Using Action Tags in REDCap

Action Tags are an excellent way to customize the data entry experience for surveys and forms. They are special terms that begin with the '@' sign that can be placed inside a field's Field Annotation box when adding or editing a field. Each action tag has a corresponding action that is performed for the field when displayed on data entry forms and survey pages.

To start using Action Tags, navigate to the Online Designer in a project. When adding or editing a field, add the action tag into the Action Tags / Field Annotation box in the Edit Field popup (see below) by either clicking on “Add” next to the Action tag(s) you want to use or by typing them in manually. Some action tags need to be further configured while others do not need any additional configuration.

You can use as many action tags as you want for a single field, but if you do use more than one tag for a field, make sure to put a space or line break between them. Because the action tags are used as part of the Field Annotation, they are not displayed anywhere on the page.


A list of all available action tags, and their descriptions, can be found here: https://redcap.ctsi.ufl.edu/redcap/redcap_v8.8.2/Design/action_tag_explain.php

Below is a list of all available action tags that you may use on a data entry form or survey, their descriptions, use cases and configuration examples.
@CHARLIMIT: Sets the maximum number of characters that can be entered into a Text or Notes field, and displays the number of characters remaining. The format must follow the pattern @CHARLIMIT=XX, in which XX is the desired max character count (alternatively, the number value can be inside single or double quotes, i.e., @CHARLIMIT='XX').

NOTE: This action tag cannot be used at the same time as @WORDLIMIT for the same field.

Use cases:
1) Limiting the amount of characters that can be typed into a non-validated Text field or Notes field.
2) Especially useful for data fields collecting ‘Comments’ where project administrators don’t want a novel’s worth of text entered.

Example and demo of setting up a maximum of 50 characters:

Configuration:

Data entry view:

@DEFAULT: Sets a field's initial value. This allows a field to have a specified default value when viewing the field on a survey or data entry form that has not yet had any data saved for it (i.e., when the form status icon is gray or when a survey page has not had any data entered yet). The format must follow the pattern @DEFAULT="????", in which the desired default value should be inside single or double quotes. For checkbox fields, simply separate multiple checkbox values with commas - e.g., @DEFAULT='1,3,6'. For text fields, you may even perform Piping inside the default value to pipe data from another field in the project - e.g., @DEFAULT='Name: [first_name] [last_name], DOB: [dob]'.

NOTE: If being used on a date or date/time field, the date value inside the quotes must be in Y-M-D format - e.g., @DEFAULT='2007-12-25' - regardless of the field's validated date format.

Use cases:
1) Having numerous yes/no questions on a form, where almost all users will have the answer of ‘No’ for every question (saves data entry time).
2) Setting a field’s value when all records will have the same value, such as project close out date or date of continuing review.
3) Piping data into a radio button/drop down field type where there will only be one answer to choose from and the piped data needs to be downloadable (see example with subject’s enrollment date below).
In the Checkbox field type (multiple answers) example below, options ‘1’ and ‘3’ (‘Cancer’ and ‘Migraines’) will be checked by default when the form is accessed:

In the Radio Button field type (single answer) example below, option ‘1’ (Asian) will be checked by default when the form is accessed:

In the example below, the subject’s ‘date of enrollment’ [demo_date] needs to be piped into the “What date was the subject enrolled into the study?” data field, and the piped labeled data needs to be downloaded onto a report.

Note: If a radio button field type was not used, the date would be piped into the field but this data will only show up when viewing a form/survey. The piped data would not download onto a report or for data analysis unless it is selected as seen above (circle is filled in).
@HIDDEN: Hides the data field on the survey page and the data entry form. Field will stay hidden even if branching logic attempts to make it visible. This action tag is useful when you want to retire a data field from the project but don’t want to lose any data that’s already been collected.

Use cases:
1) Retiring a question/data field on both the form and survey so that the question no longer appears but no existing data for this question will be lost.
2) Hiding calculated fields from those entering data and submitting surveys so that they cannot view that data but the data is still stored in the database and can be viewed by downloading the data.

In this example, I want to no longer display the Ethnicity question on the data entry form/survey:

Before @HIDDEN was used:

After @HIDDEN was used:

Note that the Ethnicity data field has been hidden from the form and survey.

