2.4.5 Form Logic

Adding logic to your forms can help make them "smart" by improving the user experience, promoting cleaner data, and automating calculations.
Form logic includes:

- calculations
- cross-form logic
- skip logic
- constraints

When writing expressions for form logic it is necessary to use the correct syntax, including punctuation and spaces. Form logic is not case-sensitive.

Referencing Items

There are two ways to reference an item on your form within a logic expression. Values can either be an item reference (e.g. **age**) or the literal value of an item (e.g. **yes** or **1**).
[table id=55 /]
To use values from other forms see below.

Conditions

Note: *You can use either single quotes or double quotes, but single quotes are generally preferred for clarity. You would use double quotes if the string already contained a single quote. If you want this condition to work with multiple choices, use the formula twice with **and** / **or**.*

[table id=56 /]

Operators & Boolean Logic

[table id=57 /]
Boolean operators (**and** & **or**) return a value of either True or False. If you have a sequence of statements, all linked together with **and**, then to have a value of True every statement must be true; if even one is not true, then the whole expression's value is False. With **or**, the threshold is much lower: only one of the statements needs to be true to evaluate to True (though any number can be true, as long as at least one is).

Parentheses control the order of operations of a statement and are often used even when not needed to clarify and to increase the readability of the code (e.g.(${enddate}-${startdate}) * 2). Lastly, modulus (**mod**) divides one number into another and returns the remainder. This has many uses well beyond the EDC realm, but one of the simplest examples is that many programs use it to tell when a number is even or odd (if mod 2 is 0 it's even, 1 it's odd), or to know if a number is equal to, or a multiple of, another number (in this case if mod-ing by say 7, any value that results in 0 would be a multiple of 7).

Calculations

Enter the formula from the syntax column in the **Calculation** field (Form Designer) or the

Calculate column (Form Template). Replace the items in brackets with the names of your own items.

[table id=59 /]

Required

Enter the formula from the syntax column in the field that appears when you select **Conditional** in the **Required** field (Form Designer) or the **Required** column (Form Template). Replace the items in brackets with the names of your own items. **Required** conditions are only evaluated for items that are currently null (for example, no value has been entered).

[table id=62 /]

Skip Logic/Relevant

Enter the formula from the syntax column in the field that appears when you select **Skip Logic** and click the **${} Manually enter your skip logic in XLSForm Code** button (Form Designer) or the **Relevant** column (Form Template). Replace the items in brackets with the names of your own items.

[table id=60 /]

Relevant logic is ignored if the default column is populated for an item in a form. If an item has a default value, it populates when the form is opened or when a repeat is added regardless of whether the item is relevant or not. To avoid default values displaying before an item becomes relevant, use triggered calculations so the values don't populate until the item becomes relevant, or use an if() statement. To use an if() statement in this case, use a default that has an if() statement that sets the value to null if the relevant condition is not met and sets it to the desired default if the relevant condition is met.

Validation Criteria/Constraint

Enter the formula from the syntax column in the **Constraint** field (Form Designer) or the **Constraint** column (Form Template). Replace the items in the syntax column in brackets with the names of your own items. **Constraint** conditions are only evaluated for items that are currently non-null (for example, a value has been entered).

[table id=61 /]

Advanced Form Logic

Hard Edit Checks (Form Template)

Hard edit checks are exclusive to the Form Template and can be used to enforce stricter standards on data entry than soft edit checks. Because they can make data entry more difficult for users, it is recommended that they be used only when the benefits to data quality outweigh the difficulties for the study personnel.

All checks that are not explicitly defined as hard edit checks are soft edit checks in all forms. The exception to this rule are forms used in the Participate module in which case all edit checks are automatically hard edit checks.

When a value is entered and it violates a hard check on an item, the value is rejected before it is stored, and the user will see a pop-up message. If the value entered causes a hard edit check to be violated on a different item (such as one already entered and saved that passed the validation criteria at the time) then the user will see an error message displayed on the item with the violated hard check, and it will have an orange background (distinct from the soft red background used for soft check error messages).

A user cannot navigate forward in the form or mark the form complete while a hard check error message is present. The user can add a manual or automatic query to a hard check item to close the form. However, the query will not hide the hard check error message.

Important: Special care must be taken when using relevant logic (hide/show) for an item with a hard required check. Once an item has a value, a required hard check will not allow the item to be cleared. To avoid this, the item's required logic must indicate that the item is required only when the item's relevant logic is met. If this is not implemented properly, the user may get into a state where they cannot clear an item even though that is the only way to make the data consistent with the relevant logic.

For example, if the item (or the group it is in) has relevant logic **${yn_item} = 1**, then the hard required logic must also include **${yn_item} = 1** to ensure that the form works as intended. That will enforce the hard required check when the item is relevant, but will not enforce the check when the item is not relevant.

