



## InfoCapture: Constraints

Constraints are rules applied to form fields to define how data can be entered into them.

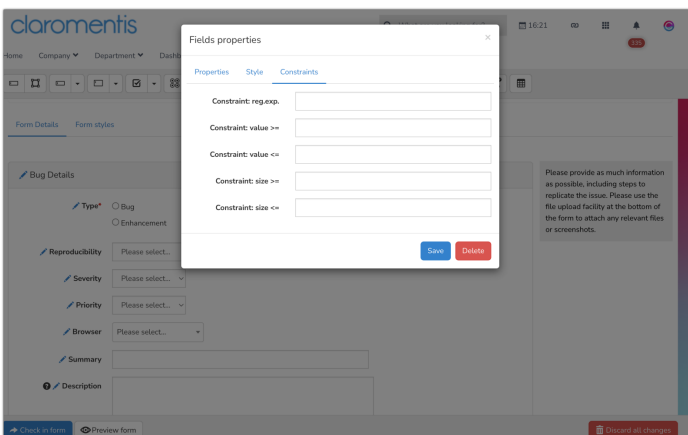
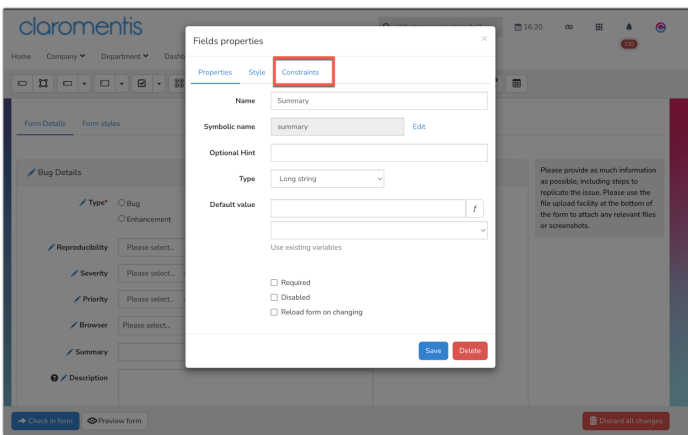
This is an optional feature utilised to suit certain form purposes but likely isn't required for every field, only a particular few.

Read on to see how constraints could apply to your forms.

### Admin > InfoCapture > Form title > Edit form fields

To add a constraint to a field, edit a field from the admin side of the form.

Click on the 'Constraints' tab to reveal the options:



Some of the below options can be coupled to create data entry ranges or used independently to create limits.

- **Reg.exp:** Stands for 'regular expression' which are character sequences that specify a pattern. When used as a constraint data entry must adhere to the rules the expression represents. (Written to suit a particular purpose)
- **Value >=:** Numerical data entry in the form field will need to be greater than or equal to this number (if used alone) or the number is the maximum entry allowed (if used in tandem)
- **Value <=:** Numerical data entry in the form field will need to be less than or equal to this number (if used alone) or the number is the minimum entry allowed (if used in tandem)
- **Size >=:** Length of data entry must be greater than or equal to this number (when used alone or in tandem)
- **Size <=:** Length of data entry must be less than or equal to this number (when used alone or in tandem)

Enter your constraint values into the correct field(s) and click 'Save' to apply this to the form version, checking this in to apply to either new tickets or all past ones.

The next submission (or edit of a past ticket if applied to these) will see fields with constraints preventing data entry that does not fit the rules applied.

## Which constraint do I need?

The two size constraints deal with the length of entries into a field that is acceptable, whether between a range or above/below, including the given number.

The two value constraints determine if entries must be higher or lower than the given number rule or if they must be between a certain range of numbers.

Regular expressions are unique as the rules they create are determined by the characters used to define the expression.

The examples below will demonstrate this functionality in further detail.

## Examples

### Size constraint: Data entry has a maximum character limit

Policy numbers at a company are always 8 characters long or less, so the form field needs to allow this but not entries over 8.

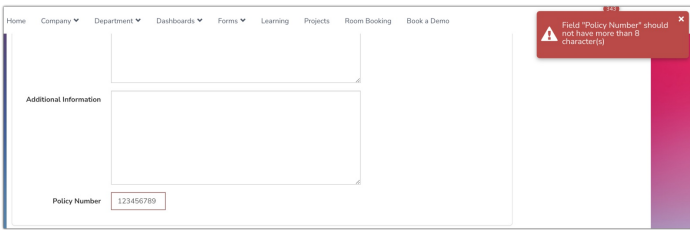
The size less than or equal to constraint can be used to set a maximum length of 8 but anything shorter than this is also allowed.