@HIDDEN-FORM: Hides the field only on the data entry form (i.e., not on the survey page). Field will stay hidden even if branching logic attempts to make it visible.

Use cases:
1) Displaying certain questions only to survey participants but not to those with data entry rights.
2) Retiring a question/data field on a data entry form so that the question no longer appears but no existing data for this question will be lost.
In the example below, I want to hide the Comorbidities question from anyone not entering data into the form as a survey:

When the form is opened as a survey, you see the Comorbidities question:

When the form is used for normal data entry (via Add/Edit Records), you do not see the Comorbidities question:
@HIDDEN-SURVEY: Hides the field only on the survey page (i.e., not on the data entry form). Field will stay hidden even if branching logic attempts to make it visible.

Use cases:
1) Hiding questions or data fields from survey participants, such as fields calculating their BMI, age, instrument scoring, etc., but not from the data entry staff.
2) Retiring a question/data field on a survey form so that the question no longer appears but no existing data for this question will be lost.

In the example below, the user wants to hide the calculated total score from survey participants, but allow data entry staff to see the score when viewing the form:

When the form is completed as a survey, a user does not see the total score at the bottom:

When the form is accessed via Add/Edit Records, a user does see the total score at the bottom:
@HIDEBUTTON: Hides the 'Now' or 'Today' button that is typically displayed to the right of date, time, and date/time fields.

Use case:
1) Hiding the ‘Today’ button on date fields, where the date value could not possibly be today, such as date of birth or any dates in retrospective data collection.

In the example below, I do not want the ‘Today’ button to appear next to the Date of birth field, which is the default when you validate a field as a date or a date/time:

How this field would normally look:

```
Date of birth
```

This is how it looks with the @HIDEBUTTON action tag:

```
Date of birth
```

Note that the ‘Today’ button disappears but the calendar widget is still usable.
@HIDECHOICE: Hides one or more choices of a Checkbox, Radio, Drop-down, Yes-No, or True-False field. The format must follow the pattern @HIDECHOICE='??', in which the coded values should be inside single or double quotes for the choice(s) you wish to hide. If more than one choice needs to be hidden, then provide all the coded values separated by commas.

For example, to hide the choice ‘Monday (1)’, you would have @HIDECHOICE='1', but if you wanted to additionally hide ‘Tuesday (2)’, you would have @HIDECHOICE='1,2'. NOTE: If the hidden choice has already been selected for a given record, then the choice will not be hidden on the survey or form for that record/event. NOTE: This action tag works only in limited fashion with a matrix of fields, in which it will simply hide the checkbox/radio for a field in the matrix but still display the column for that choice in the matrix.

Use case:

1) Retiring answer choice(s) after utilizing it for a while in data collection, thus allowing you to hide the choice from that point afterward without orphaning any of the choice’s data, which would happen if you simply deleted the choice.

In the example below, I no longer want users to be able to select option ‘2’ (Jane Doe) or ‘3’ (Albert Alligator), but I do not want to delete these choices since it would delete existing data.

How the form/survey looks pre-action tag:

How the form/survey looks post-action tag:
@LATITUDE: Allows a Text field to capture the latitude of the user, in which the user will be prompted on the webpage to allow or deny this. Once the value is captured, it will not be changed when visiting the page later. The field will be disabled and will not allow users to edit the value, but it will allow them to remove the value altogether.

@LONGITUDE: Allows a Text field to capture the longitude of the user, in which the user will be prompted on the webpage to allow or deny this. Once the value is captured, it will not be changed when visiting the page later. The field will be disabled and will not allow users to edit the value, but it will allow them to remove the value altogether.

Use case:
1) Collecting the location where data entry occurred.

NOTE: While the GPS value will be very accurate when captured on a mobile device, it will not be nearly as accurate when using a browser on laptops and desktop computers.

NOTE: You need to create two separate fields to collect both latitude and longitude. You cannot combine these action tags into one field.
@MAXCHECKED: Allows a checkbox field to have a maximum number of checkboxes that can be checked. If other checkbox options are clicked after the maximum has been reached, those choices will not be able to be checked. The format must be @MAXCHECKED=#, in which # is a positive integer. For example, @MAXCHECKED=3 will allow a maximum of 3 checkbox choices to be checked.

NOTE: This action tag can only be utilized by Checkbox field-types, and it does not get enforced during data imports.

