



Infocapture: Triggers

Triggers are similar to [field condition sets](#) in that they are denoting conditions under which something should behave in a specific way. The difference is that triggers may be **transitional states**.

Difference between condition sets and triggers:

Example	Field Condition Sets	Triggers
1	Field = A	
2	Field = B	
3	Field = C	
4		Field = changed from A to B
5		Field = changed from B to C
6		Field = changed from C to A

A reminder of when to use field condition sets, and when to use triggers:

Field Condition Sets	Triggers
Field Visibility sets	Notifications
SLA	Automatic Changes
Workflow	
Within Triggers	

Example triggers that could be useful in your own forms:

- A new ticket has been submitted
- A note is added to the ticket
- A specific field has been changed by a user editing the ticket
- When there has been no change in a specific field on saving a ticket edit
- When field condition sets have been met on saving, or were not met before but now are on saving.

Triggers can be used in [automatic changes](#) to assign users or change the ticket status, as well as to send [notifications](#).

Based on the above options there is vast scope to implement different use cases and custom curate a form for your needs.

If you are ever unsure of how to best set up your triggers to achieve something, let us know if [a support ticket](#) and we can advise further :)

Creating Triggers

Head to Admin > Infocapture > (your project) > Triggers

Tick to specify which rule (or rules) you would like to include.

Here's an example trigger:

The screenshot shows the 'Add/Edit a trigger' interface. At the top, the breadcrumb is 'Admin / Infocapture / Bug Tracker / Triggers / Add/Edit a trigger'. The 'Trigger properties' section has a 'Name' field containing 'New ticket + Minor'. Below this is a 'Rules' section with a table:

Name	Value
<input checked="" type="checkbox"/> Ticket activity	<input checked="" type="radio"/> New ticket submitted <input type="radio"/> Public or private note has been added <input type="radio"/> Public note has been added <input type="radio"/> Private note has been added <input type="radio"/> File uploaded
<input type="checkbox"/> Any of the following fields changed	Type (type) Reproducibility (reproducibility) Severity (severity) Priority (priority)
<input type="checkbox"/> None of the following fields changed	Type (type) Reproducibility (reproducibility) Severity (severity) Priority (priority)
<input type="checkbox"/> Condition was	Default (Always)
<input type="checkbox"/> Condition was not	Default (Always)
<input checked="" type="checkbox"/> Condition is now	severity = minor
<input type="checkbox"/> Condition is not now	Default (Always)

At the bottom, the 'Notification type' is set to 'Email' (selected over 'In-system'). A 'Save' button is at the bottom left.

It is capturing a scenario where a new ticket has been submitted and the field condition set for when a field called 'Severity' has the 'minor' option chosen.

Ticket Activity

There are five possible choices here, explained by their names as you can see above.

The trigger can fire only if it's a new form being submitted, or if notes or files are added.

Field changes

As above, you'll see two rules available that look for a change in a field - or no change!

This is useful if you want to keep an eye on a particular area of the form.

Let's say someone edits the form but changes a date field called 'needed by', you might like to receive a notification email when that happens.

Field condition matches

These rules are used for looking at [field conditions](#), and whether or not they are or were true or false.

The screenshot shows a section of the form with four rows of condition options:

<input type="checkbox"/> Condition was	Default (Always)
<input type="checkbox"/> Condition was not	Default (Always)
<input type="checkbox"/> Condition is now	Default (Always)
<input type="checkbox"/> Condition is not now	Default (Always)

It's generally good to use condition rules **as pairs**.

Above I have chosen *Condition is now = Severity is Minor*

As my form moves through its various statuses, every time the form is saved or changed, that condition would still be true and the trigger would 'fire' every time. Then any other logic applied to this trigger will keep firing every time the ticket is saved. I might then receive an email notification every single time the form is changed!

Therefore it's important that I only want to be looking at the **exact point at which that change occurs** to avoid potential disruption or the form not working as I want it to.

So, I want my trigger to look for the change from when it *wasn't* true that 'Minor' was selected, to the point at which it *is* true that 'Minor' has been selected. That way, the trigger would only ever fire once - when the ticket is submitted.

I should create a trigger with these rules included to achieve this:

1. Condition WAS NOT = Severity is Minor
2. Condition IS NOW = Severity is Minor

Name	Value
<input checked="" type="checkbox"/> Ticket activity	<input checked="" type="radio"/> New ticket submitted <input type="radio"/> Public or private note has been added <input type="radio"/> Public note has been added <input type="radio"/> Private note has been added <input type="radio"/> File uploaded
<input type="checkbox"/> Any of the following fields changed	Type (type) Reproducibility (reproducibility) Severity (severity) Priority (priority)
<input type="checkbox"/> None of the following fields changed	Type (type) Reproducibility (reproducibility) Severity (severity) Priority (priority)
<input type="checkbox"/> Condition was	Default (Always)
<input checked="" type="checkbox"/> Condition was not	severity = minor
<input checked="" type="checkbox"/> Condition is now	severity = minor
<input type="checkbox"/> Condition is not now	Default (Always)

Notification type: In-system Email
This setting only has an effect if you use this trigger to send notifications.

[Save](#)

Another example: Let's say I'm looking to receive a notification whenever my form is changed to the status of 'Approved'. I would create a trigger that states:

1. Condition WAS NOT = Status is Approved
2. Condition IS NOW = Status is Approved

[Check out our guide on time based triggers here](#)

Recommended next article:
[Field Visibility](#)