



Reactor

SKAARHOJ

The screenshot shows the Reactor web interface in a browser window. The address bar shows '192.168.11.8'. The navigation menu includes 'Home', 'Configuration', 'Simulator', 'Packages', 'Settings', and 'Device: rackfusionlive'. The main content area is titled 'Panels' and shows a configuration for 'Rack Fusion Live' (Panel ID: 2). It features a 'Camera Selector' with a grid of camera options: COMBO Studio Camera 4K Pro Marshall CV730, EOS-C300, AW-UE70, Sony ILME-FR7, CR-N500, Lumens VC-A50P, BirdDog P400, and an 'Add' button. Other sections include 'ATEM Inputs' (44 entries), 'Tally Forwarding' (Kaspers ATEM Mini), and 'Routing Trigger' (KUMO 1616). On the right, a list of connected devices is shown: Panasonic PTZ AW-UE70, BMD ATEM Kaspers ATEM Mini, Tyler's LA ATEM, Arri Cameras Arri Amira (with a 'Missing required configuration' warning), and JVC RCP GY-HM660. The footer of the interface shows 'v1.0.7-pre1 Copyright © 2023 SKAARHOJ' and 'Support' links.

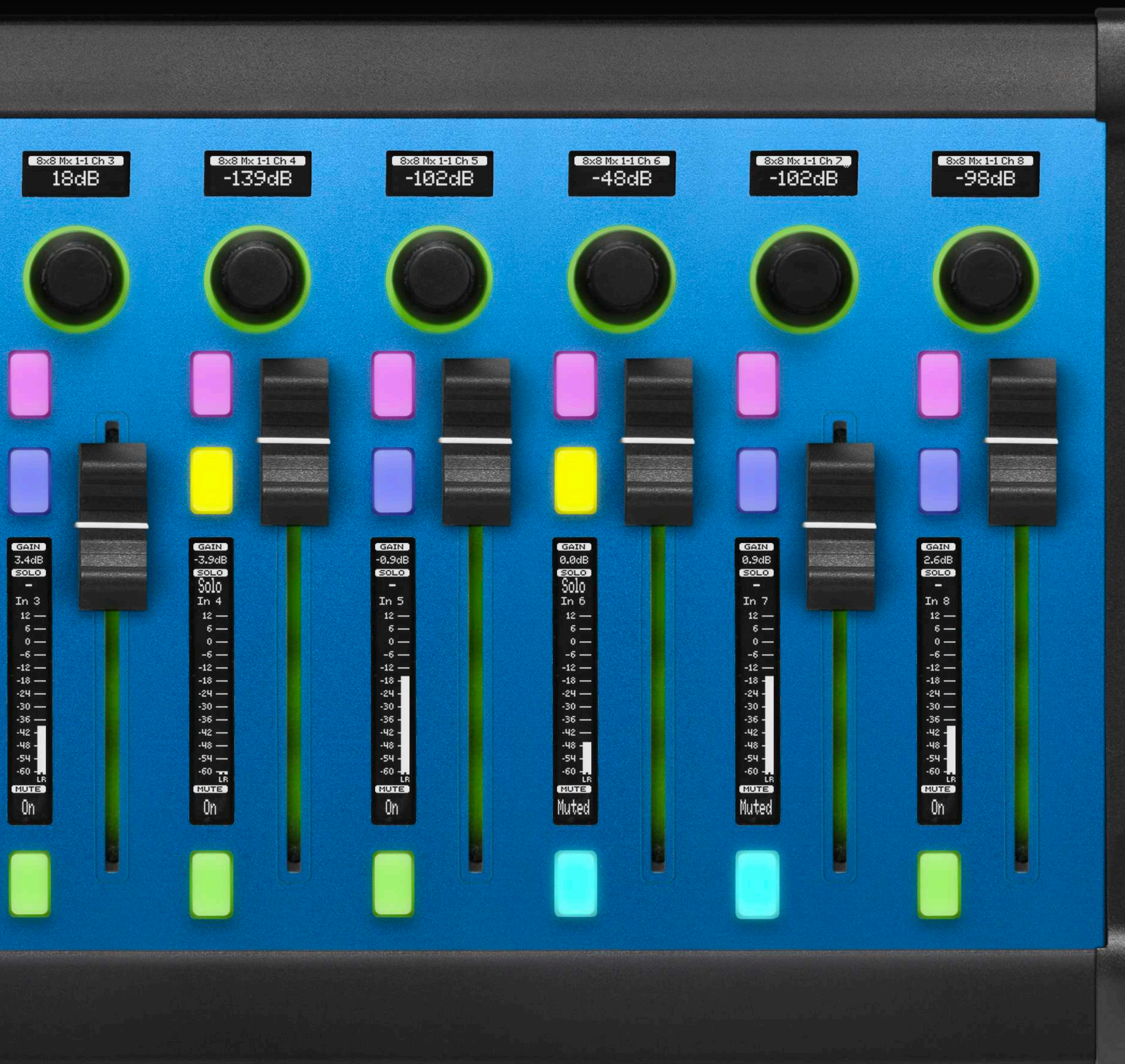
Your panels

Your configurations

Your devices

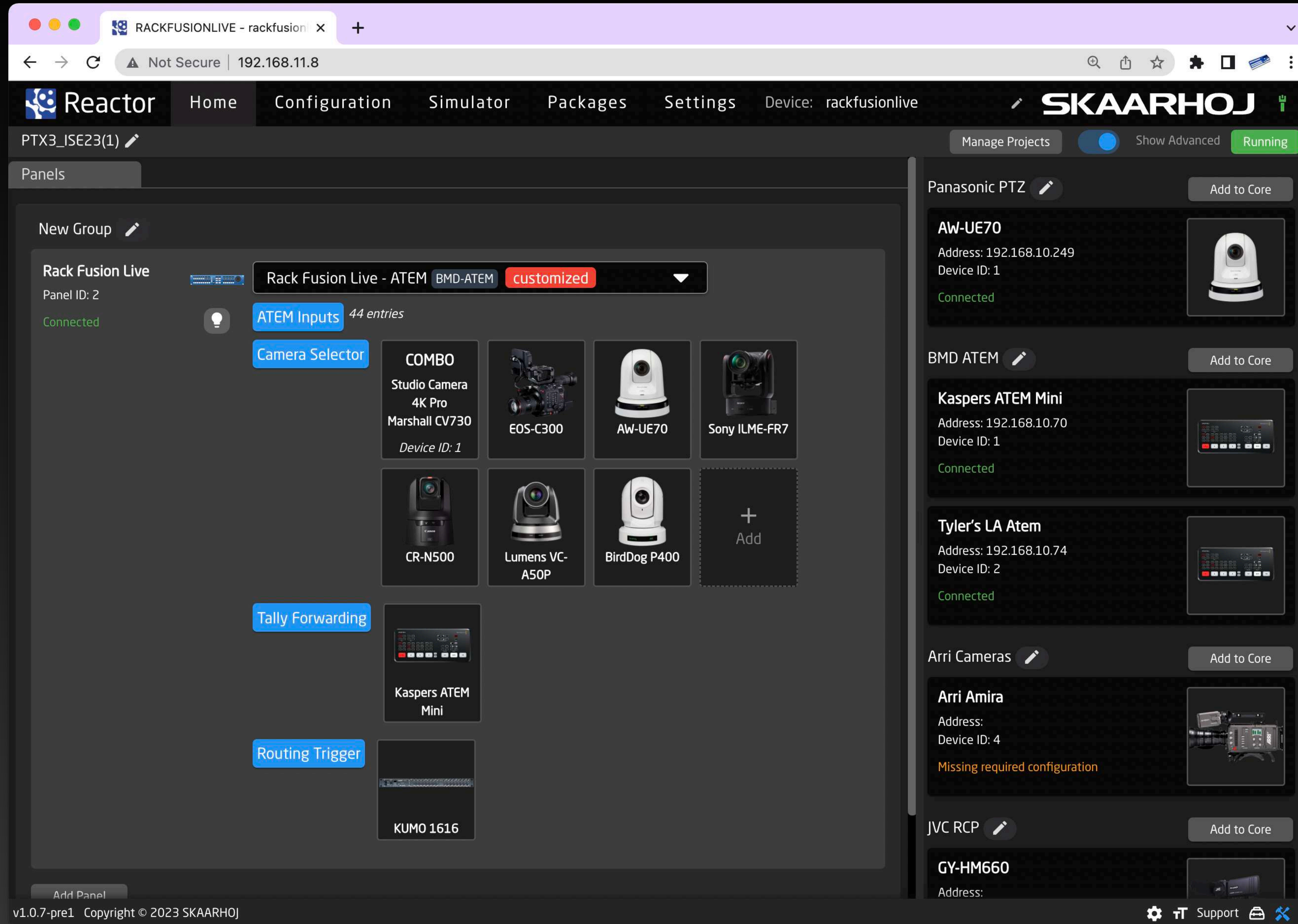
Unleashing Limitless Broadcast Control





Reactor, SKAARHOJ's comprehensive broadcast control and automation system, streamlines production workflows, control panels, and audio-visual devices for varying scales of operations. Its web application interface facilitates the creation and management of multiple projects with customizable panels, configurations, and devices. Reactor's modularity ensures seamless integration with numerous broadcast and AV devices, while its customization options allow users to assign behaviors to hardware components for adaptable, unique control systems.