Use case:
1) Having a list of 10 checkbox options but a user should only be able to select 5 of them instead of all of them, ensuring that more boxes are not checked than should be, i.e., a checkbox field type collecting “What are your top 5 health concerns” with more than 5 health concerns listed.

In this example, I do not want more than 2 options selected:

On the data entry form/survey, if I try to click on Migraines, it will not let me select that box because I already have 2 options selected.

To be able to select Migraines, I must un-select one of the other options:

Another example:
@MAXCHOICE: Causes one or more specified choices to be disabled (i.e., displayed but not usable) for a checkbox, radio button, or drop-down field after a specified amount of records have been saved with that choice. For example, @MAXCHOICE(0=50,1=75,2=50) would imply that once 50 records have selected the '0' coded choice, that choice will become disabled for any record viewed afterward that does not have that choice saved, such as when the form/survey is opened for a new record, and thus 75 records for choice '1', 50 for choice '2', etc. The format must be the action tag name followed by the choice designations inside parentheses, in which each will follow the format of ChoiceCode=MaxCount with each choice separated by commas.

**Note:** For longitudinal projects, when counting the number of choices saved for records, it will treat all events as separate, meaning that in the example above, choice '0' can have a max number of 50 selections in EACH event in the project.

**Use cases:**
1) Only allowing an option to be selected ‘x’ number of times so that it becomes disabled after it has been selected ‘x’ number of times.
2) Having an even distribution among answer choices.

In the example below, I only have 1 gift card, 10 stuffed animals, 50 coffee mugs, etc., so I do not want more than one person to select the Gift Card option:

Participant 1 has selected a Gift Card:

Therefore, participant 2 cannot select Gift Card, so the option is not selectable and grayed out:
@NONEOFTHEABOVE: Allows for the designation of a checkbox choice to be a 'none of the above' option, thus ensuring that no other choices are checked if that one choice is selected. This means that if that particular option is selected, it will alert the user that all other checked-off choices will be unchecked. And if the 'none of the above' option is already checked when the user checks off another choice, it will automatically uncheck the 'none of the above' option.

The format must follow the pattern @NONEOFTHEABOVE=??, in which ?? is the coded value of the 'none of the above' checkbox choice - e.g., @NONEOFTHEABOVE=98. Alternatively, it is possible to set multiple options as a 'none of the above' option at the same time, in which the coded values must be comma delimited inside single or double quotes - e.g., @NONEOFTHEABOVE='45,99'. This is useful for certain situations, such as if you have a 'none of the above' option and also have a 'refuse to answer' option, thus allowing either of those to be the only option that can be selected at a time.

NOTE: This action tag can only be utilized by Checkbox fields.

Use cases:
1) Making it possible to have a 'None of the above' and/or 'Refuse to answer' option for a checkbox field without the risk of users mistakenly having other choices selected at the same time.
2) Being able to use the ‘required field’ feature for checkbox field types when adding a 'None of the above' and/or 'Refuse to answer' option.

In the example below, I have a list of comorbidities. I do not want someone to be able to select 'None of the above' and a comorbidity, since I would now know which answer was correct.

If a user selects any of the checkboxes relating to a comorbidity and then tries to select ‘None of the above,’ it will force the user to either unselect ‘None of the Above’ or all the other selected options:
@NOW: Loads the current user's **date+time** into a blank Text field - similar to the @TODAY tag but additionally includes the time portion. If the field has validation, the value will adjust to match the date format. NOTE: The time used will be the user's local time, which is derived from their device/computer and might be different from the server if in another time zone. Also, do not use this tag on fields with branching logic because it will always prompt the user to erase the value, so look at using @HIDDEN instead if you wish to hide the field.

@TODAY: Loads the current user’s **date** into a blank Text field - similar to the @NOW tag but without the time portion. If the field has validation, the value will adjust to match the date format. Also, do not use this tag on fields with branching logic because it will always prompt the user to erase the value, so look at using @HIDDEN instead if you wish to hide the field.

**Use cases:**

1) Having the today’s date or date/time field automatically populate today’s date or date/time to reduce data entry.

2) When used with one of the @HIDDEN action tags, capturing a survey submission date on the last page of a survey which cannot be changed by the survey participant. Otherwise, a project user would need to run a report that includes the survey submission date/time stamp to get this information.
@PASSWORDMASK: Masks the value of a Text field so that the true value is **not visible** on the webpage after it has been entered (like password fields on login pages). This data is still available upon data export or via a report.