The image shows a 'Fields properties' dialog box with a close button (X) in the top right corner. It has three tabs: 'Properties', 'Style', and 'Constraints', with 'Constraints' selected. There are five rows of constraint fields, each with a label and an input box:

- Constraint: reg.exp. [ ]
- Constraint: value >= [ ]
- Constraint: value <= [ ]
- Constraint: size >= [ ]
- Constraint: size <= [ 8 ]

At the bottom right, there are two buttons: 'Save' (blue) and 'Delete' (red).

Submissions cannot be made until the rule is followed and users will be instructed on screen when data entry is not accepted.



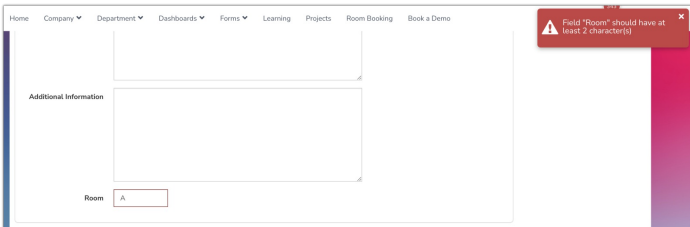
## Size constraint: Data entry has a minimum character limit

A hotel's naming convention for their rooms has a minimum of two characters.

To ensure at least two characters are entered into this form field, the size greater than or equal to constraint can be set on it as below.

Constraint	Value
Constraint: reg.exp.	
Constraint: value >=	
Constraint: value <=	
Constraint: size >=	2
Constraint: size <=	

Anything less than the minimum will be flagged to the user and submission prevented until it is rectified.



## Size constraint: Data entry has to be a number of characters within a range

### - String fields

A 'Reference code' field in a form will only ever have 4-6 characters.

To ensure this rule is applied to the field using constraints, a range of acceptable characters is applied using both size constraints to create an upper and lower limit.

Fields properties

Properties Style **Constraints**

Constraint: reg.exp.

Constraint: value >=

Constraint: value <=

Constraint: size >=

Constraint: size <=

Save Delete

When a user enters a number of characters outside the range, a tooltip will appear to inform them of the rule, as well as an on-screen error.

The screenshot shows a form with a 'Reference Code' field containing '123'. A tooltip above the field reads 'Value must be between 4 and 6 characters'. In the top right corner, a red error message box says 'Invalid field values'.

Submissions cannot be made until the rule is followed and an entry between 4 and 6 characters is given.

**- Free text fields**

A 'Description' field in a form about bug reports is a free text field that allows an open answer, however, the administrators would like to limit this to 200 characters to ensure helpful but concise answers are facilitated.

The size constraints can be set up to create an acceptable range for this as below.

Fields properties

Properties Style **Constraints**

Constraint: reg.exp.

Constraint: value >=

Constraint: value <=

Constraint: size >=

Constraint: size <=

Save Delete

The tooltip instruction and on-screen error will appear if a user makes an entry outside of the range so that the user can rectify their answer to allow submission.

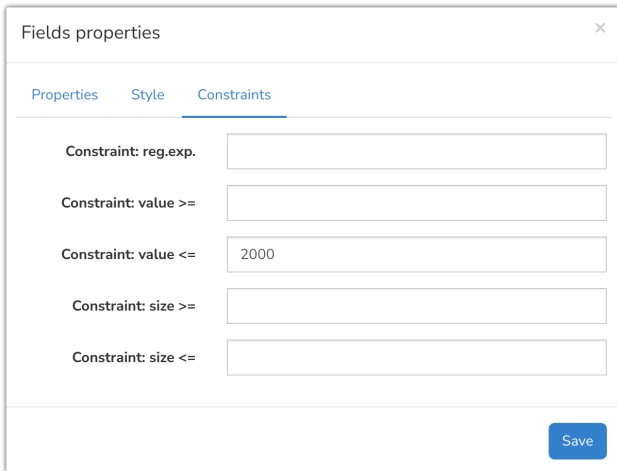
The screenshot shows a 'Description' field with a tooltip above it that reads 'Value must be between 30 and 200 characters'. The field contains the text 'Can't open link'.

## Value Constraint: Data entry has a maximum value

A 'Cost' field in a Purchasing form should never exceed £2000.

To prevent users from making submissions in the form with an amount over this, the value less than or equal to constraint can be put in place.

This defines that only numbers less than or equal to 2000 will submit successfully.