- ★ Centralized management: Panels and devices in one place.
- ★ Modularity: Integrate panels seamlessly.
- ★ Section View: Visualize configurations easily.
- ★ Event handlers: Define hardware actions.
- ★ Virtual Triggers: Advanced system automations.
- ★ Generators: Auto-create layers and behaviors.
- ★ Scripting Engine: Custom automation with JavaScript.
- ★ Simulator: Virtual panel in the web browser.



Panels and Configurations

Assign a default configuration or create your own for any Raw Panel-compliant device.

Device Association

Effortlessly add devices like cameras, video switchers, routers, and audio processors depending on the configuration.

Additional Panels

Include more panels as part of existing configurations or manage them individually.

Home Screen

Reactor's Home Screen is the core of your production workflow. Effortlessly organize control panels, configurations, and AV devices, access high-level configurations through mapping tables or constant sets, and manage IP addresses and settings. Ideal for small or large-scale productions, the Home Screen serves as the central hub for integrating all components of your production environment.

Projects

Create numerous projects containing panel setups, configurations, and devices.

Devices and Device Cores

Device cores are applications connecting to specific device types. Add devices to projects from the Home Screen.

Settings

Reactor functions like a native app, allowing font size and setting changes, support team access, and advanced debugging tools.

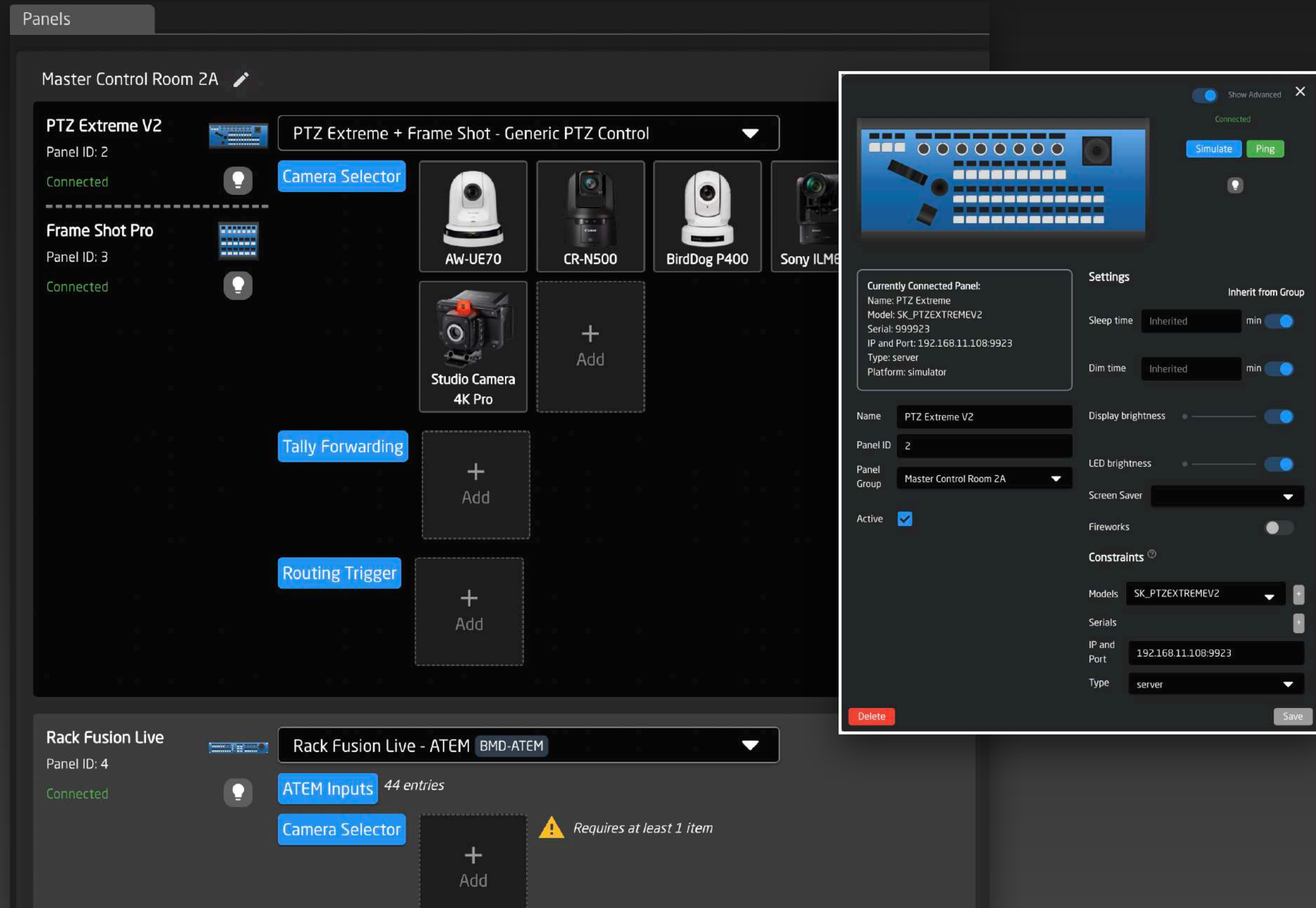
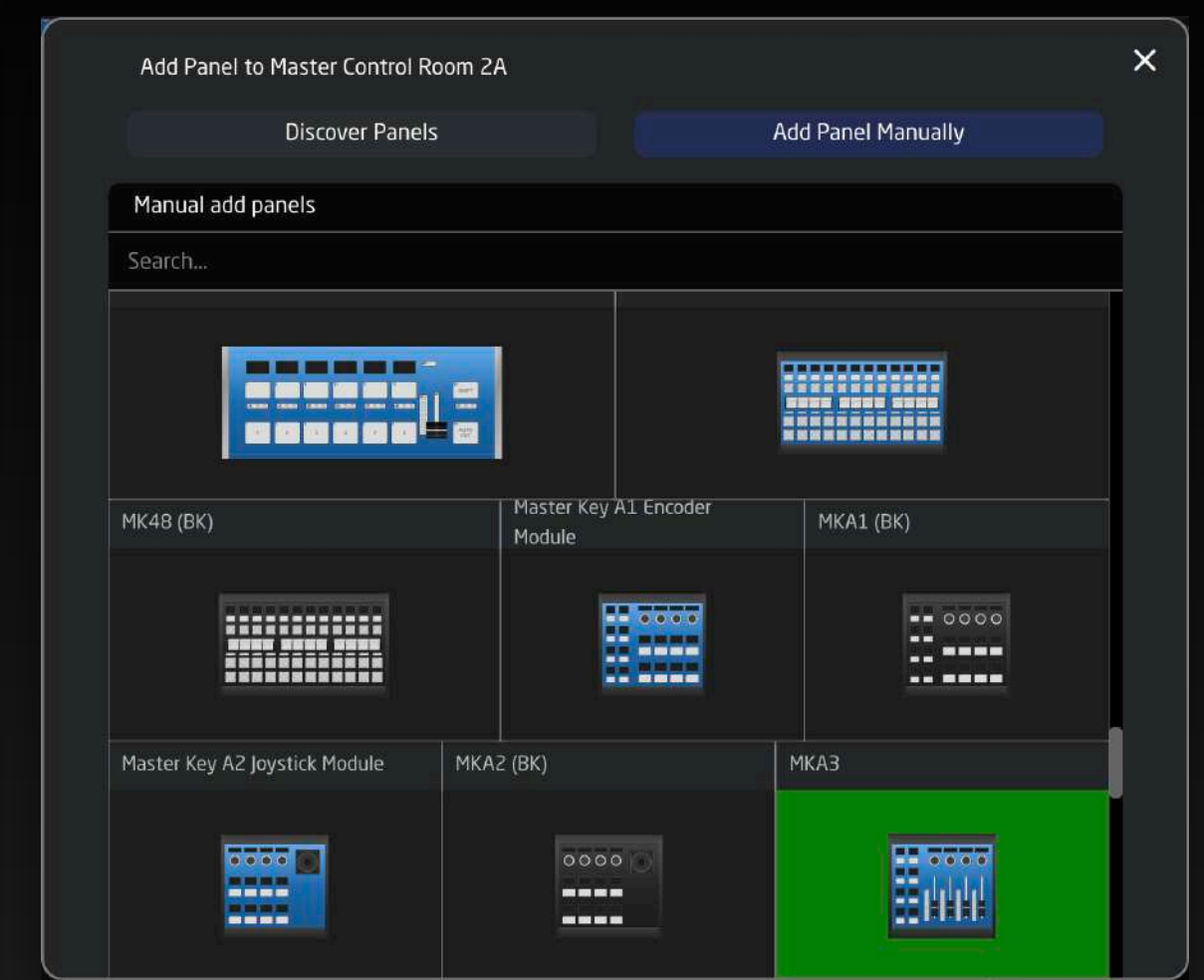
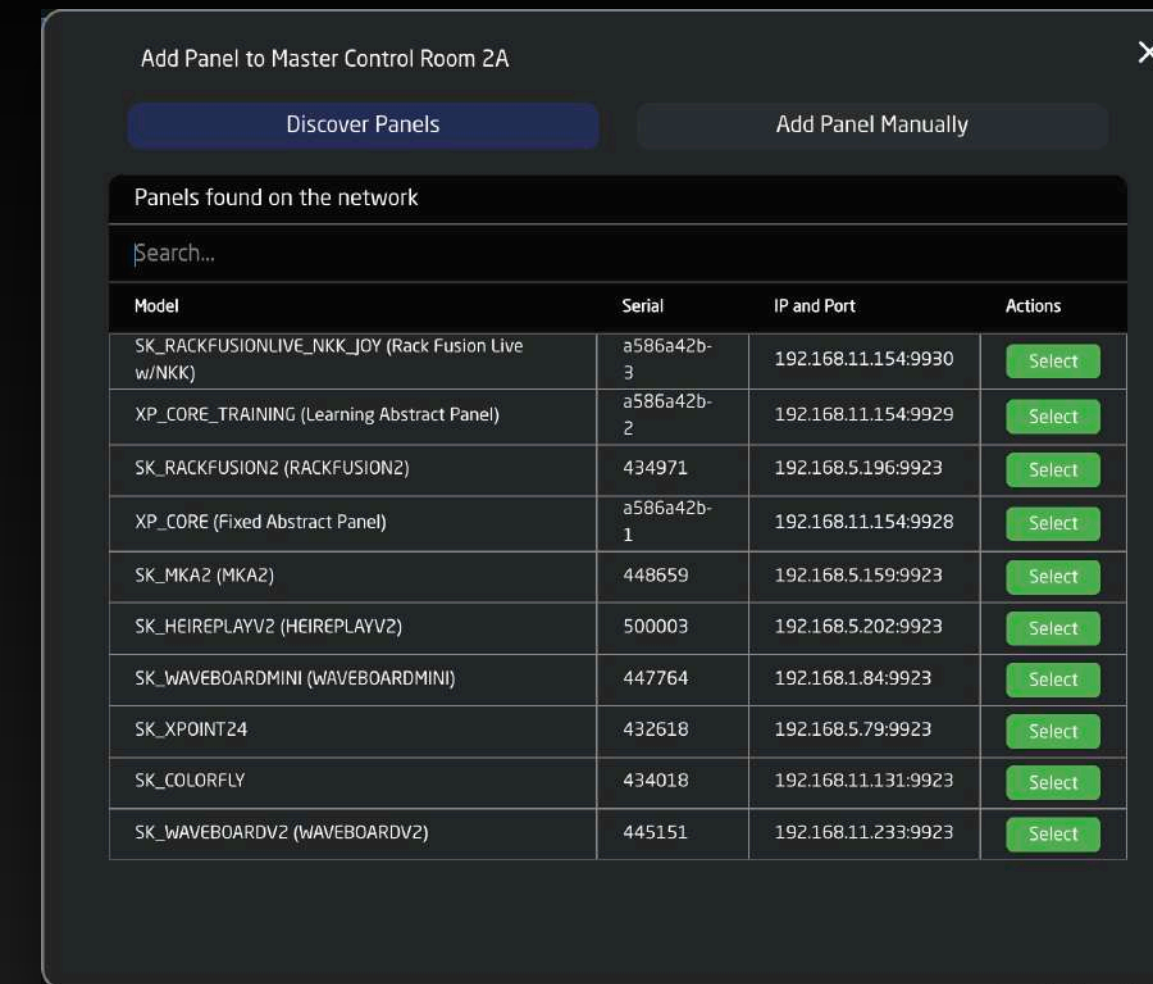
Panels

