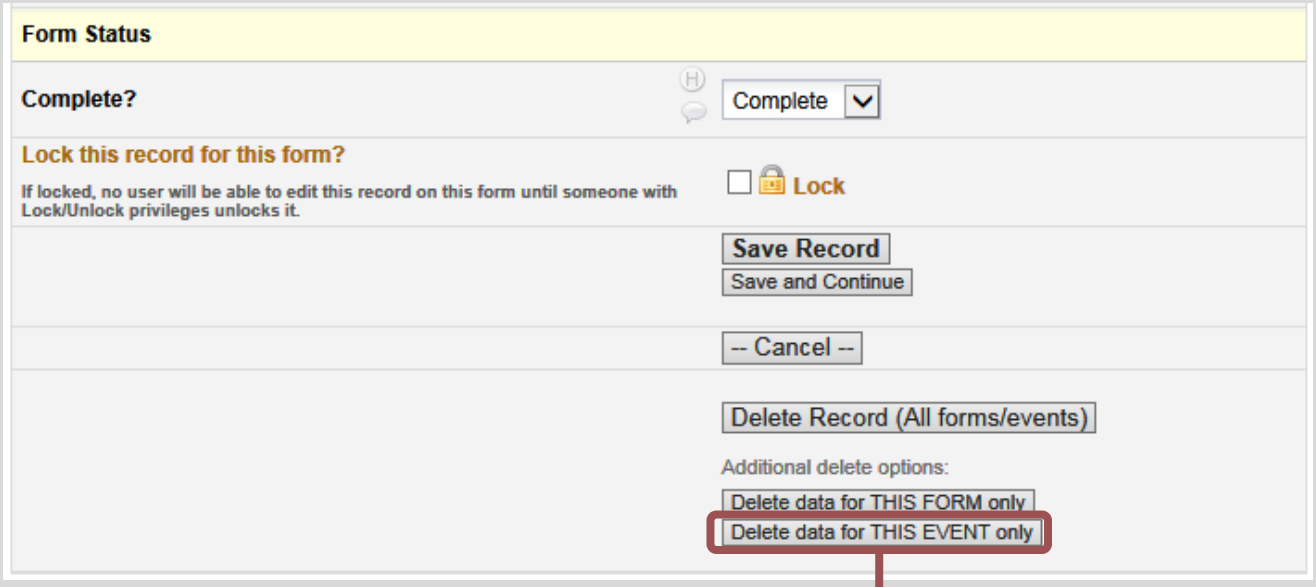
OPTION 2 will delete the record from only the specific form chosen:





1. DELETING RECORDS – OPTION 3

OPTION 3 will delete the record from only the specific event chosen:



Data Collection Instrument	Baseline (1)	Visit 1 (2)	Visit 2 (3)	Visit 3 (4)
Demographics	●			
Medical History	●			
Lab Values	●	●	○	○
Test Scores	●	●	○	○

2. DELETING EVENTS (LONGITUDINAL)

- Events in REDCap are auto-named unlike variable fields which need to be named.

	Event #	Event Name	Unique event name (auto-generated)
	1	Baseline	baseline_arm_1
	2	Visit 1	visit_1_arm_1
	3	Visit 2	visit_2_arm_1
Add new event		Descriptive name for this event	

- If an event is deleted, the link to the data is broken so data will not be 'viewable'.

- Deleted events cannot be restored without the Administrator's help.

3. RENAMING FIELDS

Once the project is in production with live data collected, **there is no renaming** of fields because the field names are linked directly to the REDCap data. If a new field name is absolutely necessary, you will need to export the data, create a new field with the new name, upload the data to the new field, and delete the old field with the old name. The field name will also need to be manually changed in the following places:

- Branching logic
- Calculations
- Auto-survey logic
- Survey queue logic
- Piping
- Reports
- Field comments (export Field Comment Log before you change the field names)

4. ENUMERATION CHANGES

Once the project is in production with live data collected, Dropdown list, Radio Buttons, and Check Boxes **should not be renumbered** because the numbers are linked directly to the REDCap data. The field choices can be in any order regardless of the number assigned, as the choices will be shown in the **order listed** rather than **numeric order**.

For example, let's start with:

1, Allentown
2, Beaumont
3, Carson

Say you need to add "Bakersville". Do not add as follows:

1, Allentown
2, Bakervsille
3, Beaumont
4, Carson

Instead, add "Bakersville" as follows (without changing numbers):

1, Allentown
4, Bakervsille
2, Beaumont
3, Carson

5. NOT TESTING YOUR PROJECT

- It is important to test the **essential components** of your project before moving it into production.
- Create a few **test records** and entering some data for each to ensure that your data collection instruments look and behave how you expect, especially branching logic and calculations.
- Review your test data by creating **reports** and **exporting** your data to view in Excel or a statistical analysis package.
- If you have **surveys**, complete the surveys as if you were a participant by using the Public Survey Link or Participant List by sending a survey invitation to yourself.
- The **best way** to test your project is to use it as if you were entering real production data, and it is always helpful to have team members or colleagues take a look at your project to get a fresh set of eyes looking at it.

