

# REDCap's Form Render Skip Logic (FRSL) External Module: Introduction and Guide to Setting It Up in your Project

Note: there is a **\$100 one-time fee** to activate this module in each REDCap project. This fee does not include setting up the FRSL control fields or configuring the module for your project. There is a \$100/hour fee if you would like the REDCap Team to configure the module for your project's needs.

## Introduction:

The FRSL external module functions as branching/skip logic for *entire* REDCap forms and surveys. FRSL provides a means for hiding or displaying entire forms/surveys based upon one or more control fields (i.e., data fields) or conditions (i.e., a record's data access group). FRSL hides unneeded forms for a specific record on the list of *Data Collection Instruments* due at each event, and grays out the colored buttons for the unneeded forms on the *Record Status Dashboard* and the *Record Home Page*.

## **Control Fields:**

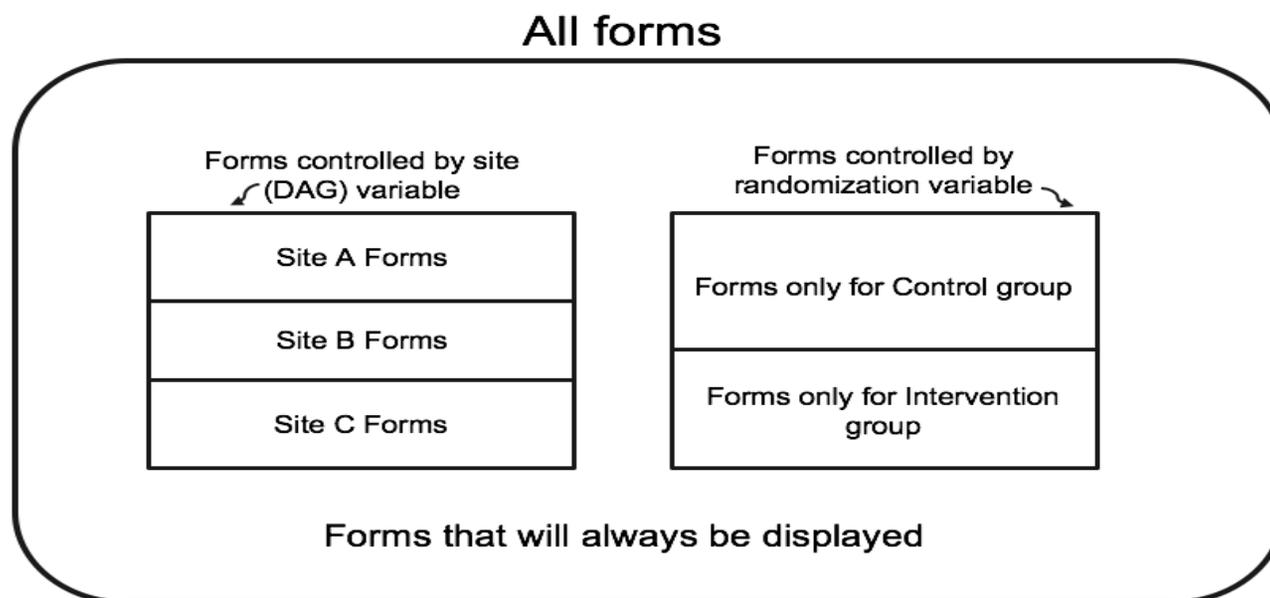
A control field is a single REDCap data or metadata field whose value will be tested to determine which forms will be displayed. If the condition evaluates as true, the forms listed under the condition will be displayed. If the condition is false, and no other true condition displays them, the forms will be hidden. Any form that is not configured within the FRSL module will display for all records.

Conditional fields can utilize event names for longitudinal projects, data piping and Smart Variables. The FRSL conditions and control fields can also be restricted to specific events in longitudinal projects.

### Some use cases of FRSL include:

- Avoid putting records into separate arms if subjects have different CRFs and/or time and event tables (i.e., controls vs. interventions).
- Make subsequent forms 'unavailable' if subject is not eligible or withdraws from the study, to avoid further data collection.
- Make site-specific forms available only when a record is assigned to a DAG (i.e., site-specific Informed Consents for multi-center studies).
- Eliminate the need to use the *Survey Queue* when not all subjects will receive the same set of surveys.
- Allows project designers to use the 'Auto-continue to the next survey' survey-termination feature even if the 'next' survey should not be administered to the subject.