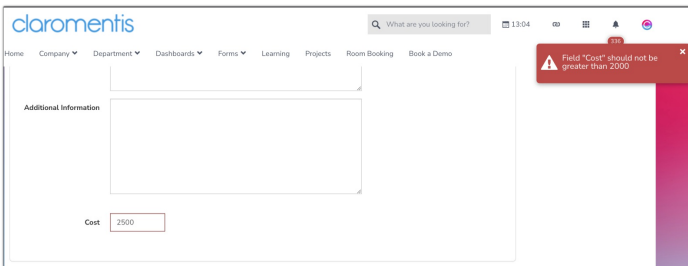


The screenshot shows a 'Fields properties' dialog box with three tabs: 'Properties', 'Style', and 'Constraints'. The 'Constraints' tab is active. It contains five rows of constraint settings, each with a label and an input field:

- Constraint: reg.exp. (empty)
- Constraint: value >= (empty)
- Constraint: value <= (2000)
- Constraint: size >= (empty)
- Constraint: size <= (empty)

A blue 'Save' button is located at the bottom right of the dialog.

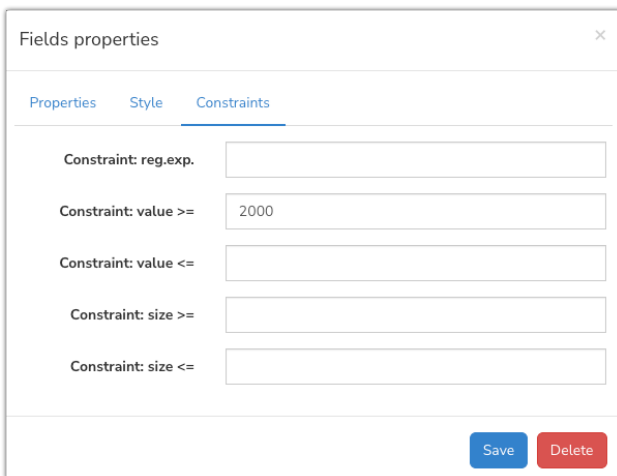
Submissions cannot be made with a value higher than the constraint allows and an on-screen error will detail this.



The screenshot shows a web form interface for Claramentis. The 'Cost' field is set to 2500. A red error message is displayed: 'Field "Cost" should not be greater than 2000'. The form includes a search bar, navigation menu, and a sidebar.

## Value Constraint: Data entry has a minimum value

Using this logic it can also be ensured that instead submissions of over or including 2000 are made by using the greater than or equal to value option...

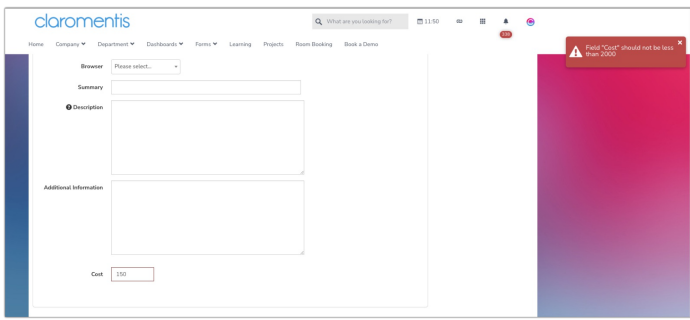


The screenshot shows a 'Fields properties' dialog box with three tabs: 'Properties', 'Style', and 'Constraints'. The 'Constraints' tab is active. It contains five rows of constraint settings, each with a label and an input field:

- Constraint: reg.exp. (empty)
- Constraint: value >= (2000)
- Constraint: value <= (empty)
- Constraint: size >= (empty)
- Constraint: size <= (empty)

At the bottom right, there are two buttons: a blue 'Save' button and a red 'Delete' button.

Submissions cannot be made with a value lower than the constraint allows and an on-screen error will detail this.



## Value Constraint: Data entry has a numerical range

It is also possible to implement a numerical range of acceptable answers by using both value constraint options e.g. between 5 and 50.

Fields properties

Properties Style Constraints

Constraint: reg.exp.

Constraint: value >=

Constraint: value <=

Constraint: size >=

Constraint: size <=

Save Delete

The tooltip instruction with the rule will appear for incorrect entries, as well as an on-screen error.

Reference Case

Value must be between 5 and 50

Number of items (5 minimum)

In this example, the administrator has included the minimum amount required in the field label to curtail incorrect data entry from the users.

The below examples use regular expressions which need to be written by someone with prior knowledge to work as intended as they fulfil rules beyond character or value restrictions. If you are unsure how to implement a constraint please [submit a support ticket](#) and we can assist you further :)

## Reg exp: Limit data entry of a field to only x characters

In a form the string field 'Employee number' should only ever contain 10 characters.