To Make a Hard Edit Check for a Required Item:

1. Click the **survey** tab in the Form Template spreadsheet.
2. Add a column with the header **bind::oc:required-type**.
3. Enter **strict**.

To Make a Hard Edit Check for a Constraint:

1. Click the **survey** tab in the Form Template spreadsheet.
2. Add a column with the header **bind::oc:constraint-type**.
3. Enter **strict**.

Cross-Checks (Form Template)

Referencing item values across forms and/or events

Reference the following examples when checking data across events, forms, repeating occurrences in a single event, or repeating occurrences in separate events. There are also examples of how to check specific values captured when a user adds a new participant.

The following steps apply all of the examples below, unless otherwise stated:

1. Create an item with a **type** of **calculate**.
2. In the **bind::oc:external** cell, for that newly added calculate field, select **clinicaldata** from the drop-down list.
3. In the **calculation** cell for the new **calculate** item, copy and paste the appropriate sample text from the examples below, and replace the bold, italicized text with the OIDs (or object names) from your study.
4. Reference the new **calculate** field (as defined above) in any of the following cells to use the

externally referenced value for display or in a logic expression:
    - **label**
    - **hint**
    - **calculation**
    - **constraint**
    - **required**
    - **relevant**
5. (Optional) To optimize performance and enhance data privacy, configure the **crossform_references** column on the **survey** sheet to include only the events that are needed for the cross-form logic on your form. Cross-form logic will work without this step, but the form will load more slowly as the full participant record is processed through the browser. Simply enter a comma-separated list of the Event OIDs used in your logic. Include **current_event** in the list if any logic on the form uses [@OpenClinica:Current='Yes']. With this configuration used, only the indicated events are processed through the browser.

For example, you would enter "SE_BASELINE,SE_VISIT2,current_event" if your form uses cross-form logic containing the events, "SE_BASELINE", "SE_VISIT2", and "[@OpenClinica:Current='Yes']".

Notes:

- *Even if your logic references an event by name instead of OID, you must include the **Event OID** in the list.*
- *If you use any of the examples that reference OIDs, the study must first be published in order to generate OIDs. You can publish the study to the test environment, then locate and reference the OIDs as needed (they do not change between the test & production environments), and include those OIDs in the calculation field as indicated below.*
- *You can reference object names instead of OIDs, but if the object name changes, you must update the calculation to include the updated object name. OIDs do not change for objects, so it is easier to maintain forms that include references to OIDs.*
- *You can create any single calculation by using a combination of OIDs and object names.*

Cross-Form Usage Tips:

- Cross-form items defined using the method below will start out as null every time the form is opened or reopened. The cross-form logic will be processed immediately after the form opens to retrieve the expected values. This means that any other form logic (such as calculations or relevant logic) that is dependent on cross-form values will initially respond to the cross-from item as null and then subsequently respond to the retrieved cross-form value. Care should be taken in form design to account for this behavior.
- To prevent a form item that is being set to a cross-form value from being calculated as null briefly during form load, instead of using a calculation like **${crossform_item}**, consider using a calculation like **if(${crossform_item} = '', ., ${crossform_item})**. This will update the target item to the cross-form value, but it will not allow it to be set to null if it has been calculated to a non-null value.
- For calculated values that are based on data not contained within the form (for example, randomization values added by the Randomize module, values imported from another system as part of a data migration, or randomly generated numbers), do not put these calculated values under **relevant** logic that is dependent on cross-form values. The temporary null cross-from value during form load may cause the **relevant** logic to behave unexpectedly and may cause these calculated values to be cleared. Instead, ensure calculated values that are based on data from outside the form are either not under **relevant** logic or their **relevant** logic is not

dependent on cross-from values.

Cross-check data against an item value in another event. Referencing the item value by item name

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following to the **calculation** cell, replacing the bold, italicized text with your specific object names:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@OpenClinica:EventName='***Event Name Here***']/FormData[@OpenClinica:FormName='***Form Name Here***']/ItemGroupData[@OpenClinica:ItemGroupName='***Item Group Name Here***']/ItemData[@OpenClinica:ItemName='***Item Name Here***']/@Value

Cross-check data against an item value in another event. Referencing the item value by OID

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following to the **calculation** cell, replacing the bold, italicized text with your specific OIDs:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@StudyEventOID='***Event OID Here***']/FormData[@FormOID='***Form OID Here***']/ItemGroupData[@ItemGroupOID='***Item Group OID Here***']/ItemData[@ItemOID='***Item OID Here***']/@Value

Cross-check data against an item value in a different form that is in the same event as this form