**Use case:**

1) Proving extra security when collecting sensitive information, such as social security number (SSN), UFID, database password, etc., since it cannot be viewed on the form itself.

In the example below, I want to collect a subject’s SSN, but have it masked upon entry:

![Data entry view](image-url)

In the example below, I want to collect a subject’s SSN, but have it masked upon entry:
@PLACEHOLDER: Is used to specify a short hint that describes the expected value of a Text field or Notes field (e.g. a sample value or a short description of the expected format). The placeholder is displayed inside the field before a value is entered. The format must follow the pattern @PLACEHOLDER='????', in which the text to be displayed should be inside single or double quotes.

Use cases:

1) Providing clarification on what type of data you are trying to collect in a text field, where you cannot provide a list of acceptable answers.

View when first accessing the form/survey:

The user then types data over the placeholder:
@RANDOMORDER: Randomizes the order of multiple choice field options as displayed on survey pages and data entry forms, in which their order will always be different each time the page is loaded by any given user.

NOTE: This action tag can only be utilized for the following field types: Checkbox, Radio, Drop-down, Yes-No, and True-False. This tag also works for enhanced radios and checkboxes on surveys. However, this tag does not work for a matrix of fields.

Use case:
1) Randomizing the order of available responses to force users to pay attention to how they are answering questions.

In the example below, I want the order of the answers to the Race question to be different for each user.

Ordering of answers on the Online Designer page:

Record 1’s view:

Record 2’s view:

Record 3’s view:

Note that the order of the answer choices differs for each user.
**@READONLY:** Makes the field read-only (i.e., disabled) on the survey page, the data entry form, and in the REDCap mobile app so that its value cannot be changed.

**@READONLY-FORM:** Makes the field read-only (i.e., disabled) only on the data entry form (i.e., not on the survey page or mobile app) so that its value cannot be changed.

**@READONLY-SURVEY:** Makes the field read-only (i.e., disabled) only on the survey page (i.e., not on the data entry form or mobile app) so that its value cannot be changed.

**Use cases:**
1. Piping data onto a form or survey that cannot be changed.
2. Making certain questions/data fields read-only for some users while other users can change the data.

In all the examples below, I am using the action tag @TODAY with one of the @READONLY action tags.

**@READONLY:**

<table>
<thead>
<tr>
<th>View from a survey (value cannot be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Today's date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>View from a form (value cannot be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today's date</td>
</tr>
<tr>
<td>01-30-2019 M-D-Y</td>
</tr>
</tbody>
</table>

**@READONLY-FORM:**

<table>
<thead>
<tr>
<th>View from a form (value cannot be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today's date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>View from a survey (value can be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Today's date</td>
</tr>
</tbody>
</table>

**@READONLY-SURVEY:**

<table>
<thead>
<tr>
<th>View from a form (value can be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today's date</td>
</tr>
<tr>
<td>01-30-2019 M-D-Y</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>View from a survey (value cannot be changed):</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Today's date</td>
</tr>
</tbody>
</table>
@USERNAME: Sets a field's value to the username of the current REDCap user. If this is used on a survey, the value will be '[survey respondent]'. Once the value is captured, it will not be changed when visiting the page later.

This action tag is best utilized with one of the @READONLY action tags so that the value cannot be changed.

Use case:
1) Capturing who entered data onto a form without having to look in the Logging module.

When I access the form, my user name is automatically entered into the field:

![Who's filling out this form?](image)

@WORDLIMIT: Sets the maximum number of words that can be entered into a Text or Notes field and also displays the number of words remaining. The format must follow the pattern @WORDLIMIT=XX, in which XX is the desired max word count (alternatively, the number value can be inside single or double quotes, i.e., @WORDLIMIT="50"). **NOTE:** This action tag cannot be used at the same time as @CHARLIMIT for the same field.

Use cases:
1) Limiting the amount of words that can be typed into a non-validated Text field or Notes field.
2) Especially useful for data fields collecting 'Comments' where project administrators don’t want a novel’s worth of text entered.

In the example below, there is a max 25-word count:

![Edit Field](image)

View from the form:

![Comments](image)

Notice the word count at the bottom of the comments field.