Multiviewing inputs and controlling it via the API

By Håvard Njåstad

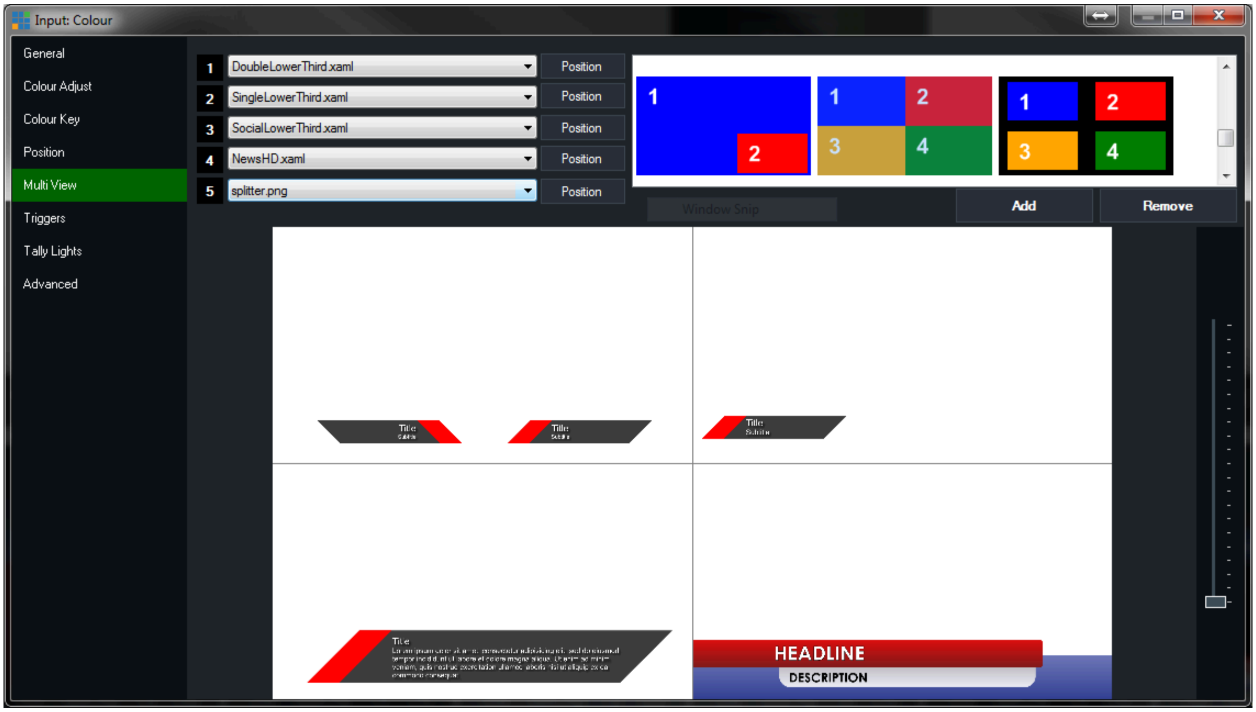
Table of contents:

Multiviewing inputs and controlling it via the API:

Downloadable components

[Images used for the MultiView]