### FRSL Form Diagram Example:

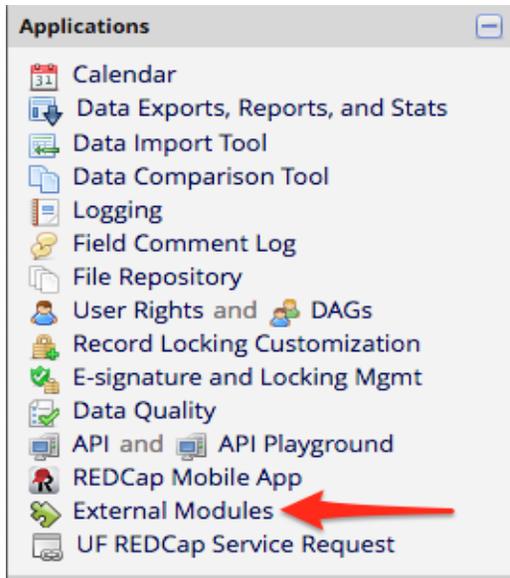


## Requirements to Use/Configure the Module:

- A REDCap project with *multiple* data collection forms/surveys.
- The User Right of ‘Project Design and Setup.’”
- A copy of the project’s Codebook or Data Dictionary to view the field names and raw data choices (i.e., Male = 1 and Female = 2) if using the Default (control field) method.
- Provide the REDCap Team with your fiscal contact information for the \$100 activation fee (to be billed upon module activation).

## Steps to enable and use the FRSL Module:

- 1) Build all your project’s data collection forms and data fields.
- 2) Assign all forms to study events (if project is longitudinal).
- 3) Contact the REDCap Support Team to enable to module ([CTSI-REDCAP-SUPPORT-L@LISTS.UFL.EDU](mailto:CTSI-REDCAP-SUPPORT-L@LISTS.UFL.EDU)) and provide your fiscal contact information in the email for the \$100 activation fee.
- 4) Once activated by the REDCap Support Team, click on the ‘External Modules’ link under the *Applications* menu in the project.
- 5) Click on ‘Configure’ next to the *Form Render Skip Logic* module.
- 6) Configure the module.
- 7) Click on ‘Save.’
- 8) Test the project’s FRSL configuration using test data.
- 9) Request to move the project into Production status.



Note: you will need the User Right of 'Project Design & Setup' to have access to the External Modules page to configure the module.

---

**Form Render Skip Logic - v3.3.3** Discoverable

This module hides and shows instruments based on the values of REDCap form fields. Multiple control fields can be defined to control the display of non-overlapping sets of forms.

[View Documentation](#)

[Configure](#)

[Disable](#)

When you click on the Configure button, you will be taken to the screen shown below, where you will set up the FRS� rules. There are two control modes available: *Default (Control Field)* and *Advanced (Equation with piping and/or Smart Variables)*. These modes are explained in the next section.

### Configure Module: Form Render Skip Logic ✕

Project Settings	Value
Prevent hiding of filled forms:	<input type="checkbox"/>
<b>1. Control fields:</b> <span>+</span>	
1. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)
1. Field:	<input type="text"/> <input type="text"/>
1. Default/fallback value:	<input type="text"/>
<b>1.1. Branching logic:</b> <span>+</span>	
1.1. Condition:	<input type="text"/> = <input type="text"/>
1.1.1. Target forms:	<input type="text"/> <span>+</span>
1.1. Restrict this rule to specific events:	<input type="checkbox"/>

Cancel Save

## Configuring the Module: Default Method (Control Fields):

The Default (Control Field) method uses a data field in the project to determine which forms to display or not display. For example, if you want to hide or reveal forms based on the subject's sex, and the subject's sex is a radio button field in the project (coded 1=Male and 2=Female), you would select the field in your project that contains the sex data for 'Field' and then select '1' as the condition for Males and '2' as the condition for Females. Under the '1' Condition, you would choose the 'Target forms' that you only want to show if the subject is male, and under the '2' Condition, you would choose the 'Target forms' that you only want to show if the subject is female. Any form not listed under the 'Target Forms' will appear for both males and females.

### Two 'Control mode' methods:

- 1) **Default** - uses a data field in the project to determine forms.
  - ❑ If using the Default method, you would choose the project field that determines which forms will be displayed.
  - ❑ If the project is longitudinal, you can also select the event that contains the control field.

