

InfoCapture scenario: Making fields appear

Introduction

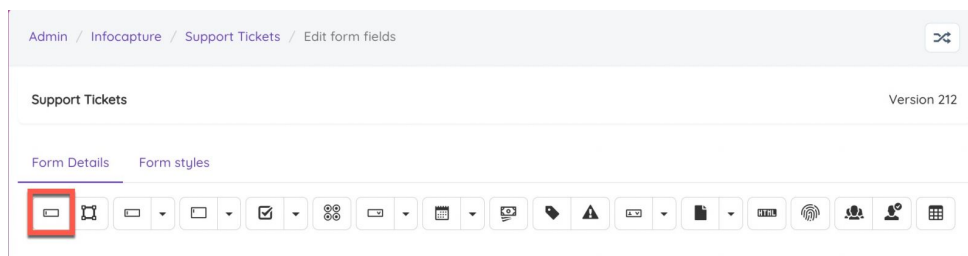
In this article, we will provide an understanding of creating field visibility in a practical form building sense.

For this scenario, we will be making a drop-down field to enable a Text field to appear when selecting one of the options in the drop-down.

This can be used for situations where you might want to create a 'type' field with a series of options to choose from. Filling it out with all types, however, might be an impossible task. So we can use the drop-down to highlight the most common types, but have an 'Other' option in the drop-down which can allow this text field to appear for people to add their own type making the whole process more streamlined.

Form Building

In order to create this process, two different types of fields need to be added to your form. From the Project Summary tab, check out the form to add the new fields.



1. You will need to add a drop-down 'Select' field and enter each of your values into the field as per usual.

Fields properties

Properties Style Constraints

Name: Type of Issue

Symbolic name: type_of_issue Edit

Optional Hint:

Type: Select

Values: Local Machine,#v001
Printer,#v002
Internet Connection,#v003
Monitor,#v004
Other,#v005

☐ Use advanced UI for selecting elements

Default value: Local Machine

☒ Show "Please select..." as default

☐ Required

☐ Disabled

☒ Reload form on changing

Save Delete

Please note: A very important aspect when creating this field is making sure 'Reload form on changing' has been ticked - this feature will make the form refresh mid submission allowing it to check for any rules attached to this field. Without this selected, the field visibility rule will not work.

2. Once this has been applied, we can add the 'Text' field. The field should be positioned directly below the drop-down field as the field will appear where it's placed in the form structure and could disrupt the form structure otherwise.

Fields properties

Properties Style Constraints

Name: Issue Details

Symbolic name: issue_details Edit

Optional Hint:

Type: Medium text

Default value:

Use existing variables

☐ Required

☐ Disabled

☐ Reload form on changing

Save Delete

Once these fields have been added, we can check the form to prepare for the next functions.

Please note: At this point, there are no rules in place for this functionality so the intended hidden field will show up at all times when filling out a form. It is strongly recommended that the form is not live when creating this functionality and the form is either in testing or blocked until all changes have been made.

Field Condition Sets

With the new fields in place, it's time to set the foundations for the actual functionality to occur. To do this, we need to create a new field condition set for the fields we have created.

Simply put, a field condition set is a rule to determine that making a field equal to a certain value will allow certain functionality to happen within the form. A field condition set can include multiple fields to make a much more complex rule.

Project options

Project Summary

Edit project properties

Project permissions

Statuses

Conditions

Field condition sets

Triggers

Field condition sets

Set true or false statement based on whether or not a condition is currently being met within your form.

Field condition sets are used in four places: 'Field visibility', 'Workflow', 'SLAs', and sometimes in 'Triggers'. These four functions are dependant upon conditions being met, which are defined here. Two conditions are included by default with new projects: 'Default (always)' is a condition that is always being met. This is useful for field visibility. 'Default (being reported)' is a condition that is met only while the user is submitting the form.

+ Add new condition set

Name / Order #	Field conditions	Use for fields rights	Use for workflow	Use for SLA
Default (Always)		Yes	Yes	Yes
Default (Being reported)	Being reported	Yes	Yes	Yes

1. To keep this simple, all we need to do at this point is to give the condition set a Name and tick the 'Use for field rights' box.

Admin / Infocapture / Project properties / Field condition sets / Edit fields conditions set

Demo Form B

Properties of Field condition set

Name

Type of Issue = Other

☒ Use for fields rights

☐ Use for workflow

☐ Use for SLA

Field Conditions

Name	Type	Value
<input type="checkbox"/> Client Type	Radio	<div>New</div> <div>Existing</div>
<input type="checkbox"/> Name of New Client	Medium string [Text]	<div><input type="checkbox"/> ==</div> <div><input type="checkbox"/> >=</div> <div><input type="checkbox"/> <></div> <div><input type="checkbox"/> <=</div>
<input type="checkbox"/> Type of Issue	Select	<div>Local Machine</div> <div>Printer</div> <div>Internet Connection</div> <div>Monitor</div>

The condition name can be anything to make sense to you but the best practice is to name it 'Field name = Value' as when you come back to this in the future it will be clear which rule it is referring to.