vMix Setup Multiview only



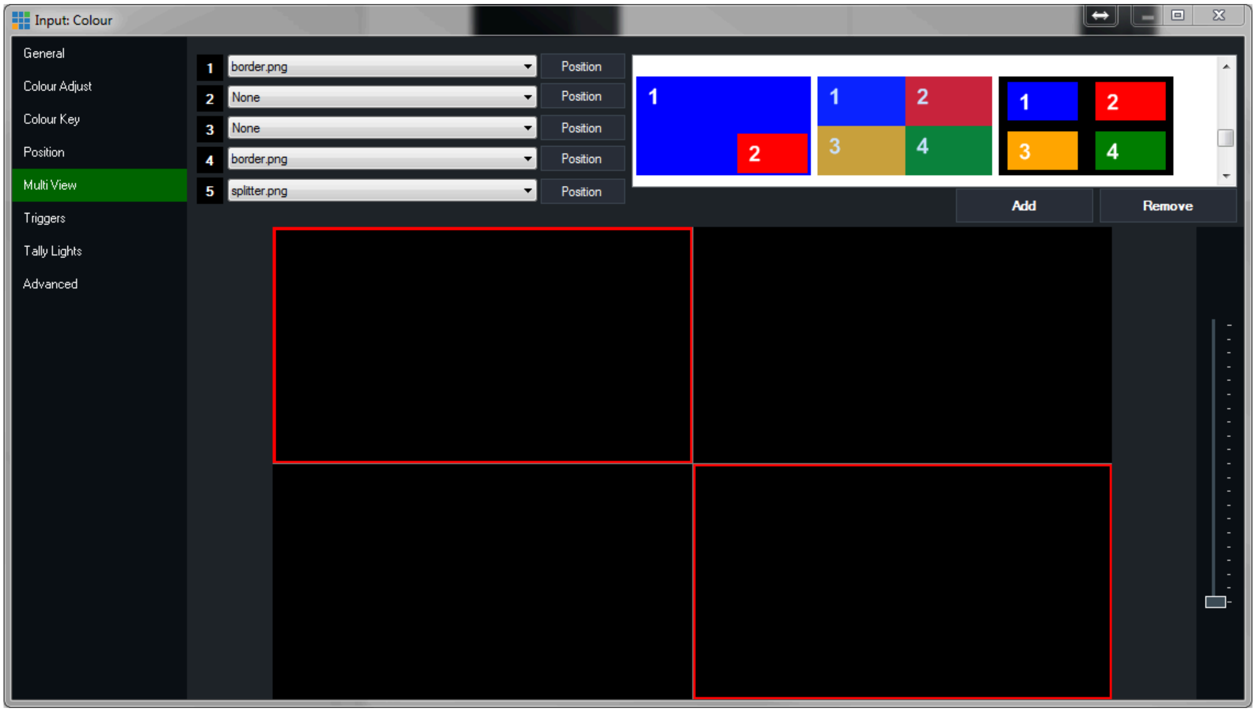
1 DoubleLowerThird.xaml
2 SingleLowerThird.xaml
3 SocialLowerThird.xaml
4 NewsHD.xaml
5 splitter.png

Onto a white “Colour” input

Note the mode that the MultiView is in, 4 inputs covering the whole input

vMix Setup with Active/Non-Active

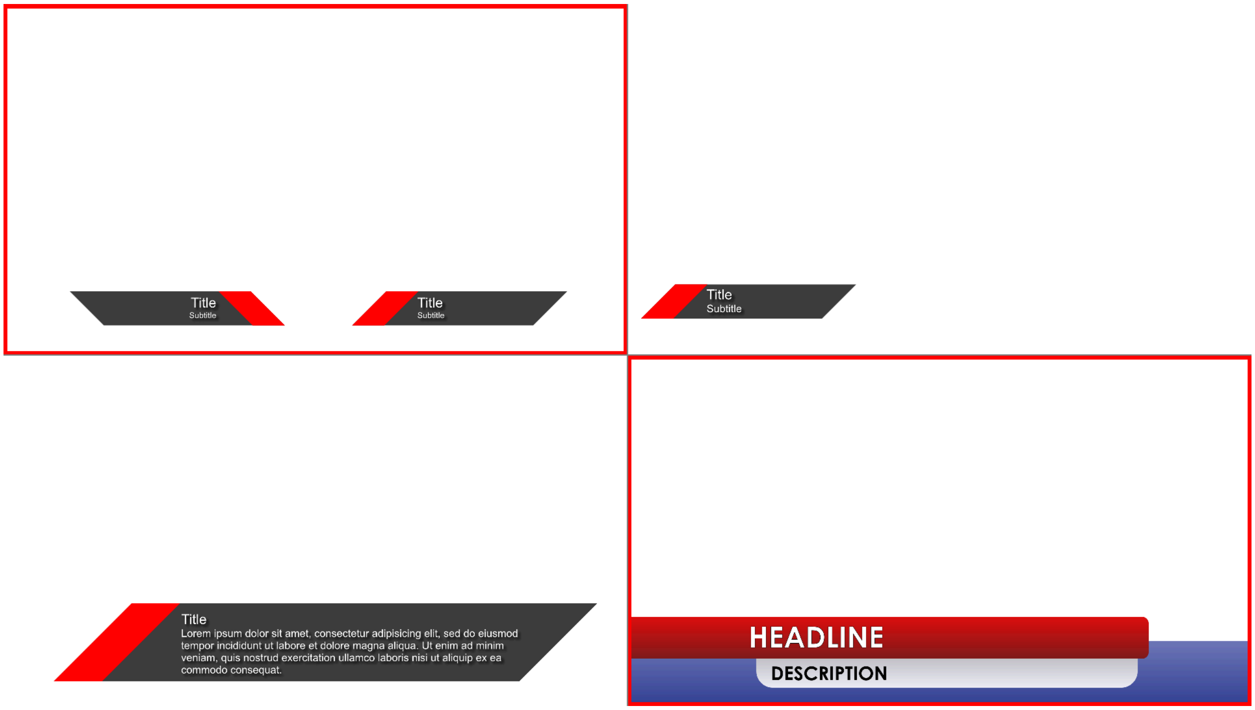
Change MultiView overlay number 5 from the “splitter.png” to a transparent “Colour” input which has these MultiView settings:



1 border.png Position
2 None Position
3 None Position
4 border.png Position
5 splitter.png Position

The “border.jpg” input is set accordingly to what input is active later on.

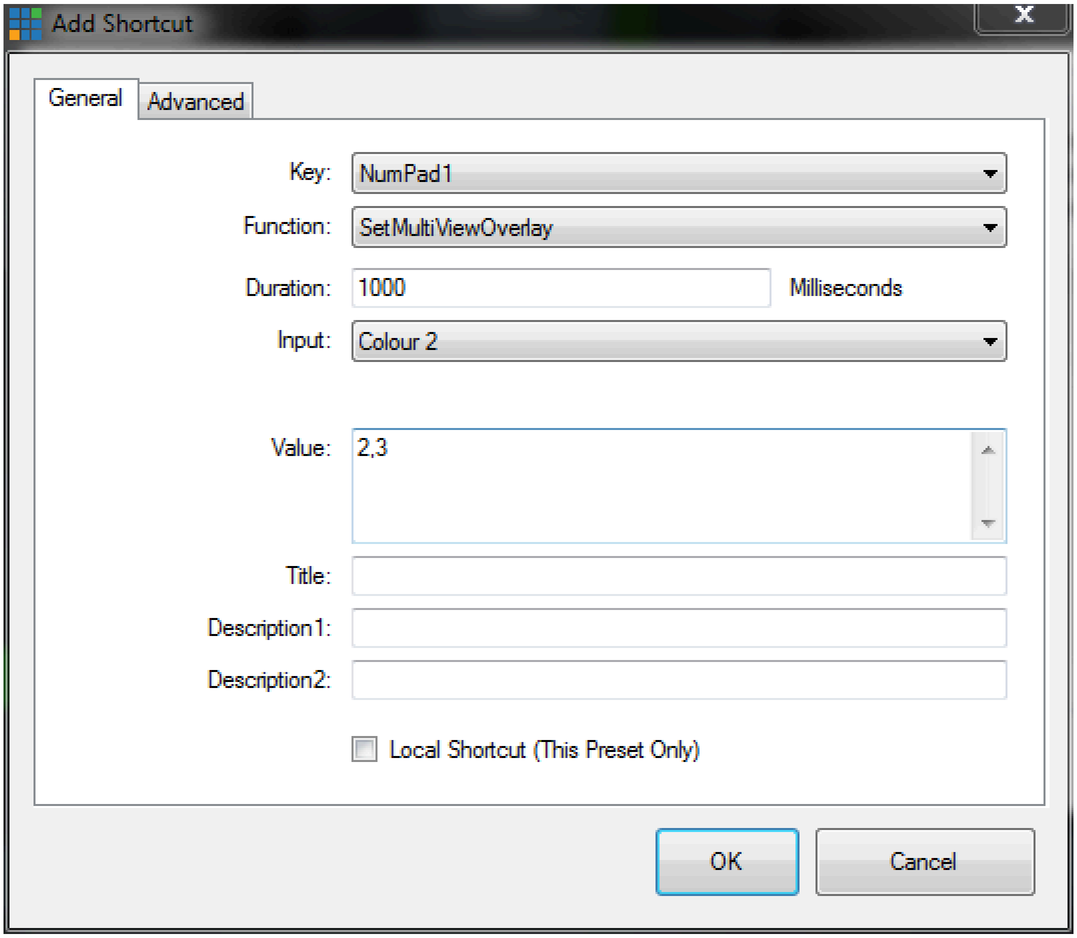
Result


Title
Subtitle
Title
Subtitle
Title
Subtitle
Title
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do elusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
HEADLINE
DESCRIPTION

