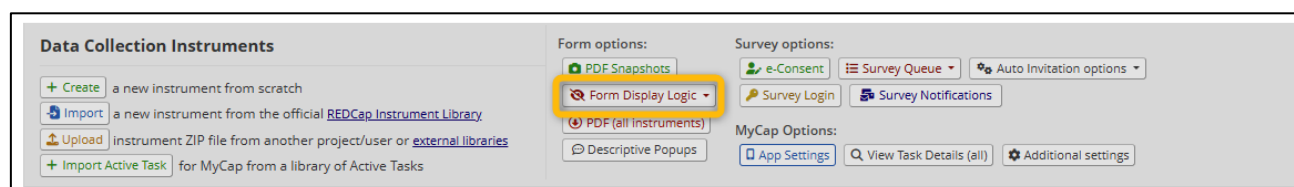




Form Display Logic: Uses in MyCap

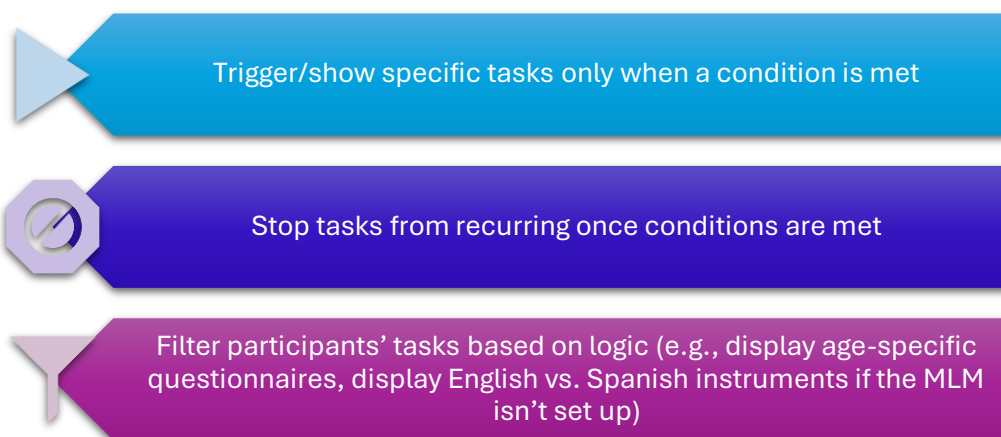


Contents

Introduction	1
Important Notes & Limitations	2
How to Use FDL in a Project with MyCap Enabled.....	3
Example Uses	4
1. Display Task Based on Response in Another Non-MyCap Enabled Tasks	4
2. Display Task Based on Response in Another MyCap Enabled Task	4
3. End Dependent Task Based on Response in Current Task.....	5
4. Display Tasks Based on Data Access Group (DAG)	5
5. Order Tasks or Display Task After Completion of Another	6

Introduction

With the release of REDCap v15.5.1, MyCap will support Form Display Logic (FDL). This feature will enhance study teams' capabilities by allowing tasks to filter or trigger within MyCap based on participant input. Previously, this was only possible if study teams created separate arms or projects to customize sets of tasks based on participant characteristics. Now, using FDL, study teams will be able to:



Important Notes & Limitations

Before using FDL in MyCap, it's important to understand the following requirements and limitations.

1. **Participants Must be Online to Trigger FDL:** Tasks that depend on FDL (e.g., to appear or stop occurring) will only respond if the device is online. If tasks are completed while the device is offline, the task schedule will update once the device is back online. This means that, if participants are offline, they could continue to complete tasks that should be stopped or may not see tasks that should appear when FDL is met.
2. **Specify the Instance:** Any instrument enabled for MyCap automatically becomes a repeating instrument. Therefore, if the variable(s) used to activate FDL is within a repeating instrument (or any MyCap task), the **instrument instance must be specified using the `[last-instance]` smart variable** appended to the variable name (e.g. `[variable_name][last-instance]`).
3. **Include the Event Arm on Longitudinal Projects:** In a longitudinal project, the arm and event variable must be prepended to the variable name within FDL (e.g., `[event_2_arm_1][variable_name]`).
4. **Tasks triggering FDL cannot allow Retroactive Completion in MyCap Settings:** MyCap's Retroactive Completion setting cannot be used for tasks that activate FDL. To keep tasks available until they are completed (and therefore not relying on retroactive completion), consider making tasks that activate FDL One-time Tasks or Repeating>Daily Tasks with **End By** criteria (e.g., After it has been completed **1** times).

Relative to which date task should be scheduled?

☐ Install Date

☒ Baseline Date

Schedule it how many times?