The 'Condition' is the value that's being evaluated.

Target Forms are the forms that will appear only if the condition is true. You can restrict the FRSL rules to specific events as well (i.e., different time/event tables for subject types).

The screenshot shows the 'Configure Module: Form Render Skip Logic' window. It is divided into 'Project Settings' and 'Value' sections. The 'Project Settings' section includes a checkbox for 'Prevent hiding of hidden forms' and a '1. Control mode:' dropdown set to 'Default (Control field)'. The 'Value' section includes a '1. Field:' dropdown set to 'sex - Gender' and a '1. Default/fallback value:' field. Below this is a '1.1. Branching logic:' section with a '1.1. Condition:' dropdown set to '=' and a text field containing '1'. Underneath are two '1.1.1. Target forms:' dropdowns: 'patient\_morale\_questionnaire' and 'completion\_project\_questionnaire'. At the bottom, there is a '1.1. Restrict this rule to specific events:' checkbox which is currently unchecked. The 'Cancel' and 'Save' buttons are at the bottom right.

Annotations with arrows point to the following elements:

- A black arrow points from the text 'uses a data field in the project to determine forms.' to the 'sex - Gender' dropdown.
- A purple arrow points from the text 'you would choose the project field that determines which forms will be displayed.' to the 'sex - Gender' dropdown.
- An orange arrow points from the text 'you can also select the event that contains the control field.' to the 'Arm: Drug A - Event: Er' dropdown.
- A red arrow points from the text 'The 'Condition' is the value that's being evaluated.' to the '=' dropdown.
- A green arrow points from the text 'Target Forms are the forms that will appear only if the condition is true.' to the 'patient\_morale\_questionnaire' dropdown.
- A blue arrow points from the text 'You can restrict the FRSL rules to specific events as well (i.e., different time/event tables for subject types).' to the '1.1. Restrict this rule to specific events:' checkbox.

## Default (Control Field) Configuration Example:

For the example below, there are two forms that need to appear for all animal species (*Animal Identification* and *Lifestyle & Reproductive Data*) and then there is an animal-specific form that should only appear if that animal species is selected (i.e., the *Cat Data* form if cat is selected as the animal species).

The image shows a configuration interface for a form. It consists of three main parts:

- Species Selection:** A table with columns 'species' and 'Animal species'. A radio button is shown next to 'Animal species' with a list of options: 1 Cat, 2 Jellyfish, 3 Frog, 4 Marlin, 5 Snake, 6 Butterfly, 7 Parrot.
- Instrument List:** A table listing instruments with their names, field counts, PDF icons, and actions.
- Project Settings:** A configuration panel for 'Project Settings' with a 'Value' column. It shows branching logic for 'Control fields'.

Arrows indicate the configuration flow:

- A cyan arrow points from the 'Animal species' radio button to the 'Control fields' section.
- A black arrow points from the 'Cat' radio button to the '1.1. Condition' field.
- A black arrow points from the 'Cat' radio button to the '1.1.1. Target forms' field.
- A green arrow points from the 'Cat Data' instrument to the '1.1.1. Target forms' field.
- A black arrow points from the 'Jellyfish' radio button to the '1.2. Condition' field.
- A green arrow points from the 'Jellyfish data' instrument to the '1.2.1. Target forms' field.

Instrument name	Fields	View PDF	Instrument actions
Animal Identification	7		Choose action
Lifestyle & Reproductive data	9		Choose action
Cat Data	9		Choose action
Jellyfish data	8		Choose action
Frog data	9		Choose action
Marlin data	8		Choose action
Snake data	7		Choose action
Butterfly data	9		Choose action
Parrot data	10		Choose action

Project Settings	Value
Prevent hiding of filled forms:	<input type="checkbox"/>
1. Control fields:	+ Default (Control field) Advanced (Equation with Piping and/or Smart Variables)
1. Fields:	species - Animal species Arm: Arm 1 - Event: Ev
1. Default/fallback value:	
1.1. Branching logic:	+ -
1.1. Condition:	= 1
1.1.1. Target forms:	cat_data +
1.1. Restrict this rule to specific events:	<input type="checkbox"/>
1.2. Branching logic:	+ -
1.2. Condition:	= 2
1.2.1. Target forms:	jellyfish_data +

Note that Cat is coded as '1' and Jellyfish is coded as '2' in the [species] data field, which is why '1' and '2' are used as the 'Conditions' to control the species-specific forms. If the value of '1' (Cat) is selected species, then the Cat Data form should be shown, while hiding the other species-specific forms.