Modularity Reimagined

SKAARHOJ's modularity allows for seamless panel integration in Reactor's Home Screen. Panels like PTZ Extreme and Frame Shot Pro can be managed as a single unit with a shared configuration. The Rack Fusion Live panel is the host and runs the Reactor instance, while guest panels connect via the Raw Panel protocol, creating a versatile and powerful system.

Panels Settings

Reactor panels have customizable settings, such as sleep time, brightness, IP address, and model constraints, offering complete control over panel functionality.

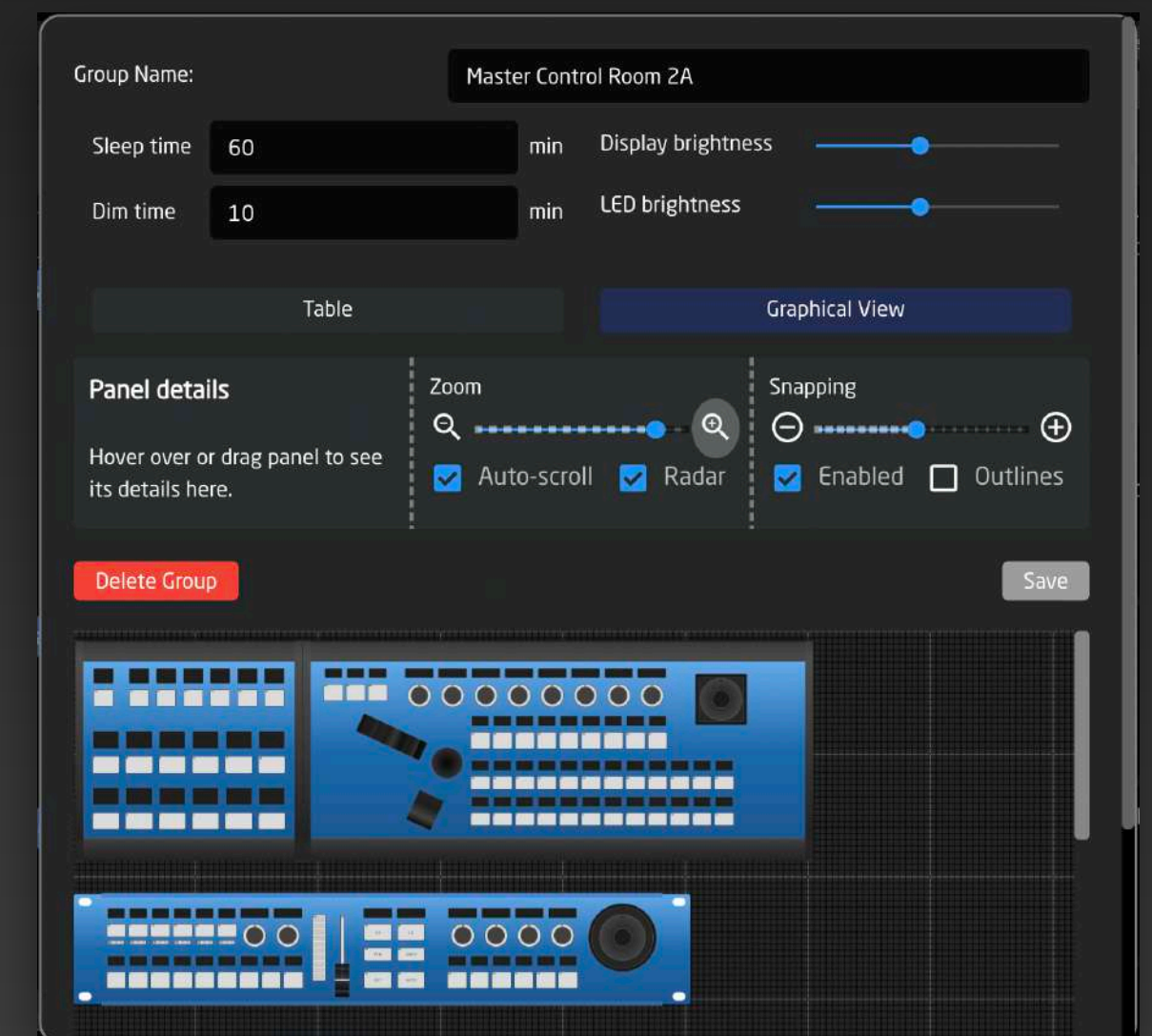


Adding Panels Made Easy

Adding panels to Reactor is easy, as it automatically discovers Raw-Panel compliant devices on the same network. Reactor's panel management system offers flexibility and seamless integration of new or unknown panels.

Panel Database

Reactor's panel database enables adding offline panels.



Panel Groups


Panels are organized into groups, simplifying alignment, and collective settings management, like sleep time and brightness. Groups represent modular panels, streamlining control for users.

Devices

Reactor connects to various Broadcast and AV devices, mapping functions to Raw-Panel compliant control surfaces for versatile integration.