☐ One Time

☐ Infinite

☒ Repeating

Repeats:

☐ Fixed

On Day(s):

Number of days to delay:

End this Repeating task

☐ Never

☒ When any of the below conditions are TRUE:

☒ After it has been completed times

☒ After days have elapsed

☐ At exact date: M/D/Y

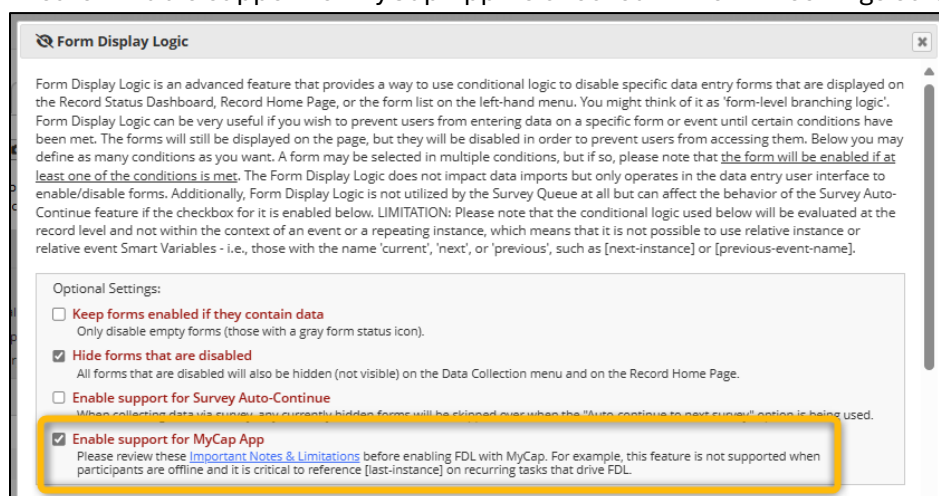
5. **FDL in MyCap cannot use the Infinite Schedule:** Form Display Logic should not be used on tasks that are scheduled to repeat infinitely.

6. **Task Schedules are Based on Baseline or Install Date:** Task schedules always start relative to the baseline or install date, not any FDL settings.
7. **Changes are immediate:** Any changes made and saved in FDL are instantly applied in MyCap. There is no need to publish a new version.
8. **Simplicity in Logic:** We recommend keeping logic simple!

How to Use FDL in a Project with MyCap Enabled

After enabling MyCap on a project and designing your instruments, follow the steps below to use FDL within MyCap:

1. You can configure your MyCap task schedules either before or after setting up FDL. However, keep in mind that tasks that activate FDL should use the Infinite schedule or have retroactive completion enabled in MyCap Settings.
2. Ensure “Enable support for MyCap App” is checked in the FDL settings screen.



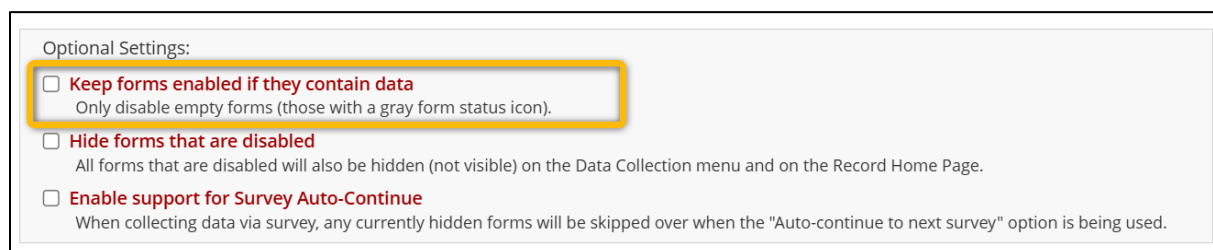
Form Display Logic

Form Display Logic is an advanced feature that provides a way to use conditional logic to disable specific data entry forms that are displayed on the Record Status Dashboard, Record Home Page, or the form list on the left-hand menu. You might think of it as 'form-level branching logic'. Form Display Logic can be very useful if you wish to prevent users from entering data on a specific form or event until certain conditions have been met. The forms will still be displayed on the page, but they will be disabled in order to prevent users from accessing them. Below you may define as many conditions as you want. A form may be selected in multiple conditions, but if so, please note that the form will be enabled if at least one of the conditions is met. The Form Display Logic does not impact data imports but only operates in the data entry user interface to enable/disable forms. Additionally, Form Display Logic is not utilized by the Survey Queue at all but can affect the behavior of the Survey Auto-Continue feature if the checkbox for it is enabled below. **LIMITATION:** Please note that the conditional logic used below will be evaluated at the record level and not within the context of an event or a repeating instance, which means that it is not possible to use relative instance or relative event Smart Variables - i.e., those with the name 'current', 'next', or 'previous', such as [next-instance] or [previous-event-name].

Optional Settings:

- ☐ **Keep forms enabled if they contain data**
Only disable empty forms (those with a gray form status icon).
- ☒ **Hide forms that are disabled**
All forms that are disabled will also be hidden (not visible) on the Data Collection menu and on the Record Home Page.
- ☐ **Enable support for Survey Auto-Continue**
When collecting data via survey, any currently hidden forms will be skipped over when the "Auto-continue to next survey" option is being used.
- ☒ **Enable support for MyCap App**
Please review these [Important Notes & Limitations](#) before enabling FDL with MyCap. For example, this feature is not supported when participants are offline and it is critical to reference [last-instance] on recurring tasks that drive FDL.