2. Next, select the value of the field which should have something that looks like this.

Admin / Infocapture / Project properties / Field condition sets / Edit fields conditions set

Demo Form B

Properties of Field condition set

Name

Type of Issue = Other

☒ Use for fields rights

☐ Use for workflow

☐ Use for SLA

Field Conditions

Name	Type	Value
<input type="checkbox"/> Client Type	Radio	<div>New</div> <div>Existing</div>
<input type="checkbox"/> Name of New Client	Medium string [Text]	<div><input type="checkbox"/> ==</div> <div><input type="checkbox"/> >=</div> <div><input type="checkbox"/> <></div> <div><input type="checkbox"/> <=</div>
<input checked="" type="checkbox"/> Type of Issue	Select	<div>Printer</div> <div>Internet Connection</div> <div>Monitor</div> <div>Other</div>

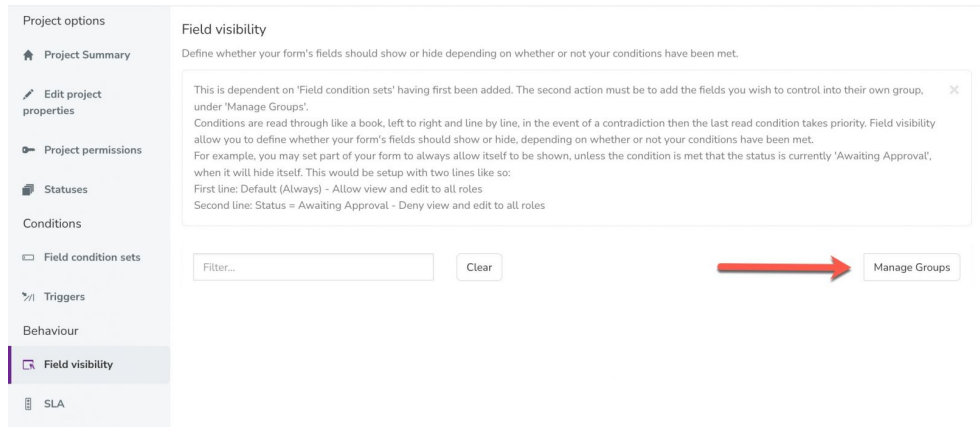
Please note: As mentioned above, the condition set is simply the foundation of the rule and creating this logic will still not create the desired functionality there is one final step that needs to be taken.

Field Visibility

With our fields added to the form and our condition set created we can finally put in place a rule to hide our text field until the desired option from the drop-down field above has been selected.

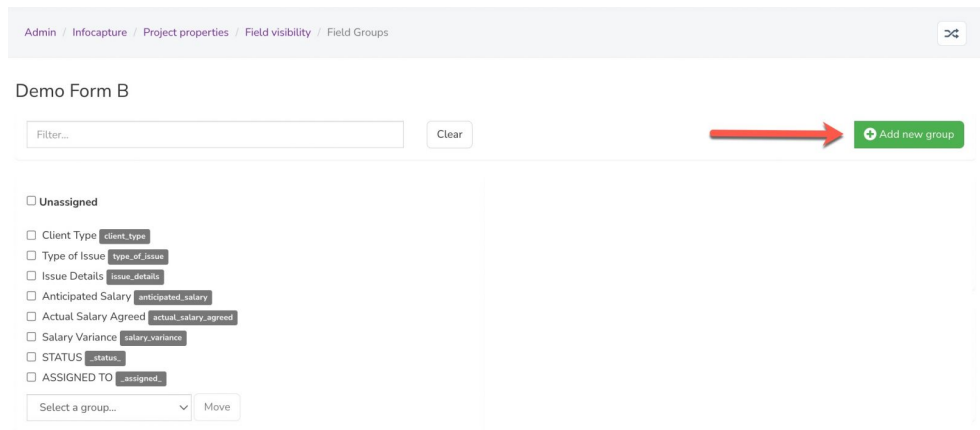
In order to do this, we will need to go to the field visibility section of our form. This aspect of InfoCapture can eventually get quite complex but for the purpose of this scenario, we are going to keep this as simple as possible.

The first time you access the field visibility this will be completely empty with only the option to 'Manage Groups' which will take us to the manage group section where we can create groups for fields we want to make field visibility rules for.