Devices and Device Cores

Devices feature configuration details like name, model, IP address, and device ID, while Device Cores, as software components, provide an overview of models and parameters. They can also run remotely on Blue Pill-enabled units.



EOS-C300
Canon EOS C300 Mark III

✕

Active

Name: EOS-C300

Device ID: 2

Model ID: EOS-C300

Description:

IP: 192.168.10.229
The IP address of the Canon device

Username: admin
The Username for the Canon device

Device Password: 👁
The password that is set on the device

Connected

Development status: released

Ping Core Logs

Delete
Save


Devices

BMD Cam Control ✎
Add to Core

Studio Camera 4K Pro

Address: 192.168.10.96
Device ID: 1

Unconnected




Visca PTZ ✎
Add to Core

Marshall CV730

Address: 192.168.10.10
Device ID: 1


Unconnected



Sony ILME-FR7

Address:
Device ID: 3

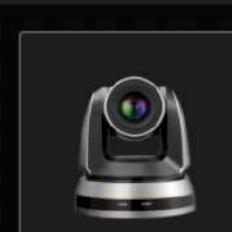
Missing IP



Lumens VC-A50P

Address:
Device ID: 7


Missing IP



BirdDog P400

Address:
Device ID: 8

Missing IP




Canon XC ✎
Add to Core

EOS-C300

Address: 192.168.10.229
Device ID: 2


Connected



CR-N500

Address:
Device ID: 5

Missing IP



AJA KUMO
Development Status: released

Address: local Simulate

Description:

Save Current State for Simulation
Parameter List
Autofix potential problems

Core specific settings:

PersistentStorage Save presets between reboots, if disabled it will clear all saved presets!

Delete
Save

Adding Devices Made Easy

Adding devices is easy, as many can be discovered on the network and added with a single click.

Select Device To Add ✕

Discover Devices
Add Manually
From Device Collection

Advanced filters Create combo devices

Devices found on the network Searching for devices...

Search...

Device Name	Device Core	Description	IP	Actions
EOS-C300	core-canon-xc-4	Canon EOS C300 Mark III	192.168.10.229	Select
ATEM 2 M/E Production Studio 4K	core-bmd-atem-13	ATEM 2M/E Production Studio 4K	192.168.10.240	Select
ATEM Television Studio HD	core-bmd-atem-8	ATEM Television Studio HD	192.168.10.58	Select
Kaspers ATEM Mini	core-bmd-atem-2	ATEM Mini	192.168.10.70	Select
ATEM Mini (Kenneth)	core-bmd-atem-2	ATEM Mini	192.168.10.72	Select
Tyler's LA Atem	core-bmd-atem-2	ATEM Mini	192.168.10.74	Select
PP	core-protocol-midi-1	A generic midi model with 16 channels	192.168.11.160	Select
Christoffers ATEM Mini Pro	core-bmd-atem-3	ATEM Mini Pro	192.168.5.80	Select

Reusing Devices in Configurations

Devices in Reactor's collection can be reused in multiple configurations.

Select Device To Add ✕

Discover Devices
Add Manually
From Device Collection


Advanced filters Create combo devices

Search:

CR-N500

Address: Missing IP
Device ID: 5

Missing IP



Select

core-panasonic-ptz @ local

AW-UE70

Address: 192.168.10.249
Device ID: 1

Connected


Select

core-bmd-atem @ local

Mapping Inputs and Cameras to Buttons

Mapping inputs and cameras to buttons uses a tabular view or mapping table, covering all essential settings for device-panel integration, including names, device numbers, configurations, tally indexes, and button colors.

Camera Selector

Description: This sets up the cameras using Standard Class configurations. [Learn more on the wiki](#) ✕

Order	Mute	Binding	Device Number:	Camera Name:	Device Config: ?	Tally Forward Config: ?
⋮	<input type="checkbox"/>	COMBO Studio Camera 4K P	1 ▼	Studio	SKAARHOJ.Devices.BMD-CamControl+VISCA.StdClass.Basic ▼	SKAARHOJ.Devices.BMD-CamC
⋮	<input type="checkbox"/>	EOS-C300	2 ▼	EOS-C:	SKAARHOJ.Devices.Canon-XC.StdClass.Basic ▼	
⋮	<input type="checkbox"/>	AW-UE70	1 ▼	AW-UE	SKAARHOJ.Devices.PanasonicPTZ.StdClass.Basic ▼	SKAARHOJ.Devices.PanasonicI
⋮	<input type="checkbox"/>	Sony ILME-FR7	3 ▼	Sony IL	SKAARHOJ.Devices.VISCA-Allstars.StdClass.Basic ▼	SKAARHOJ.Devices.VISCA-Son
⋮	<input type="checkbox"/>	CR-N500	5 ▼	CR-N5t	SKAARHOJ.Devices.Canon-XC.StdClass.Basic ▼	SKAARHOJ.Devices.Canon-XC.
⋮	<input type="checkbox"/>	Lumens VC-A50P	7 ▼	Lumen	SKAARHOJ.Devices.VISCA-Allstars.StdClass.Basic ▼	SKAARHOJ.Devices.VISCA-Son



Air Fly Pro w/NKK

Projects

Projects organize panels, devices, and configurations, allowing swift switching to a new operational context. Blue Pill-enabled SKAARHOJ panels can host unlimited projects.

Advanced users benefit from collections grouping panels and devices. Multiple projects can reference the same collection, enabling reuse of device or panel sets. Reactor's efficient management showcases its power and versatility.

Show Advanced
✕

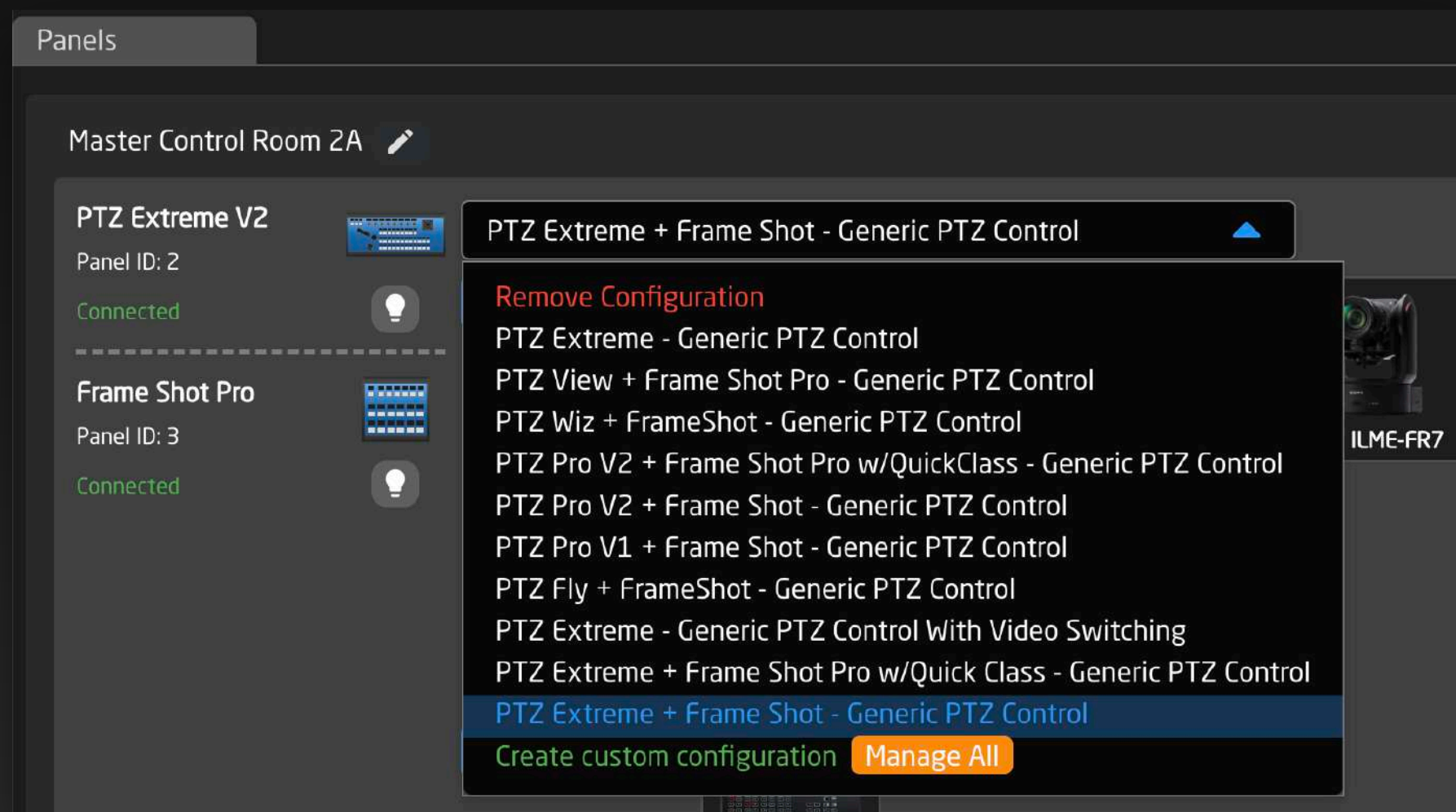
Search...