Controlling the MultiView

Using shortcuts

Using shortcuts is the simplest (but not the best) way of doing so:



Key: NumPad1

Function: SetMultiViewOverlay

Duration: 1000 Milliseconds

Input: Colour 2

Value: 2,3

Title:

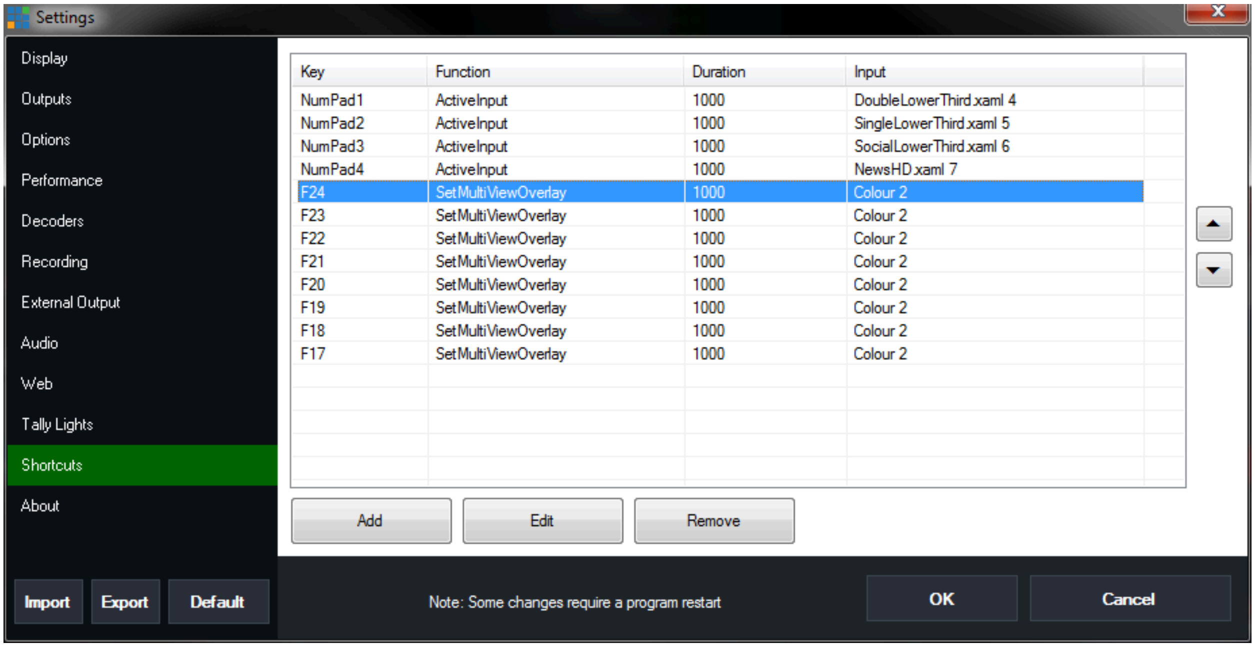
Description1:

Description2:

□ Local Shortcut (This Preset Only)

This is the basic function of using shortcuts to change a tile of the MultiView, it will change the “Colour 2” input’s second MultiView to input 3. This way you can select which inputs to monitor on the MultiView.