Reminder: Again this can get quite complicated so we will only be creating a single group for the Text field we are planning to hide. Whenever creating these groups always remember it is the field you wish to hide that will be added to the group, not the field that will be used to impact the visibility.

1. Simply press 'Add New Group' and name the group.



2. Select the field we will be hiding that will be currently be positioned in the 'Unassigned' column then select the group in the drop-down below this column. Press the 'Move' button and save the changes.

Admin / Infocapture / Project properties / Field visibility / Field Groups

Demo Form B

Filter... Clear Undo Save Add new group

☐ Unassigned

☐ Client Type client_type
☐ Type of Issue type_of_issue
☒ Issue Details issue_details
☐ Anticipated Salary anticipated_salary
☐ Actual Salary Agreed actual_salary_agreed
☐ Salary Variance salary_variance
☐ STATUS status
☐ ASSIGNED TO assigned_to

Hidden Group: Other Type

Move

Upon returning to the field visibility page you will see the below group. This new item is a representation of the group we have created and allows us to add field condition sets to the group to allow all Project Roles in the form to be able to either view or edit the field based on these condition sets.

Hidden Group: Other Type											
	All		Users		Admin		Submitter of ticket		Ticket handler		
Field condition set	View	Edit	View	Edit	View	Edit	View	Edit	View	Edit	
+ Add Condition Set											

3. Now press 'Add condition set' to apply some condition sets to this group to enable this functionality. We will be using the field condition set called Default (Always) and the field condition set we previously created in order to help achieve this functionality.

Hidden Group: Other Type											
	All		Users		Admin		Submitter of ticket		Ticket handler		
Field condition set	View	Edit	View	Edit	View	Edit	View	Edit	View	Edit	
= Default (Always)	Not set	Not set	Not set	Not set	Not set	Not set	Not set	Not set	Not set	Not set	
+ Add Condition Set											

Please note: Default (Always) is a field condition set that will automatically be within your InfoCapture form as it is hardcoded into the construction of all IC forms

4. Where it says 'Not Set' under the 'All' column, change this to 'Deny' for both view and edit changing the row to look like this. Doing so has now changed the functionality of the form meaning that this field will always be hidden and no roles will be able to interact with this field in any way.

Hidden Group: Other Type											
	All		Users		Admin		Submitter of ticket		Ticket handler		
Field condition set	View	Edit	View	Edit	View	Edit	View	Edit	View	Edit	
= Default (Always)	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	
+ Add Condition Set											

5. Add the condition set we created earlier to this group and change the 'Not Set' option to 'Allow'. This has now created a rule in our form to say that by default this field will be hidden to all users unless the field we have defined in our condition set is selected.

Hidden Group: Other Type											
	All		Users		Admin		Submitter of ticket		Ticket handler		
Field condition set	View	Edit	View	Edit	View	Edit	View	Edit	View	Edit	
= Default (Always)	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	
= Issue Type: Other	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	
+ Add Condition Set											

6. Lastly, we need to save the changes made here by pressing the Save button.

Field visibility

Define whether your form's fields should show or hide depending on whether or not your conditions have been met.

This is dependent on 'Field condition sets' having first been added. The second action must be to add the fields you wish to control into their own group, under 'Manage Groups'.

Conditions are read through like a book, left to right and line by line, in the event of a contradiction then the last read condition takes priority. Field visibility allow you to define whether your form's fields should show or hide, depending on whether or not your conditions have been met.

For example, you may set part of your form to always allow itself to be shown, unless the condition is met that the status is currently 'Awaiting Approval', when it will hide itself. This would be setup with two lines like so:

First line: Default (Always) - Allow view and edit to all roles

Second line: Status = Awaiting Approval - Deny view and edit to all roles

Hidden Group: Other Type											
	All		Users		Admin		Submitter of ticket		Ticket handler		
Field condition set	View	Edit	View	Edit	View	Edit	View	Edit	View	Edit	
Default (Always)	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	Deny	
Issue Type: Other	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	Allow	
<input type="button" value="+ Add Condition Set"/>											

With this done you now have a fully functioning feature in your form allowing you to hide and reveal fields on your form as needed.

Please note: When creating these rules always make sure the Default (Always) rule is the top condition set as pictured above. Based on how InfoCapture checks the rules for field visibility if Default (Always) is at the bottom of the list the form will not check for different rules and always use the logic you set for this condition set.

Last modified on 17 July 2024 by [Veronica Kim](#)

Created on 20 April 2021 by [Hannah Door](#)

Tags: [infocapture](#), [infocapture scenario](#), [intranet](#), [user guide](#)