Project Title	Description	Panel Collection	Device Collection	Configuration	Status	Duplicate	Actions
MashUp		MashUp/latest	MashUp/latest	MashUp/latest	⌂ Activate	📄	🗑️
PTX3_ISE23(1)		PTX3ISE23/latest ✎	PTX3ISE23/latest ↕ ✎	PTX3ISE23/latest ✎	⌂ Activate	📄	🗑️
PTZ with Side Car		PTZwithSideCar/latest	PTX3ISE23/latest ↕ ✎	PTZwithSideCar/latest	⌂ Active	📄	🗑️
Playing With Modularity		PlayingWithModularity/latest	PlayingWithModularity/latest	PlayingWithModularity/latest	⌂ Activate	📄	🗑️
Video		Video/latest	Video/latest	Video/latest	⌂ Activate	📄	🗑️
VirtualTriggers		VirtualTriggers/latest	VirtualTriggers/latest	VirtualTriggers/latest	⌂ Activate	📄	🗑️
rackfusionlive	Congratulations with your new rackfusionlive! It's going to be amazing from here...	default/latest	default/latest	default/latest	⌂ Activate	📄	🗑️
	This collections is not used in any of your projects		PTZwithSideCar			📄	🗑️

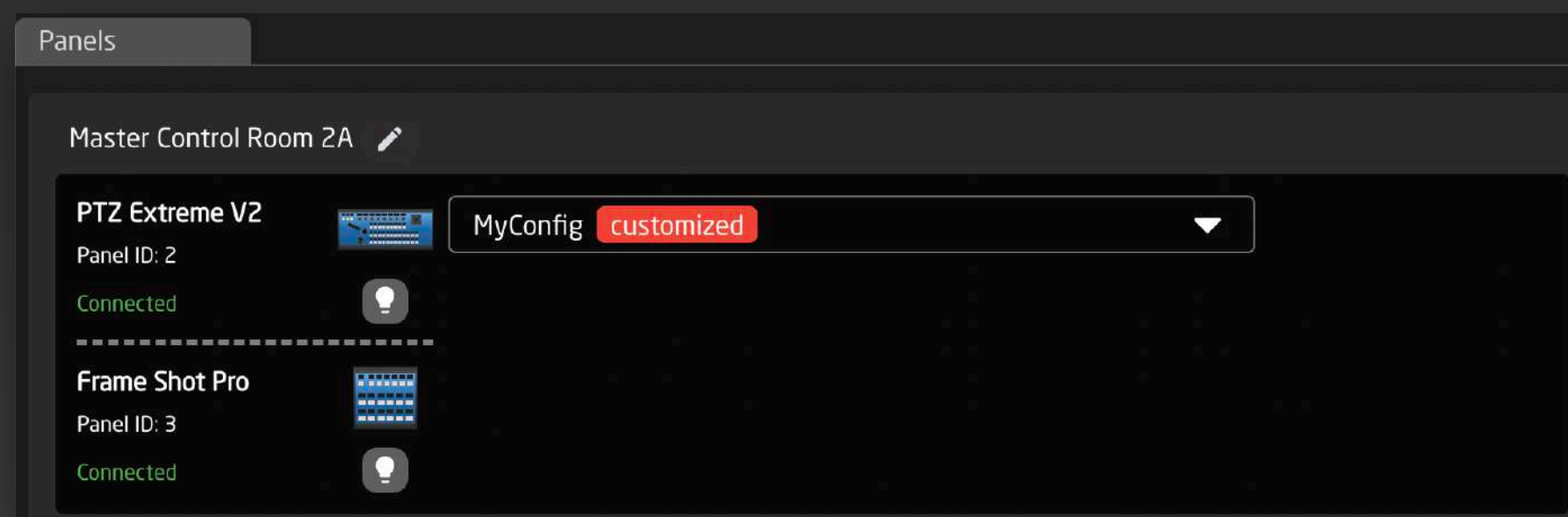
Create new project
Import/Export

Configurations

Configurations enable panel hardware components to communicate with devices, dictating responses to button presses or joystick movements. Reactor provides default configurations for panels, compatible with popular broadcast and AV devices, and allows configurations to span multiple panel types for cohesive control surfaces. This modularity supports creating large control surfaces from smaller panels.



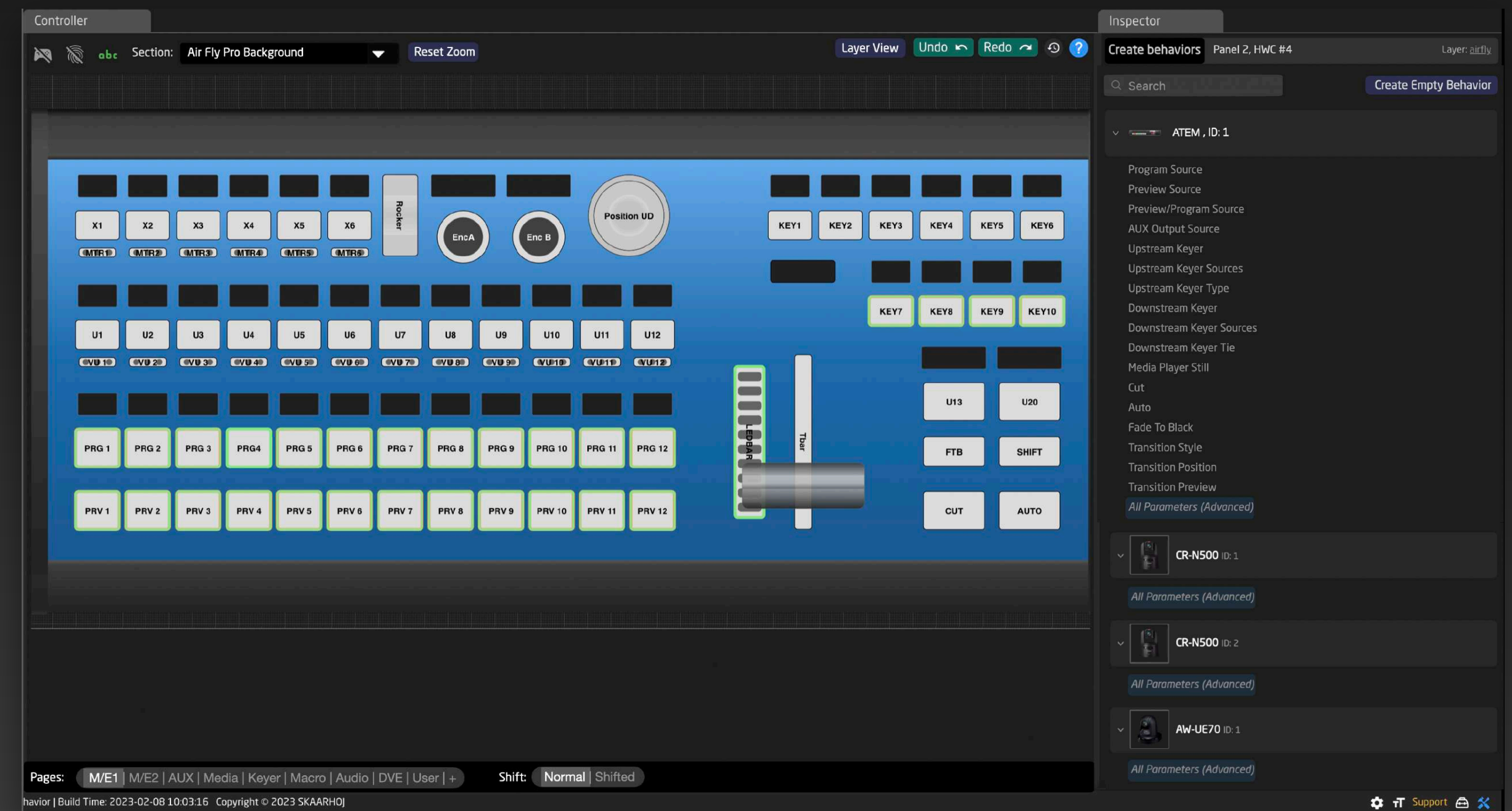
While default configurations enable immediate use, many SKAARHOJ users opt to create custom configurations from scratch.



Customize!