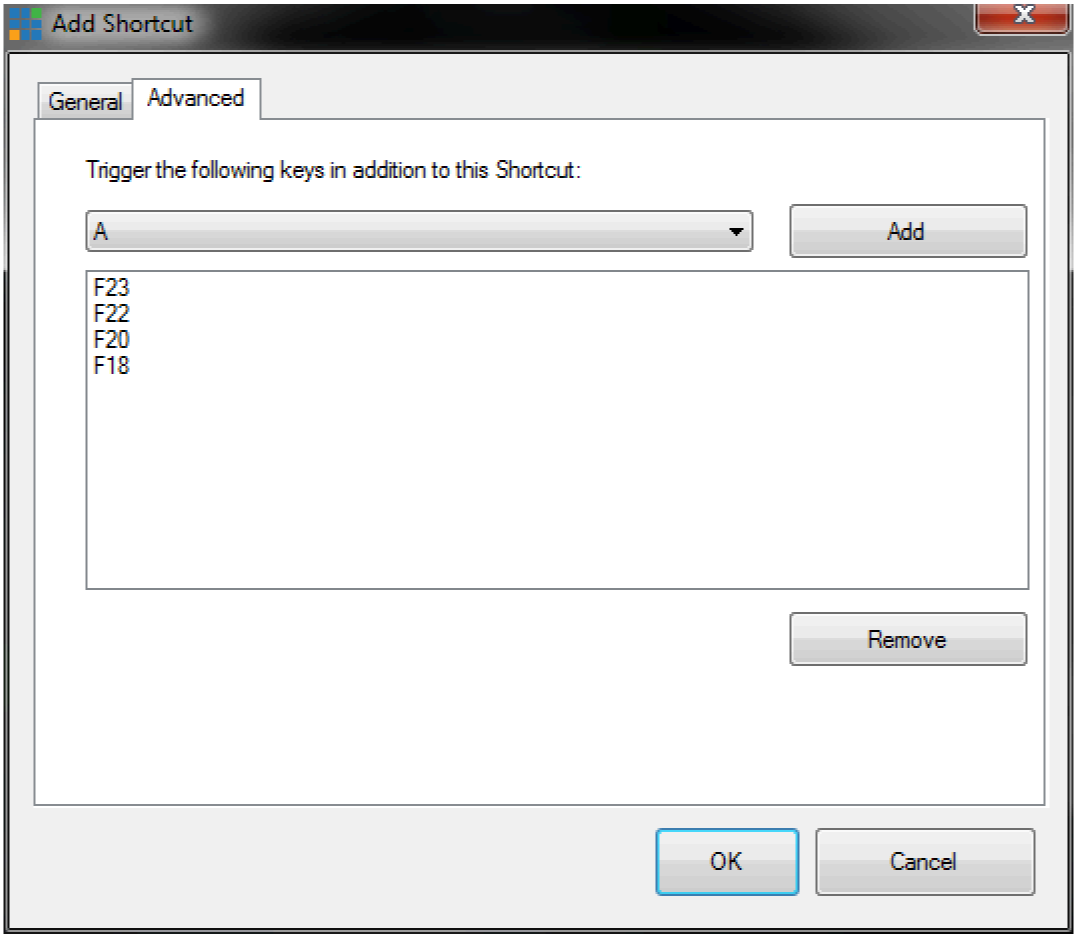
If you further advance the shortcuts you’ll be able to make the MultiView show a red border around the pressed input:



Key	Function	Duration	Input
NumPad1	ActiveInput	1000	DoubleLowerThird.xaml 4
NumPad2	ActiveInput	1000	SingleLowerThird.xaml 5
NumPad3	ActiveInput	1000	SocialLowerThird.xaml 6
NumPad4	ActiveInput	1000	NewsHD.xaml 7
F24	SetMultiViewOverlay	1000	Colour 2
F23	SetMultiViewOverlay	1000	Colour 2
F22	SetMultiViewOverlay	1000	Colour 2
F21	SetMultiViewOverlay	1000	Colour 2
F20	SetMultiViewOverlay	1000	Colour 2
F19	SetMultiViewOverlay	1000	Colour 2
F18	SetMultiViewOverlay	1000	Colour 2
F17	SetMultiViewOverlay	1000	Colour 2

Note: Some changes require a program restart

Here I have the four “ActiveInput” commands at the top, which all have the “advanced” tab active:



Trigger the following keys in addition to this Shortcut:

A

F23
F22
F20
F18

The F-key shortcuts are not meant to be pressed, but used as macros. What you see above is the ActiveInput1 shortcut which has macroed:

1. F23 - Turn on red border around input 1 (Value 1,[border input no.])
2. F22 - Turn off red border around input 2 (Value 2,-2)
3. F23 - Turn off red border around input 3 (Value 3,-2)
4. F23 - Turn off red border around input 4 (Value 4,-2)

Note that the “-2” value is the same as “none”

Repeat this for all buttons and you have yourself a MultiView that responds red borders around the active output!