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following to the **calculation** cell, replacing the bold, italicized text with your specific OIDs:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@OpenClinica:Current='Yes']/FormData[@FormOID='***Form OID Here***']/ItemGroupData[@ItemGroupOID='***Item Group OID Here***']/ItemData[@ItemOID='***Item OID Here***']/@Value

Cross-check data against an item value in a specific repeat/row of a repeating group

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following to the **calculation** cell, replacing the bold, italicized text with your specific OIDs:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@StudyEventOID='***Event OID Here***']/FormData[@FormOID='***Form OID Here***']/ItemGroupData[@ItemGroupOID='***Item Group OID Here***'][***Repeat Number Here***]/ItemData[@ItemOID='***Item OID Here***']/@Value

Cross-check data against a value from a specific event occurrence in a repeating event

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following to the **calculation** cell, replacing the bold, italicized text with your specific OIDs:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@StudyEventOID='***Event OID Here***'][***Event Repeat Number Here***]/FormData[@FormOID='***Form OID***

Here']/ItemGroupData[@ItemGroupOID='***Item Group OID Here***']/ItemData[@ItemOID='***Item OID Here***']/@Value

Cross-check data against the event start date for the current event

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following text to the **calculation** cell:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@OpenClinica:Current='Yes']/@OpenClinica:StartDate

Cross-check data against the event start date in a different event

Add a **calculate** item. Include '**clinicaldata**' in the **bind::oc:external** cell and add the following text to the **calculation** or **default** cell, replacing the bold, italicized text with your specific Event OID:

substr(instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@StudyEventOID='***Event OID Here***']/@OpenClinica:StartDate, 0, 10)

Note: As noted in the previous example, you can also read in the full Event Start Date and Time for display purposes using:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@StudyEventOID='**Event OID Here**']/@OpenClinica:StartDate

Cross-check using the Study Event OID of the Event the current form is in

Add a calculate item. Include 'clinicaldata' in the bind::oc:external cell and add the following text to the calculation or default cell:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@OpenClinica:Current='Yes']/@StudyEventOID

Note: This is very useful for reusing a form in several events and using relevant logic to hide/show some items only when the form is opened for specific events.

Cross-check using the Study Event Repeat Key of the Event repeat occurrence the current form is in

Add a calculate item. Include 'clinicaldata' in the bind::oc:external cell and add the following text to the calculation or default cell:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/StudyEventData[@OpenClinica:Current='Yes']/@StudyEventRepeatKey

Note: Include "current_event" in crossform_references to improve the form's performance. See #5 above for more information.

Cross-Check to Reference User Role or Username

You can add a **calculate** item to your form to store the User Role or Username of the current user. You can then control which **note** items that user can see on a form by referencing that **calculate**

item in the **relevant** cell for the **note** item.

For example, you may have a **note** item on your form that provides instructions to the CRA regarding what items must be verified for a form that only requires partial verification.

It is not necessary for any other user roles to see that note, so you can conditionally display the note to Monitors only.

To do this, follow the steps below:

1. Use the instructions below to Lookup User Role.
2. Include the **note** item with instructions for SDV.
3. In the **relevant** cell for the **note**item, reference the **calculate** item that now stores the current user's role or username.

For example, ${currentrole}="Monitor" (where currentrole is the name of the calculate item you created)

Note: *Only the standard roles are available for reference. If you created a custom role of "Junior Monitor" based on the Monitor role, this only references the standard Monitor role. In the future, custom role names will be available for reference as well.*

Lookup User Role

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following text to the **calculation** cell:

instance('clinicaldata')/ODM/ClinicalData/UserInfo/@OpenClinica:UserRole

Lookup Username

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following text to the **calculation** cell:

instance('clinicaldata')/ODM/ClinicalData/UserInfo/@OpenClinica:UserName

Lookup Participant ID

Add a **calculate** item. Include **'clinicaldata'** in the **bind::oc:external** cell and add the following text to the **calculation** cell:

instance('clinicaldata')/ODM/ClinicalData/SubjectData/@OpenClinica:StudySubjectID

Lookup Reference Study/Site OID

Add a **calculate** item. Include '**clinicaldata**' in the **bind::oc:external** cell and add the following text to the **calculation** or **default** cell

instance('clinicaldata')/ODM/ClinicalData/@Study acOID

Note: If the Participant exists at the Study-level, this will return the Study OID. If the Participant exists at the Site-level, this will return the Site OID. The OID returned by this would need to be processed further if the Study/Site ID is needed.

Functional approval by Paul Bowen. Signed on 2026-03-17 10:43PM

Approved for publication by Kate Lambert. Signed on 2026-03-20 1:08PM

Not valid unless obtained from the OpenClinica document management system on the day of use.