

# Getting to know Actions in CommBox Control Editor 3.0.0

For complete documentation, see the latest user manual by clicking the information button at the top of the editor.

## Action Types

Button actions now have three types:

1. **Tap** - The button will respond to simple taps. This is the default and was the only type supported in previous versions of the editor. If you need to send a sequence of commands, you must use Tap.
2. **Tap Plus Long Press (Repeat Command)** - Use this to allow long press on a button, most commonly for volume buttons. It will keep sending the same command to the device as long as the button is held down. If tapped, it will send the command once.
2. **Tap Plus Long Press (Start and End Commands)** - Use this to allow long press on a button, most commonly for volume buttons. It will send a start command when you first press the button, and an end command when you release. If tapped, it will send the start and end commands once in sequence. This type of long press is rarely used.

## Condition Blocks

When you touch a button, it can now do something different depending on the state of the system.

In the screen shot, a single button is used to cycle through the inputs HDMI 1 to 4 when the screen is turned on.

In the highlighted condition block, it has two conditions. Both conditions must evaluate to TRUE for this condition block to execute.

Here you can see that it is testing to see that the power of the screen is on, and that the input is currently set to HDMI 1.

When this is TRUE, it performs no navigation and executes a single command to set the input to HDMI 2, then next input in the sequence.

Similarly, there are three more condition blocks that test that the power is on and what the input is set to. Each sets the input to the next in the sequence.

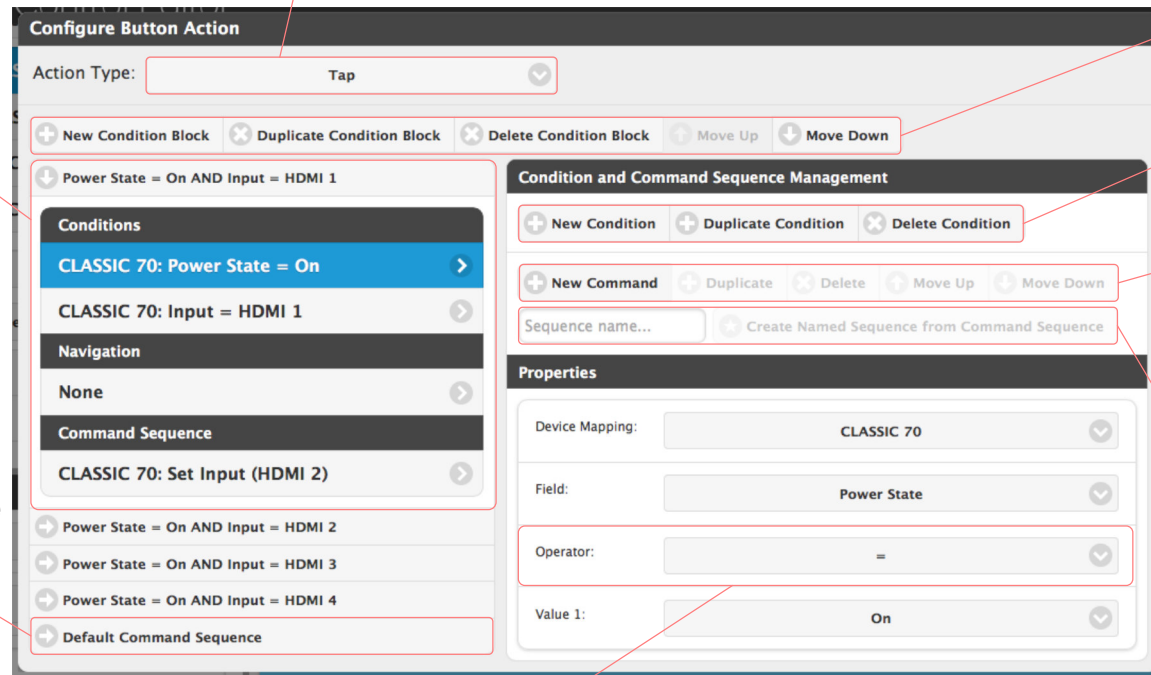
If a condition block executed, all other condition blocks are ignored. Each is tested in order, so you can arrange them using the buttons above.

## Default Command Sequence

The default command sequence is a condition block with no conditions. It will only execute if all condition blocks above evaluate to FALSE.

When you load old control designs, all of the commands you configured before will be in this section.

You can add condition blocks above, or even make a duplicate of the Default Command Sequence and add conditions to it.



## Managing Condition Blocks

This row of buttons allow you to add, duplicate, delete and move condition blocks up and down. The Default Command Sequence cannot be moved from the bottom.

## Managing Conditions

This row of buttons allow you to add, duplicate, delete conditions within a condition block. Since all conditions must evaluate to TRUE, the order does not matter.

## Managing Commands

This row of buttons allow you to add, duplicate, delete and move commands up and down within a condition block.

In addition to commands and sleeps, you can now refer to a named command sequence (see below).

## Named Command Sequences

Control Editor 3.0.0 allows you to create named command sequences to reduce duplication of your efforts.

For example, if you have a particular sequence of commands that roll down the projector screen, turn the projector on and set the input 20 seconds later, you can now create a named command sequence to do just that. If you use it in multiple places, you can now just refer to it on those button presses instead of having to program the whole sequence each time.

The highlighted text box and button on the screen shot allow you to turn the selected command sequence into a named command sequence. Just enter a name for it and press the button.

If you want to see your named command sequences and edit them, go to the controller properties and click the Named Command Sequences button.

## Operators

Above, you can see the properties for a condition. These are simple logic statements that can be evaluated to true or false.

Operators allow you to compare a status field to a value, or a range, so it is important to understand what they mean. Symbols are used because they are easier to read in condition summaries.

= (equal to) - This will evaluate to TRUE is the status field matches the Value 1 exactly.

!= (not equal to) - This will evaluate to TRUE if the status field is set to anything other than Value 1.

< (less than) - This will evaluate to TRUE if the status field is less that Value 1 (numbers only).

> (greater than) - This will evaluate to TRUE if the status field is greater that Value 1 (numbers only)

**range** (in the range) - This will evaluate to TRUE if the status field is within the range between Value 1 and Value 2 inclusive (numbers only).

**!range** (not in the range) - This will evaluate to TRUE if the status field is not within the range between Value 1 and Value 2 inclusive (numbers only).