3. Refer to the Example Uses in the next section as needed.
4. You can test your logic by filling out forms in REDCap to ensure they are enabled or disabled as expected on the record status dashboard. To better visualize whether a form is enabled or disabled based on your logic, consider leaving 'Keep forms enabled if they contain data' unchecked.



Optional Settings:

- ☐ **Keep forms enabled if they contain data**
Only disable empty forms (those with a gray form status icon).
- ☐ **Hide forms that are disabled**
All forms that are disabled will also be hidden (not visible) on the Data Collection menu and on the Record Home Page.
- ☐ **Enable support for Survey Auto-Continue**
When collecting data via survey, any currently hidden forms will be skipped over when the "Auto-continue to next survey" option is being used.

Displaying: Instrument status only | [Lock status only](#) | [All status types](#)

Record ID	Intake	Survey A	Survey B	Animal survey: End if 'dog' is selected	Survey C	Survey D	for Group 1 (DAG)	Arranging Pictures - English	Arranging Pictures - Spanish	English - Spanish	Survey E	Survey F
1												
2												
3												

Example Uses

1. Display Task Based on Response in Another Non-MyCap Enabled Tasks

e.g. `[field_name] > 18 ; [field_name] = "1"`

EXAMPLE

This task will display if the age entered in the Intake form is less than 18.

▼ Condition 2:

Keep the following forms enabled...

- Intake
- Faces & Names 1
- Faces & Names 2
- Short Walk
- Display short walk is incomplete
- Display if age is < 18
- Animal survey: End if 'dog' is selected
- Display when animal survey is complete

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is TRUE.

`[age] < 18`

e.g., `[age] > 30 and [sex] = "1"` [How to use this](#)

2. Display Task Based on Response in Another MyCap Enabled Task

EXAMPLE 1: This task will become active if the participant responds “Yes” in Survey A. In this example, both Survey A and Survey B are scheduled to repeat daily. The datediff function including the schedule date of Survey A can be used to prevent Survey B from displaying on subsequent days until Survey A is completed.

▼ Condition 1:

Keep the following forms enabled...

- Intake
- Survey A
- Survey B
- Animal survey: End if 'dog' is selected
- Survey C
- Survey D
- Display for Group 2 (DAG)
- Arranging Pictures - English

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is TRUE.

`[do_you_like_the_color_blue][last-instance]='1' AND datediff('today',[scheduleddate][last-instance],'d') = 0`

e.g., `[age] > 30 and [sex] = "1"` [How to use this](#)

✓ Valid (The determination of validity may not be 100% accurate in all contexts.)

3. End Dependent Task Based on Response in Current Task **EXAMPLE 1:** This task will remain active as long as the response selected is **not** "dog". If dog is selected, the task will no longer be active in MyCap.

Condition 3:

Keep the following forms enabled...

- Intake
- Faces & Names 1
- Faces & Names 2
- Display if age is < 18
- Animal survey: End if 'dog' is selected**
- Survey displays if french selected in Intake form
- Display for Group 1 (DAG)
- Display for Group 2 AND age >15

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is **TRUE**.

[choose_an_animal][last-instance] != "3"

e.g., [age] > 30 and [sex] = "1" [How to use this](#)

EXAMPLE 2: This task will remain active if the age entered is under 30.

Condition 7:

Keep the following forms enabled...

- Display if age is < 18
- Animal survey: End if 'dog' is selected
- Survey displays if french selected in Intake form
- Display for Group 1 (DAG)
- Display for Group 2 AND age >15
- End if age is > 30**
- Survey
- Displays if Car is selected in "Survey"

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is **TRUE**.

[age_3][last-instance] < 30

e.g., [age] > 30 and [sex] = "1" [How to use this](#)

4. Display Tasks Based on Data Access Group (DAG)

Note: The DAG ID ([record-dag-id]), Name ([record-dag-name]), or Label ([record-dag-label]) can be used for this logic.

EXAMPLE

This task will be active for participants assigned to Group 1.

Condition 6:

Keep the following forms enabled...

- Display if age is < 18
- Animal survey: End if 'dog' is selected
- Display when animal survey is complete
- Survey displays if part choose french in app
- Survey displays if french selected in form 1
- Display for Group 1 (DAG)**
- Display for Group 2 AND age >15
- End if age is >30

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is **TRUE**.

[record-dag-id]='103'

e.g., [age] > 30 and [sex] = "1" [How to use this](#)

5. Order Tasks or Display Task After Completion of Another

EXAMPLE

Faces & Names 2 will only be active in MyCap after *Faces & Names 1* has been completed.

▼ Condition 1:

Keep the following forms enabled...

Intake

Faces & Names 1

Faces & Names 2

Short Walk

Display short walk is incomplete

Display if age is < 18

Animal survey: End if 'dog' is selected

Display when animal survey is complete

(You may select multiple forms by clicking them) [View list of selected forms](#)

...when the logic below is **TRUE**.

[faces_names_1_complete][last-instance] = '2'

e.g., [age] > 30 and [sex] = "1" [How to use this](#)