**Record Status Dashboard View Example:**

Notice that only the specific animal data forms are accessible for each animal type (i.e., the *Cat Data* form only appears when the subject is a cat), while the non-specific animal forms (*Animal identification* and *Lifestyle & Reproductive Data*) appear for all animal types. The type of animal is selected on the *Animal Identification* form in this example.

Record ID	Forms not controlled by FRSL		Forms controlled by FRSL						
	Animal Identification	Lifestyle & Reproductive data	Cat Data	Jellyfish data	Frog data	Marlin data	Snake data	Butterfly data	Parrot data
1 (Cat)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2 (Frog)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 (Butterfly)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4 (Butterfly)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 (Frog)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
6 (Snake)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
7 (Jellyfish)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

### Record Home Page/Data Entry View Example:

Notice that only the specific animal data forms are accessible for each animal type (i.e., the *Cat Data* form only appears when the subject is a cat), while the non-specific animal forms (*Animal identification* and *Lifestyle & Reproductive Data*) appear for all animal types.

Record ID 1 (Cat)

Data Collection Instrument	Status
Animal Identification	<input checked="" type="radio"/>
Lifestyle & Reproductive data	<input checked="" type="radio"/>
Cat Data	<input checked="" type="radio"/>
Jellyfish data	<input type="radio"/>
Frog data	<input type="radio"/>
Marlin data	<input type="radio"/>
Snake data	<input type="radio"/>
Butterfly data	<input type="radio"/>
Parrot data	<input type="radio"/>

Unneeded forms are not accessible

The screenshot shows a sidebar titled 'Data Collection' with an 'Edit instruments' link. It contains a 'Record Status Dashboard' section with 'Add / Edit Records' and a 'Record ID 1 (Cat)' section with a 'Select other record' link. Under 'Data Collection Instruments', 'Animal Identification' is selected and highlighted in blue, while 'Lifestyle & Reproductive data' and 'Cat Data' are also selected with green radio buttons. Other instruments like 'Jellyfish data', 'Frog data', 'Marlin data', 'Snake data', 'Butterfly data', and 'Parrot data' are greyed out and have unselected radio buttons.

## Configuring the Module: Advanced Method (Equation with Piping and/or Smart Variables):

The Advanced (Equation with Piping and/or Smart Variables) method uses a project's metadata (not a data field within the project) to determine which forms to display or not display. For example, if you want to hide or reveal forms based on the subject's Data Access Group (DAG), you would use the Smart Variable [record-dag-name] in the 'Equation / Piping' section, list the DAG name (which can be found on the DAG page within the project) as the 'Condition' and then select which form(s) should only appear for that DAG under the 'Target Forms.' Any forms not listed under the Target Forms section will appear for all DAGs.

### 2) Advanced

Utilizes Smart Variables, piping or equations, instead of a single data field to determine forms.

'Default/fallback value' (optional) is the value all records will have until the control field/condition has data.

Checking 'Prevent hiding of filled forms' will show forms that already have saved data, even though they should now be hidden by FRSL.

Project Settings	Value
Prevent hiding of filled forms:	<input type="checkbox"/>
1. Control mode:	<input type="radio"/> Default (Control field) <input checked="" type="radio"/> Advanced (Equation with Piping and/or Smart Variables)
1. Equation / Piping:	<input checked="" type="checkbox"/> Smart Variables <input checked="" type="checkbox"/> Piping How do I format an equation? [record-dag-name]
1. Default/fallback value:	
1.1. Branching logic:	<input type="button" value="+"/>
1.1. Condition:	- uf_dag
1.1.1. Target forms:	patient_morale_questionnaire <input type="button" value="+"/>
1.1. Restrict this rule to specific events:	<input type="checkbox"/>

## Advanced Method (Equation with Piping and/or Smart Variables) Configuration Example Using the Data Access Group (DAG) Name to Determine Which Forms to Display:

Instrument name	Fields	View PDF	Enabled as survey
Informed Consent	28		
UF Addendum	16		
MedPlaza Addendum	16		
TowerHill Addendum	16		
SpringHill Addendum	16		
Miami Addendum	14		
Daytona Addendum	14		
Ocala Addendum	14		

1. Control fields: +

1. Control mode:   
 Default (Control field)   
 Advanced (Equation with Piping and/or Smart Variables)

1. Equation / Piping:   
[record-dag-name]

1. Default/fallback value:

1.1. Branching logic: +  
-

1.1. Condition:

1.1.1. Target forms:  +

1.1. Restrict this rule to specific events:

1.2. Branching logic: +  
-

1.2. Condition:

1.2.1. Target forms:  +

The record's DAG will determine which ICF addendum is displayed to the participant.