This will however return wrong indication of what’s active and not when changing to an input that isn’t being used shortcuts to, and will not be able to automatically show correct active input.

Advanced programming via 3rd party software (auto-responding)

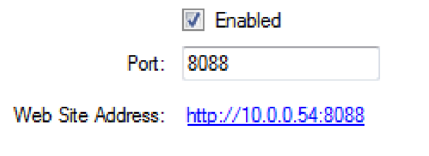
Using the vMix web API's **SetMultiViewOverlay** command, you can map a specific multiview overlay to a input, e.g.:

http://127.0.0.1:8088/api/?Function=SetMultiViewOverlay&Input=1&Value=2,3

This will change input1's second MultiView overlay to input3.

If you then are able to get your software to read the http://127.0.0.1:8088/api/, you should be able to program in your favourite language: When input is active; add red border, if not; remove. Repeat for all inputs and you should be good to go.

To reach http://127.0.0.1:8088/api/, remember that vMix must be running and "Web" must be enabled in the settings



☑ Enabled

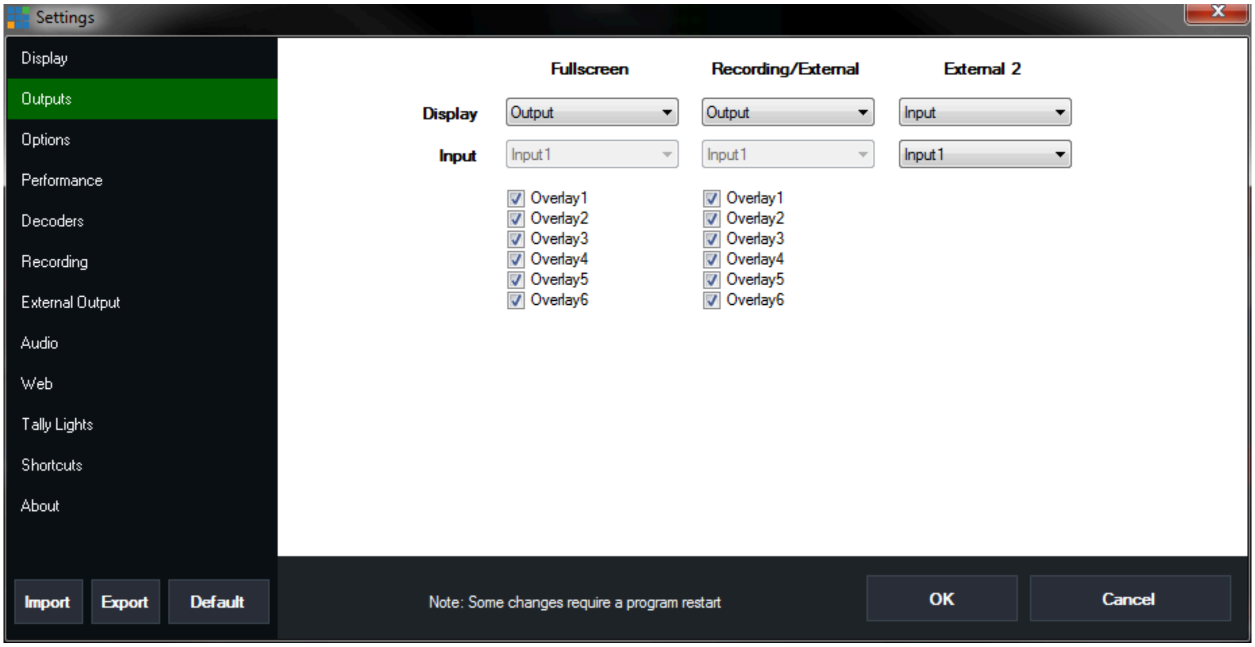
Port: 8088

Web Site Address: http://10.0.0.54:8088

Additional notes

Select Output

As many of you don't want to use this as main output to projector or big TVs when running your production, I'm recommending as for now to get a DeckLink card to do this job:



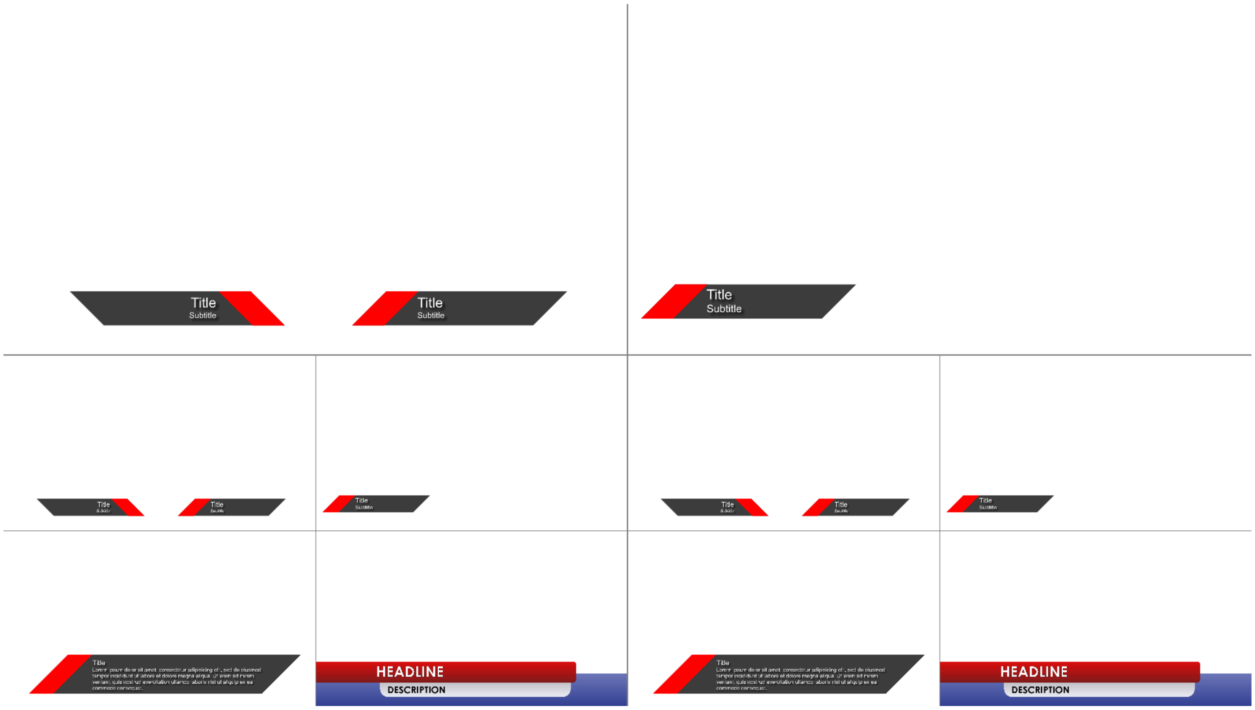
	Fullscreen	Recording/External	External 2
Display	Output	Output	Input
Input	Input1	Input1	Input1
	☑ Overlay1 ☑ Overlay2 ☑ Overlay3 ☑ Overlay4 ☑ Overlay5 ☑ Overlay6	☑ Overlay1 ☑ Overlay2 ☑ Overlay3 ☑ Overlay4 ☑ Overlay5 ☑ Overlay6	

Note: Some changes require a program restart

Extended usage of the “Fullscreen” output is coming in vMix 13, so soon these expensive output cards won’t be needed.

MultiView within MultiView

As the original vMix Multiview does, the Multiview supports pretty much as many inputs as you'd like. The way you do this is to put an 4-way MultiView inside of the MultiView (MultiViewception):



Active and Preview inputs

Again, as the original vMix MultiView supports, this kind of MultiView also supports controlling what input is active, and which one is preview.

When using the values "0" and "-1", as the input to change a specific MultiView overlay to, it will switch to Preview if the input is 0, and Active if the input number is -1.