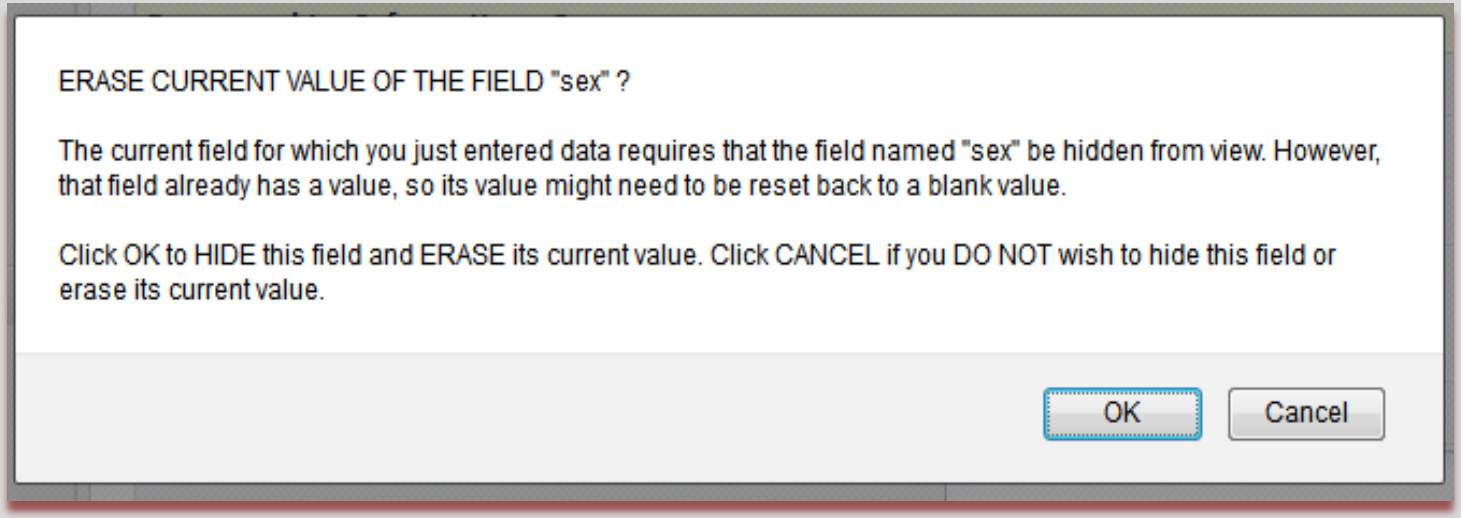
6. NOT MOVING YOUR PROJECT TO PRODUCTION

Once you have designed and tested your project, you should move your project to production for **very important reasons**:

- Your **data will be safer**. You can safely make changes to your project in "Draft Mode" while leaving the project undisturbed until the changes are actually applied.
- Renaming of fields and/or events, and renumbering of coded fields (drop-down, radio button, check boxes) will come with **warning messages**.
- Changes to forms that might compromise your data will be reviewed and approved by the **Administrator** before they are committed to the project.
- REDCap keeps a copy of **past Data Dictionaries** so you can refer back to them if there are ever any questions.

7. BRANCHING LOGIC ERROR

When a field that will be hidden by branching logic has value, REDCap offers to delete the value (to avoid having unnecessary values in the database):



** See next page for solutions.*

7. BRANCHING LOGIC ERROR *(continued)*

This branching logic error is caused by one of several issues:

1. **Design error in branching logic.** For example, branching logic uses "today" to calculate age, so the values that were valid (fields were visible) when entered became invalid (fields are now hidden). *Solution:* add a field "consent date" and calculate age against this field.

2. **Change in branching logic after data was entered.** The fields with data were visible under old branching logic, but now will become hidden. *Solution:* verify new branching logic is correct, and delete the data fields if necessary.

3. **Changed data in the field that controls whether these fields are visible.** *Solution*: look at the data entered so far (export all) and decide if data is correct.

**Also, if you do not want the field hidden once it contains data, add the following as an "OR" condition on the existing branching logic:*

[field_name]<>""

8. CALCULATED FIELDS

- **Calculations should be considered a tool not data**. Calculations should be reapplied during data analysis.

- **Do not reference calculated fields within calculated fields**. When multiple calculations are performed, the order of execution is determined by the alphabetical order of the associated field names.

- **Multiple calculations on a form**. Order of execution is determined by the alphabetical ordering of associated variable names.

  Problem: Here, Calculation 2 will occur before Calculation 1!

  - Calculation 1 [weight_met] =[weight]*.45359237
  - Calculation 2 [BMI]=[weight_met]/([height]^2)

  Solutions:

  - Best solution: calculate BMI in one step
  - Other solution: rename variable to change order

8. CALCULATED FIELDS *(continued)*

- **Avoid using "today" in calculated fields**. This is because every time you access and save the form, the calculation will run. So if you calculate the age as of today, then a year later you access the form to review or make updates, the age as of "today" will also be updated (+1 yr). Instead, calculate ages off of another field (e.g. screening date, enrollment date).

- **For longitudinal projects, the unique event name must be prepended** (in square brackets) to the beginning of the variable name (in square brackets) specified in the equation. For example:

  [enrollment_arm_1][weight]/[visit_weight]

  Presuming this calculated field exists on a form that is utilized on multiple events, it will always perform the calculation using the value of weight from the Enrollment event while using the value of visit_weight for the current event the user is on.

SUMMARY

Deleting Records – *choose correct option*
Deleting Events – *only if absolutely necessary*
Renaming Fields – *only if absolutely necessary*
Enumeration Changes – *don't renumber*
Not Testing your Project – *test, test, test!*
Not Moving to Production – *safer status*
Branching Logic Error – *review logic and data*
Calculated Fields – *use sparingly as a tool*

FOR FURTHER HELP

website

- https://cri.uchicago.edu/redcap

email

- redcap@rt.cri.chicago.edu