'Auto-Continue to the next survey' is used for the Informed Consent survey.

Note that all subjects in this example will fill out the *Informed Consent* survey and then will receive the DAG-specific survey (addendum) once they submit the *Informed Consent* survey. The 'Auto-continue to the next survey' survey-termination feature can be used for the *Informed Consent* survey so that the subject goes straight to the site-specific Addenda survey – even if the 'next' survey is not relevant to the subject. FRSL will skip over the subsequent survey(s) and take them to the correct 'next' survey. For example, if the subject is in the MedPlaza DAG, FRSL will skip the UF DAG Addendum survey and take them straight to the MedPlaza Addendum survey.

## Multiple Control Fields and/or Conditions:

To add more than one condition/control mode/target form, click on the [+] signs next to that section to add more entries.

To add more Branching Logic conditions:

Project Settings	Value
Prevent hiding of filled forms:	<input type="checkbox"/>
1. Control fields: <span style="float: right;">+</span>	
1. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)
1. Field:	sex - Gender <input type="button" value="v"/> Arm: Drug A - Ev <input type="button" value="v"/>
1. Default/fallback value:	<input type="text"/>
1.1. Branching logic: <span style="float: right;">+</span>	
1.1. Condition:	= <input type="button" value="v"/> 0 <input type="text"/>
1.1.1. Target forms:	visit_lab_data <input type="button" value="v"/> <span style="float: right;">+</span>
1.1. Restrict this rule to specific events:	<input type="checkbox"/>
1.2. Branching logic: <span style="float: right;">+ -</span>	
1.2. Condition:	= <input type="button" value="v"/> <input type="text"/>
1.2.1. Target forms:	<input type="button" value="v"/> <span style="float: right;">+</span>
1.2. Restrict this rule to specific events:	<input type="checkbox"/>

To add more Control Fields:

Project Settings	Value
Prevent hiding of filled forms:	<input type="checkbox"/>
<b>1. Control fields:</b> <span style="float: right;">+</span>	
1. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)
1. Field:	sex - Gender <input type="text"/> Arm: Drug A - E <input type="text"/>
1. Default/fallback value:	<input type="text"/>
<b>1.1. Branching logic:</b> <span style="float: right;">+</span>	
1.1. Condition:	= <input type="text"/> 0 <input type="text"/>
1.1.1. Target forms:	visit_lab_data <input type="text"/> <span style="float: right;">+</span>
1.1. Restrict this rule to specific events:	<input type="checkbox"/>
<b>2. Control fields:</b> <span style="float: right;">+ -</span>	
2. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)
2. Field:	<input type="text"/> <input type="text"/>
2. Default/fallback value:	<input type="text"/>
<b>2.1. Branching logic:</b> <span style="float: right;">+</span>	
2.1. Condition:	= <input type="text"/>
2.1.1. Target forms:	<input type="text"/> <span style="float: right;">+</span>
2.1. Restrict this rule to specific events:	<input type="checkbox"/>

## **Using the ‘Restrict this rule to specific events’ feature:**

Using this feature is very helpful when subjects have different time & event tables, i.e., those in the intervention group versus the control group, or when different forms need to be filled out for different sets of subjects, i.e., males versus females, in a longitudinal project.

Normally, you would need to put male and female subjects, or control and intervention subjects, into different arms so their time and event tables can differ or so that different forms can be assigned to different subject types. Using FRSL, you can put all subject types into one arm and then just hide the forms that are not needed. You can also hide all forms from an event if the subject type does not have any data collected at that event, while some subject types do.

In this example, males and females get labs drawn at different time points, so you want to hide the Labs form at the time points they are not needed, depending on the subject’s sex. The females get labs drawn at ‘Visit 2’ and at the ‘Final Visit.’ The males get labs drawn at ‘Visit 1’ and ‘Visit 3.’

Note: Sex is a radio button field on the *Demographics* form, with females coded as ‘0’ and males as ‘1’.

