

# CSS Positioning

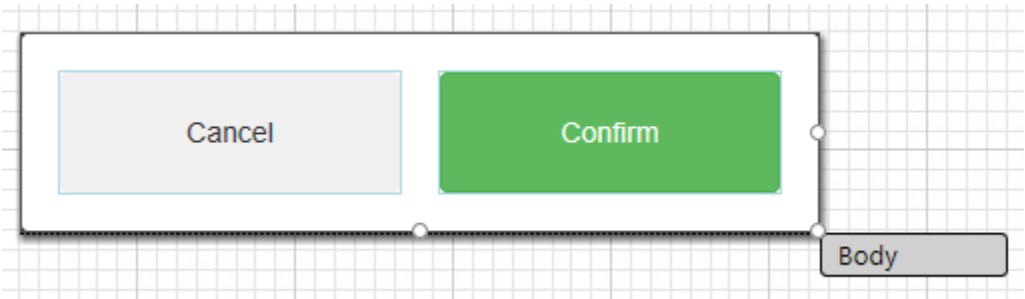
## New absolute position: css position

CSS position is a new absolute position system which replaces the old anchoring system. The new CSS position for absolute forms allows to easily build complex forms providing new features compared to the old anchoring system. Particularly it allow to adjust the size of the content depending on the browser (or dialog window) size; compared to the old anchoring system where size and location was expressed in absolute pixels, is now possible to express positioning values expressed in percentage (%) values. This means that now multiple elements on the same horizontal (or vertical) line can now adjust their size & position depending on the browser/window size; this is a great advantage compared to the old anchored layout. For example you can now design forms where, independetly from the browser size, elements are centered on the screen, or where multiple elements equally divide the available space; this means that many of the complex forms which previously could be done only using Responsive Forms can now be created as absolute forms. Note that the new CSS Position doesn't fully replace the Responsive Form; the Responsive Form has extended capabilities as the possibility to adjust element's size depending on their content and not only on the available browser size, and the possibility to adapt the actual form layout based on the browser size.

The new property, `cssPosition`, replaces `location`, `size` and `anchoring` properties. `cssPosition` uses 6 different css properties: `top`, `right`, `bottom`, `left`, `width` and `height`. The syntax for each property is exactly as css property (the value will be filled in as style property at runtime). Each property can be `-1` (default value meaning not set), an integer value in pixels , an integer value in percentages or a `calc` function ( e.g. `10px`, `50%`, `calc(50% - 10px)` see [https://www.w3schools.com/cssref/func\\_calc.asp](https://www.w3schools.com/cssref/func_calc.asp) ). This syntax allows to easily build complex forms eliminating some limitations that old anchoring system was having.

### Example:

How 2 buttons will equally divide the available width.



The 2 buttons in the image above will equally divide the available width. If the form grows in width both buttons will equally grow in width. They have 20px margin from all corners and there a 20px between the "Cancel" button and the "Confirm" button.

Layout	Layout
<div>cssPosition</div> <div>top</div> <div>right</div> <div>bottom</div> <div>left</div> <div>width</div> <div>height</div>	<div>cssPosition</div> <div>top</div> <div>right</div> <div>bottom</div> <div>left</div> <div>width</div> <div>height</div>
20,-1,20,20,calc(50% - 30px),-1	20,20,20,calc(50% + 10px),-1,-1
20	20
not set	20
20	20
20	calc(50% + 10px)
calc(50% - 30px)	not set
not set	not set

### CSS Position for the "Cancel" button.

### CSS Position for the "Confirm" button.

In the 2 images above you can see the CSSPosition of the "Cancel" button and of the "Confirm" button.

You can notice that the "Cancel" button is positioned 20px from *Top*, *Left*, *Bottom* corners; the height is not set, since both *Top* and *Bottom* are both set, the button will grow in height. The width is set to `calc(50% - 30px)`, this means will take 50% of the available width minus 30px; we calculated 20px from the left corner (=left position) and 10px from the middle of the screen, so will leave some space between the "Cancel" and the "Confirm" button.

The "Confirm" button similarly set 20px from *Top*, *Right*, *Bottom*; for the "Confirm" button the width is not set, instead the *Left* position is set to `calc(50% + 10px)`; this is equivalent to leave the *Left* position not set and set the width to `calc(50% - 30px)` as we did for the "Cancel" button.



### Note:

In case left and right are both set, width is seen as min-width of the element. The same for height, if both top and bottom are set, height will be seen as min-height.

---

## CSS position compatibility

---

CSS position is an implementation specific to NGClient, will not work on the other clients. It is based on directly outputting the values in html page styling. New solution wizard and new form wizard have been modified to make this default layout , however old anchoring system can still be used. CSS position is a form flag, so a form can either be in absolute(anchoring) position, responsive position or CSSPosition , they cannot be mixed in same form. In order to mix them you have to use containers with subforms of different layout.

CSSPosition is only used for record view forms, tableview/listview still work on same older system.

---

## CSS position conversion

---

You can convert absolute position form(s) to CSS position. We added two context menu actions for this, one on form level, one on solution level (will convert all forms in the solution). Note that this action is not undoable and there is no conversion back (from CSSPosition to anchoring).