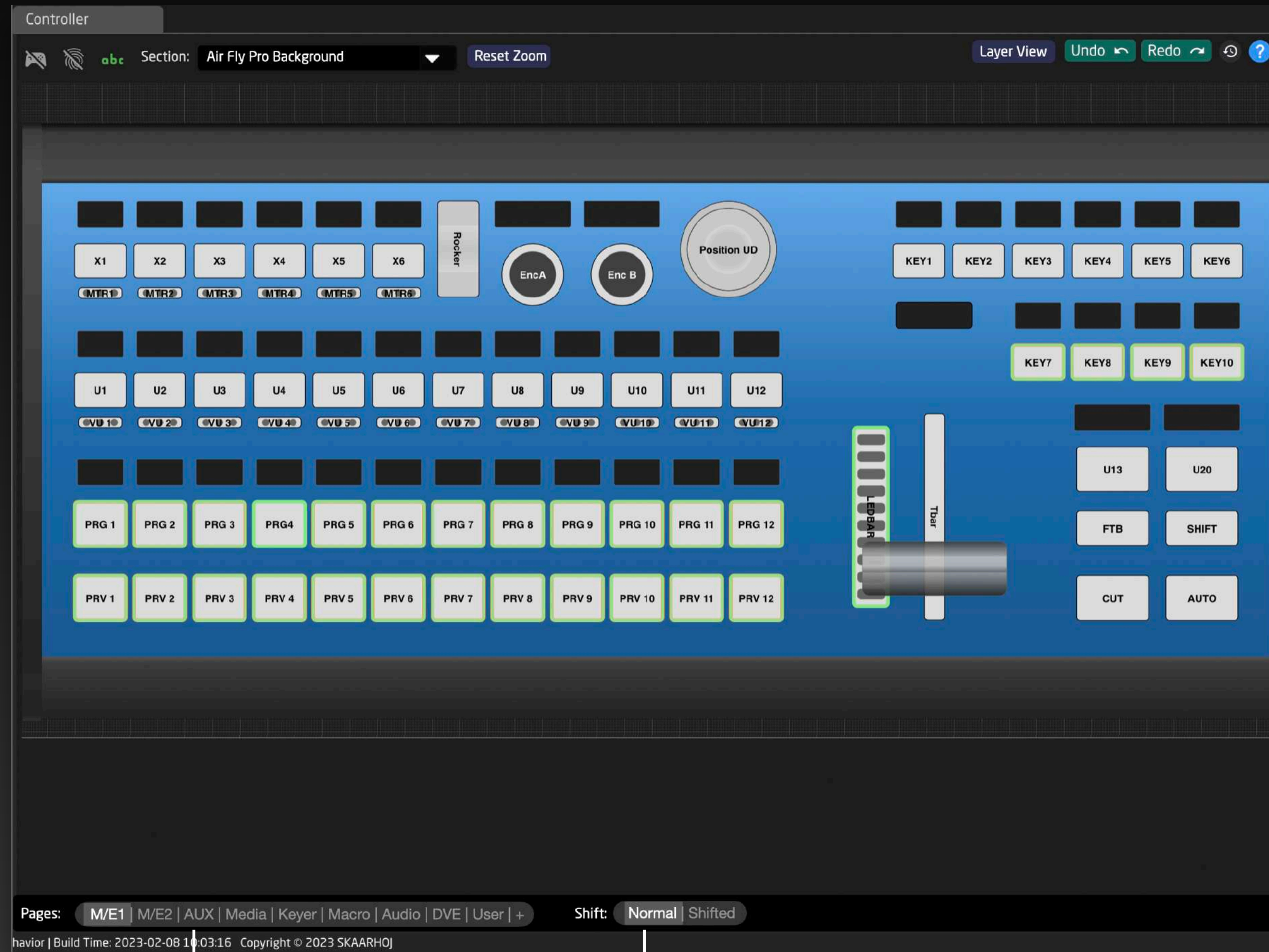
Customizing configurations is an advanced option, as default configurations handle essential adjustments on the Home Screen. Reactor's Configuration tab displays your controller graphically, enabling you to click on any hardware component to assign a behavior.

Components can be organized into sets of pages for additional functionality, allowing the creation of menu structures and adaptable control surfaces. Creating structures with pages and shift levels is generally straightforward, as you can assign behaviors specifying actions and feedback with a single click. The Configuration tab offers a user-friendly approach, grouping the controller's hardware components into sections.



Section View

Coming soon

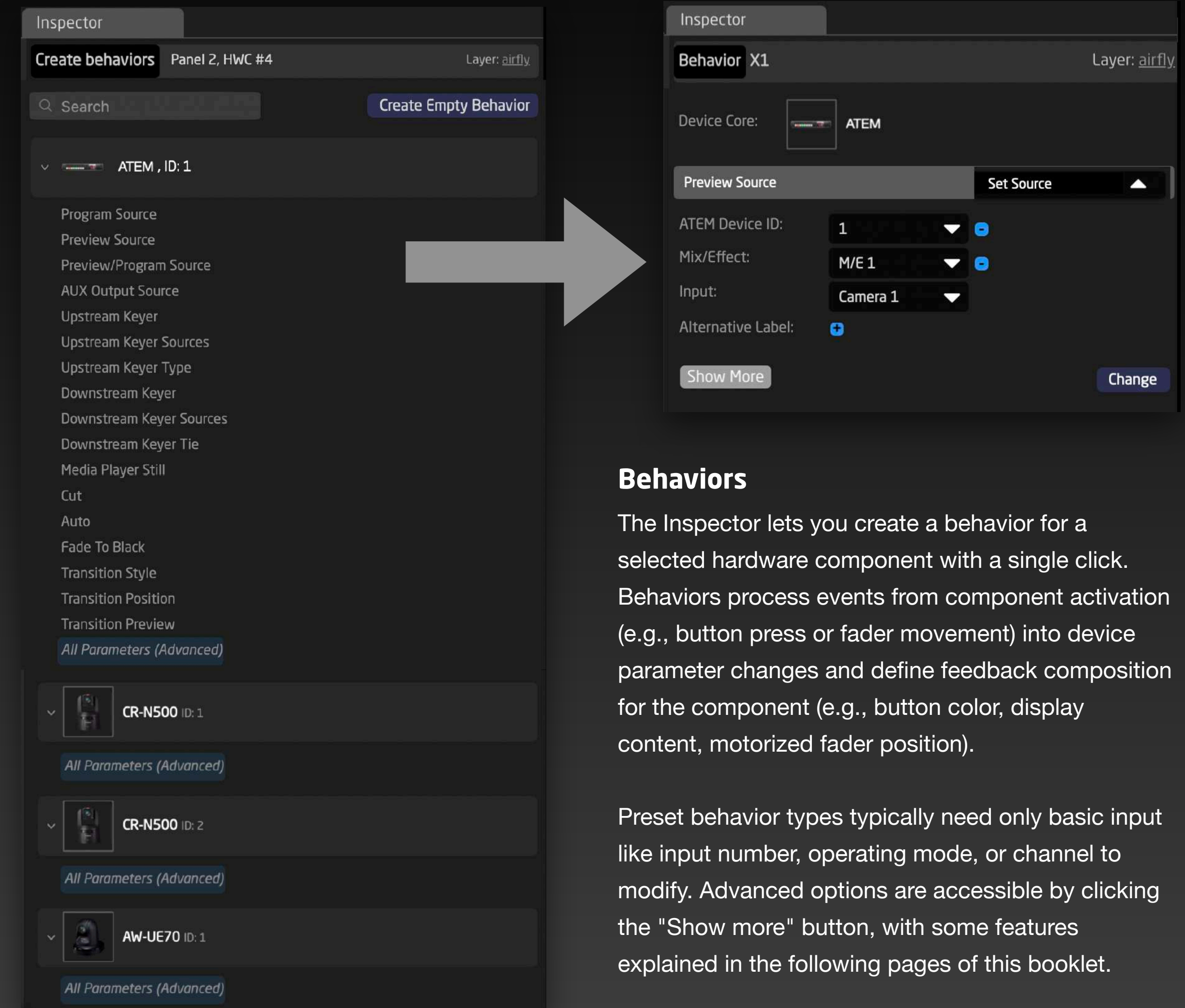


At the bottom of the screen, you can navigate between pages and add more pages within a given section.

Some sections can also include a Shift level for each page. The active page/shift level is where you add and edit behaviors in the Inspector.