Configuration example for the female subjects:

1. Control fields:		+ -
1. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)	
1. Field:	sex - Gender <input type="button" value="↕"/> Arm: Drug A - Ev <input type="button" value="↕"/>	
1. Default/fallback value:	<input type="text"/>	
1.1. Branching logic:		+ -
1.1. Condition:	= <input type="button" value="↕"/> 0 <input type="text"/>	
1.1.1. Target forms:	visit_lab_data <input type="button" value="↕"/>	+ -
1.1. Restrict this rule to specific events:	<input checked="" type="checkbox"/>	
1.1.1. Events:	Arm: Drug A - Event: Visit 2 <input type="button" value="↕"/>	+ -
1.1.2. Events:	Arm: Drug A - Event: Final visit <input type="button" value="↕"/>	+ -

Configuration example for the male subjects:

2. Control fields:		+ -
2. Control mode:	<input checked="" type="radio"/> Default (Control field) <input type="radio"/> Advanced (Equation with Piping and/or Smart Variables)	
2. Field:	sex - Gender <input type="text"/> Arm: Drug A - Ev <input type="text"/>	
2. Default/fallback value:	<input type="text"/>	
2.1. Branching logic:		+
2.1. Condition:	= <input type="text" value="1"/>	
2.1.1. Target forms:	visit_lab_data <input type="text"/>	+
2.1. Restrict this rule to specific events:	<input checked="" type="checkbox"/>	
2.1.1. Events:	Arm: Drug A - Event: Visit 1 <input type="text"/>	+ -
2.1.2. Events:	Arm: Drug A - Event: Visit 3 <input type="text"/>	+ -

Here's what a female subject's time and event table will look like with FRSL in place:

**Study ID 1 (Female)**

 Data Collection Instrument	Enrollment	Visit 1	Visit 2	Visit 3	Final visit
Demographics	<input checked="" type="radio"/>				
Contact Info	<input type="radio"/>				
Baseline Data	<input type="radio"/>				
Visit Lab Data		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Patient Morale Questionnaire		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visit Blood Workup		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visit Observed Behavior		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completion Data					<input type="radio"/>
Completion Project Questionnaire					<input type="radio"/>
Delete all data on event:	<input checked="" type="checkbox"/>				

Notice that the Labs form is now hidden for events Visit 1 and Visit 3.

Here's what a male subject's time and event table will look like with FRSL in place:

Study ID 2 (Male)

<input type="checkbox"/> Data Collection Instrument	Enrollment	Visit 1	Visit 2	Visit 3	Final visit
Demographics	<input checked="" type="radio"/>				
Contact Info	<input type="radio"/>				
Baseline Data	<input type="radio"/>				
Visit Lab Data		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Patient Morale Questionnaire		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visit Blood Workup		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Visit Observed Behavior		<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Completion Data					<input type="radio"/>
Completion Project Questionnaire					<input type="radio"/>
Delete all data on event:	<input checked="" type="checkbox"/>				

Notice that the Labs form is now hidden for events Visit 2 and Final Visit.

### Using the 'Prevent hiding of filled forms' feature:

Using this feature is very helpful when the module has been activated *after* data collection has begun, and forms that should not have been filled out were. Since the FRSL module will hide any unneeded forms, if data entry has begun on a form that should now be hidden, you will no longer have access to this unneeded form, which might contain data that is needed. By enabling this feature, you can still have access to any unneeded form that *contains data*. If the form does not contain data, and the form is not needed, then the form will remain hidden and not accessible.

## Configure Module: Form Render Skip Logic

---

### Project Settings

### Value

---

Prevent hiding of filled forms:



In the example below, a user started filling out the Jellyfish Data form for Record ID 1, even though the animal is a cat, so the Jellyfish Data form is still visible and accessible in Record ID 1 since it contains saved data. For the remainder of the records, the form stays hidden unless the animal species is a jellyfish.

Record ID	Animal Identification	Lifestyle & Reproductive data	Cat Data	Jellyfish data	Frog data	Marlin data	Snake data	Butterfly data	Parrot data
<u>1</u> (Cat)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>2</u> (Frog)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>3</u> (Butterfly)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
<u>4</u> (Butterfly)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
<u>5</u> (Frog)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>6</u> (Snake)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>7</u> (Jellyfish)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>8</u> (Marlin)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
<u>9</u> (Parrot)	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>

Questions? Contact the REDCap Support Team at: <https://redcap.ctsi.ufl.edu/redcap/surveys/?s=DUPrXGmx3L>