To ensure this rule is applied to the field using a constraint, the regular expression below can be used to achieve this: `(, '^[0-9]{10}$')`

Fields properties
✕

Properties
Style
Constraints

Constraint: reg.exp.

Constraint: value >=

Constraint: value <=

Constraint: size >=

Constraint: size <=

Save
Delete

Any entry that is not 10 characters exactly cannot be submitted:

The screenshot shows a web application interface with a navigation bar at the top. A red error banner at the top right says "Invalid field values". Below it, a form section titled "Additional Information" contains an "Employee Number" field with the value "E12M67". A tooltip above the field says "Value does not match expected range".

Notice that the error message for Regular Expressions is more generic than those of the size and value constraints.

This is because a regular expression is setting a rule for a broader range of parameters that Claromentis cannot create a unique error for.

If this message is not detailed enough for your use case, add a notice field to accompany the constraint to instruct users what the restriction is.

Must be 10 characters

**Employee Number**

## Reg exp: Restrict field entry to x decimal places

A 'Width' field in a form about measurements for a blade saw needs to give very precise numbers.

However, administrators need this to consistently have 4 decimal places specified to ensure standardised data to give the manufacturer.

The below regular expression can be used to facilitate this: `^d+\.\d{4}$`

Fields properties
✕

Properties
Style
Constraints

Constraint: reg.exp.

Constraint: value >=

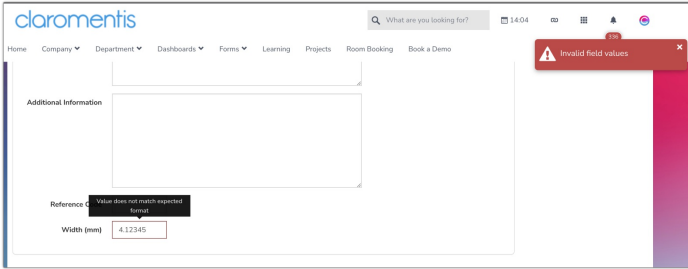
Constraint: value <=

Constraint: size >=

Constraint: size <=

Save
Delete

Any entry under or over 4 decimals will not be allowed to be submitted.

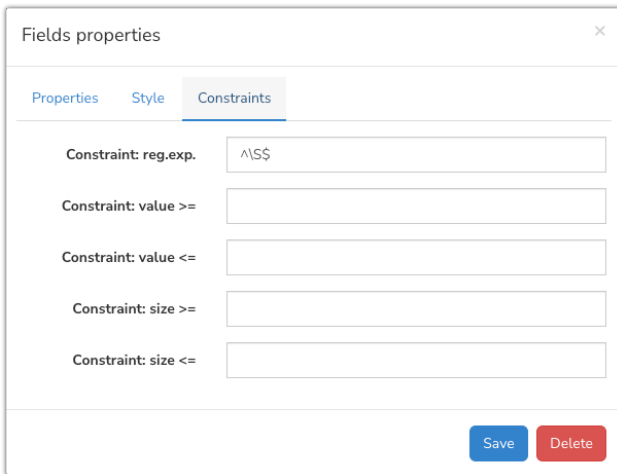


## Reg exp: Ensure data entry does not include any whitespaces

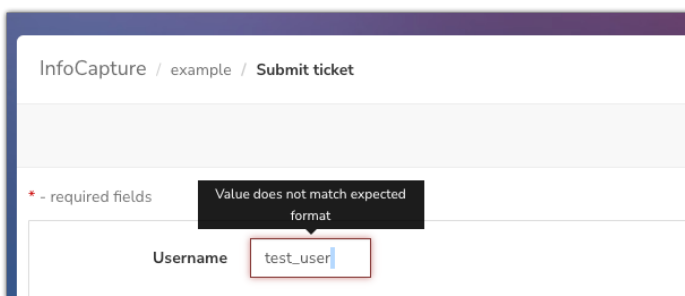
A form deals with gathering information for new user accounts.

It's of paramount importance that the username field does not contain any whitespace.

The below regular expression can be used to make this rule: `^\S$`



Any whitespace included when the ticket is attempted to be submitted will be flagged with an error and the user cannot continue until rectified.



Constraints are a great way to ensure data entry in fields is what an administrator needs from the end user.

More complex functions that cannot be achieved with constraints alone or use them as the basis for another process would likely need further investigation and custom work to support this in your form.

In these cases, please submit a [change request ticket](#) outlining your requirements and the team can assist you.



Tags: eform, Form, infocapture, constraint