Inspector

Coming soon



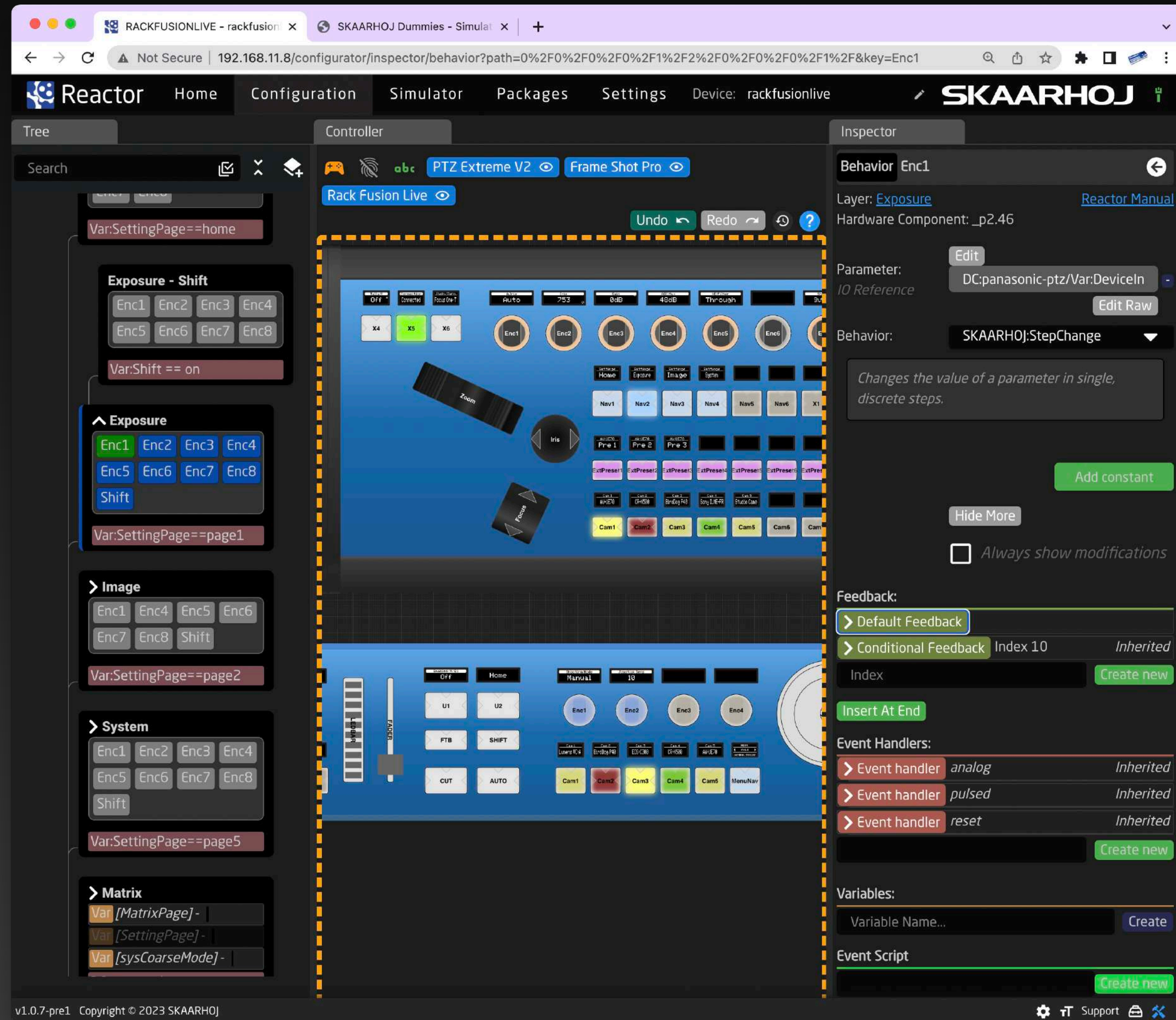
Behaviors

The Inspector lets you create a behavior for a selected hardware component with a single click. Behaviors process events from component activation (e.g., button press or fader movement) into device parameter changes and define feedback composition for the component (e.g., button color, display content, motorized fader position).

Preset behavior types typically need only basic input like input number, operating mode, or channel to modify. Advanced options are accessible by clicking the "Show more" button, with some features explained in the following pages of this booklet.

Layer View

Layer View shows the configuration as a nested layer structure of active or inactive behaviors based on their positions and visibility conditions. Layers act as the engine enabling interaction schemes in Reactor.



Behaviors like U1, U2, etc., are organized in layers, with active layers and behaviors highlighted in blue. Layers can be nested indefinitely; for example, "Navigation" and "PTZ Section" layers are children of the "Air Fly Pro Background" layer. An arrow on "PTZ Section" signifies it has child layers as well.

Layer visibility is crucial in Reactor's behavior selection for specific hardware components. Layers with visibility conditions, such as "Var:Shift == on", indicate the variable Shift must be set to 'on' for the layer and its behaviors to be active.

The "Air Fly Pro Background" layer has multiple behaviors assigned to different hardware components, including AUTO, CUT, DISP, KEY1, etc. A layer can also hold more than just behaviors, containing variables used to control menus, layer visibility (State and Shift), M/E Row selection for a switcher, device ID or camera number selection, and so on.

Master Behaviors serve as foundations for behaviors assigned to hardware components. By referencing a master behavior, a hardware component behavior can customize itself by changing specific attributes only, such as constant values (inputs, M/E row, channel, device ID, etc.), button color, or display label.

KeyMaps associate user-friendly aliases (e.g., CUT, AUTO, DISP) with a hardware component's numerical ID. KeyMaps enable entire configurations created for one panel ID to be remapped to another.

Layers can incorporate other layer files in the system, enabling the reuse and composition of larger configurations from smaller libraries of configuration.

The layer structure is referred to as a tree, with its root at the bottom.

Event Handlers

Behaviors can contain event handlers, defining actions when buttons are pressed or knobs are turned.

Handlers can filter on button edges and set values or cycle options. Sequences of actions with delays can also be configured. Unexpected triggers can be pre-processed and transformed into other triggers.

The screenshot shows the 'Event Handlers' section of the Inspector. It features a list of handlers: 'Event handler trigger' (selected), 'Event handler Fader', and 'Event handler Joystick'. The 'trigger' handler is configured with a 'Binary' type, 'ActDown' binary type, 'Top' edge filter, 'Cycle Up and Roll Over' set mode, and set values of '[1,2]'. It also has an 'Active If' parameter set to 'IO Reference' and a description 'Generate a Trigger'. Below the list is a 'Create new' button.

The screenshot shows the 'Feedback' section of the Inspector. It displays a list of feedback items: 'Default Feedback', six 'Conditional Feedback' items with indices 10, 11, 30, 40, 1000, and 1001, and one 'Conditional Feedback' item with index 50. The selected 'Conditional Feedback' item (index 50) has an 'Active If' condition: 'DC:bmd-atem/Var:DeviceIndex/ProgramInputVideoSourceAvail/Var:MErow/Behavior:Const:Input == false'. It also shows settings for 'Description', 'Intensity' (Dimmed), 'Color' (AMBER), 'Blink Pattern', 'Title' ((x) Alt Title), 'Title Font', and 'Text Size'.

Feedback

Default Feedback determines a behavior's appearance, such as color and display content. Conditional feedback offers alternative settings based on parameter values and conditions similar to layer visibility.

The screenshot shows the 'Variable' configuration panel for '[Shift] - Shift'. It includes a 'Layer' field set to 'test', a 'Name' field set to 'Shift', and a 'Description' field containing 'Drives the shift layer visibility'. The 'Default to first' checkbox is checked. The 'Type' is set to 'Options'. The 'Value' field shows 'off' and 'on', and the 'Label' field shows 'Normal' and 'Shift'. There are buttons for 'Add option', 'Show More', and 'Show JSON'.

Variables

Variables are runtime-adjustable values in Reactor, modifiable via buttons or knobs. They can have friendly names and descriptions, specific options or integer value ranges, and default values. Persistent storage maintains variable values after a panel reboot.

Variables power menus, select cameras and devices, enable modes for buttons and knobs, and store device names. They are defined within the layer tree and valid only within their branch.

Variables and Constants

"Variables" might be unfamiliar to some; simply consider them as values that can change during panel activity, modified by panel events, and used for various purposes like making layers visible or selecting cameras. Constants, unlike variables, remain fixed in the configuration and cannot change due to panel events. They are used in mapping tables or "Constant Sets" from the Home Screen.

Scripting

Reactor features a built-in JavaScript-based scripting engine with an intuitive in-browser editor for custom automation and complex actions. Users can retrieve and set IOreference values and use the sleep command to control the flow. The scripting engine, currently in beta, unlocks endless creative possibilities.

```

Auto Keyer

Enter Maximum time to run

[Save current File] [Format Code]

File is saved ✓

1 function USKLabel(a) {
2   return a == 0 ? "BKGR" : "USK" + a;
3 }
4
5 var event = GetEvent();
6 if (event.Binary != undefined && event.Binary.Pressed) { // Only start on butto
7
8   var usk = parseInt(GetIOReferenceFirstValue("Behavior:Const:USK"));
9   console.log("USK from constant: " + usk);
10
11  var meRow = parseInt(GetIOReferenceFirstValue("Var:MErow"));
12  console.log("meRow from variable: " + meRow);
13
14  // Pick up current transition states:
15  var nextTransitionStates = [];
16  for (a = 0; a < 5; a++) {
17    nextTransitionStates[a] = GetIOReferenceFirstValue("DC:bmd-atem/1/Transiti
18    console.log("Store state for " + USKLabel(a) + ": " + nextTransitionStates
19  }
20
21  // Set new transition states (starting with USKs since we need the background
22  for (a = 4; a >= 0; a--) {
23    var newValue = a == usk ? "true" : "false";
24    if (nextTransitionStates[a] != "----" && nextTransitionStates[a] != newValu
25      console.log("Change " + USKLabel(a) + " from " + nextTransitionStates[
26      SetIOReferenceValues("DC:bmd-atem/1/TransitionNextTransition/" + meRow
27      var ok = false;
28      for (wait = 1; wait <= 100; wait++) {
29        if (GetIOReferenceFirstValue("DC:bmd-atem/1/TransitionNextTransiti
30          ok = true
31          break;
32        }
33        Sleep(5);
34      }
35      if (!ok) {
36        console.log("Error setting DC:bmd-atem/1/TransitionNextTransition/
37      }
38    }
39  }
40
41  SetIOReferenceValues("DC:bmd-atem/1/Auto/" + meRow + "/"); // Trigger Auto tr
42  Sleep(100); //
43
44  console.log("Waiting for transition to complete...");
45  var ok = false;

```

Generators

Generators auto-create layers and behaviors using mapping table data for easy camera addition, input rearrangement, and preset page generation. They enhance default configurations for seamless experiences.