This will, however not be a dynamic input, and if you change one of the inputs, you'll need to run the shortcut or Web API string again. So if you want a dynamic MultiView this way, you have to make a script that sends the string a couple of times a second.

Controlling Overlays

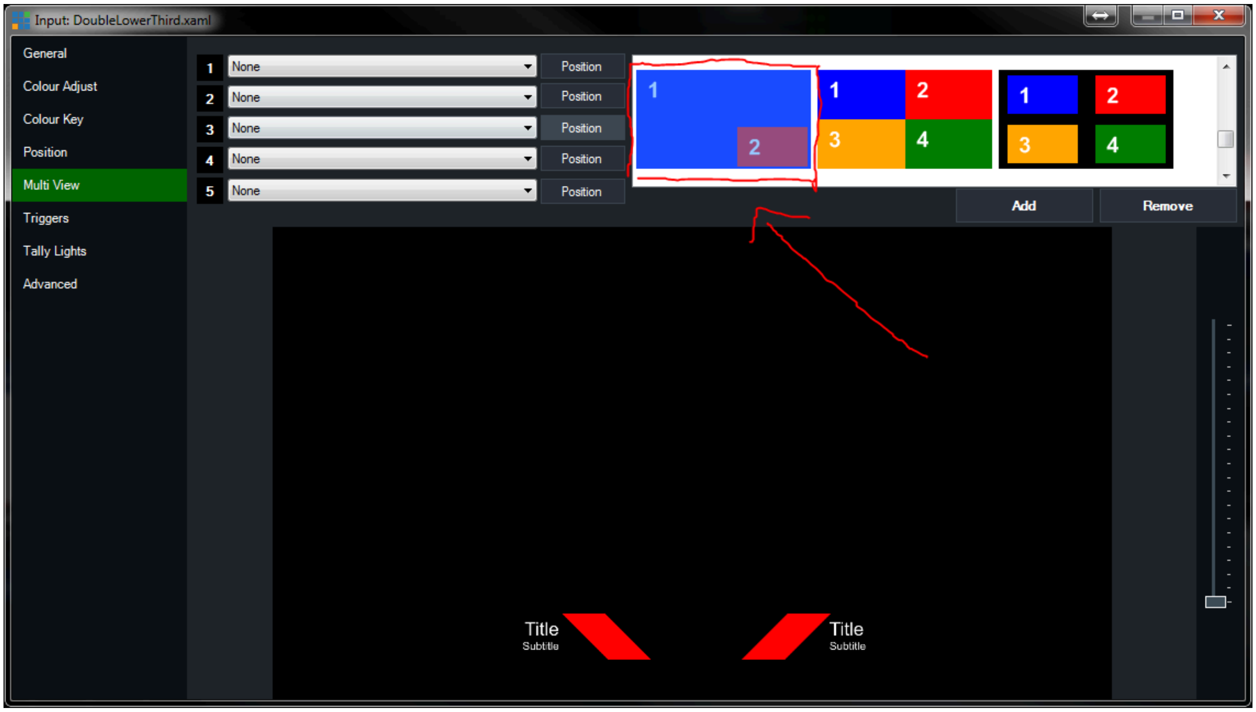
The way of checking if the inputs are active via the http://127.0.0.1:8088/api/ can also be used on the four overlay channels. This way you can easily have a MultiView for graphics and lower thirds only, which makes it really easy to set up a dedicated “Graphics Unit” in your broadcast.

Making a touch-panel vMix interface

Using the external outputs of vMix you’ll be able to read the video-stream in another application. This means if you have a C# application which can show a fullscreen DirectShow stream (vMix External Output), and add a layer of click-control above it, and splitting the fields in their respective sizes and give them commands, you’ll be able to make yourself a touch-interface for any inputs you’d like.

MultiView for Picture in Picture Control

By selecting this specific MultiView mode



Input: DoubleLowerThird.xaml

1 None Position
2 None Position
3 None Position
4 None Position
5 None Position

and binding the right shortcuts, (values 1,[background] and 2,[PiP Input]) you get a pretty neat PiP control where you can quickly change both back- and foreground images to wished inputs.