Tree

Search

- Cam 1
 - Cam1 Cam2 Cam3 Cam4 Cam5
 - Var:CameraPage == 1
- Cam 2
 - Cam1 Cam2 Cam3 Cam4 Cam5
 - Var:CameraPage == 2
- Camera Selector
 - Cam1 Cam2 Cam3 Cam4 Cam5
 - Var: [CameraPage] -
 - Gen Type: Behaviors
 - Var:ShowCamSelector == show
- SKAARHOJ.Devices.Vaddio.StdClass.Basic
 - Layer Files 1 layer imported
 - Var:LinkSelector == SKAARHOJ.Devices.Vaddio.StdClass.Basic
- SKAARHOJ.Devices.VISCA-Allstars.StdClass.Basic
 - Layer Files 1 layer imported
 - Var:LinkSelector == SKAARHOJ.Devices.VISCA-Allstars.StdClass.Basic
- No name
 - Layer Files 1 layer imported
 - Var:LinkSelector ==
- Camera control layers
 - Var: [LinkSelector] -
 - KeyMap Mapped aliases: 5
 - Gen Type: LinkIDselector

Inspector

VirtualTrigger Input1 To Kumo Out 2 Input 2

Layer: Root Layer - Virtualtriggers

Name: Input1 To Kumo Out 2 Input 2

Description: Description

Mode: Binary

In Binary mode a condition will be interpreted like a Binary trigger (basically imitating a button). While the condition is true the VirtualTrigger is "pressed" and while its false it is "released"

If...

Condition: DC:bmd-atem/1/ProgramI

Check Edit Raw

Then...

Behavior: Behavior DC:aja-kumo/1/routeInputToOutput/2/

Show More Show JSON

Virtual Triggers

Virtual Triggers in Reactor are software-generated events activating actions or behaviors, functioning like physical triggers but from internal processes or connected devices. This allows advanced customization, automation, and complex control scenarios.



Blue Pill Server

Blue Pill, SKAARHOJ's next-gen platform, transforms broadcast and AV control. Integrated into most panels and available as a compact server, it performs tasks like thumbnail frame grabbing, enabling UniSketch panels access to the Blue Pill ecosystem, and converting external devices into Raw-Panel compliant devices. For example, network-enable Stream Deck models as Raw-panel devices to complement SKAARHOJ broadcast panels by connecting them to Blue Pill and adding them to Reactor's panel collections.



```
Inspector
Various
Name: Menu
Description: 4-way menu for PTZ Fly with Sony BRC-X400, e
Active If: +
Constant Set: +
Hide JSON
Show Parent Behavior Format Save
1
2 "Name": "Menu",
3 "Description": "4-way menu for PTZ Fly with Sony BRC-X400, e
4 "IOReference": {},
5 "EventHandlers": {
6   "adjustmenu": {
7     "Description": "Cycle menu pages",
8     "AcceptTrigger": "Binary",
9     "EventPreProc": {
10      "B2B": {
11        "InputEdge": {
12          "Default": {
13            "ActDown": {
14              "OutputTrigger": "None"
15            },
16            "ActUp": {
17              "TimeWindowToPrevTrigger": 1000
18              "OutputTrigger": "ActUp"
19            }
20          }
21        }
22      }
23    },
24    "BinaryType": "ActUp",
25    "BinaryEdgeFilter": "Top",
26    "BinarySetMode": "CycleUpRollOver",
27    "BinarySetValues": {
28      "Raw": "Var:SettingPage:All"
29    },
30    "IOReference": {
31      "Raw": "Var:SettingPage"
32    }
33  },
34  "engMenu": {
35    "Description": "Toggle Engineering Menu",
36    "AcceptTrigger": "Binary",
37    "EventPreProc": {
38      "B2B": {
39        "InputEdge": {
40          "Default": {
41            "ActDown": {
42              "Repeat": "Delayed",
43              "RepeatDelay": 1000,
44              "MaxRepetitions": 1,
```

JSON

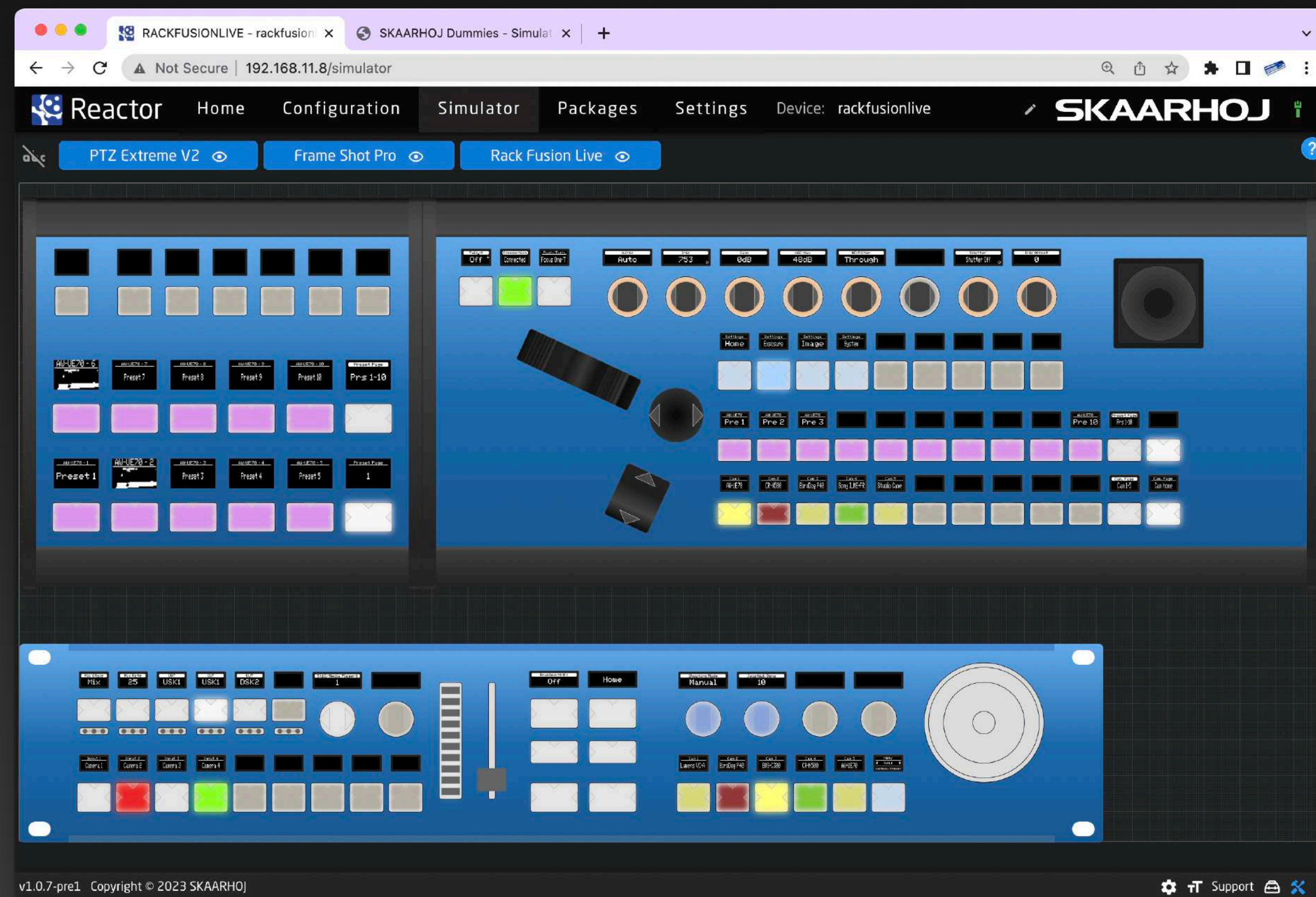
Edit Raw!

Reactor's configuration system allows seamless transition between a modern web UI and a code editor for direct JSON data modifications. This flexibility is appreciated for speed and accuracy. Reactor's built-in editor understands symbols, field names, and permitted values for fast configuration "coding." Advanced search and replace features eliminate external code editor needs, making Reactor an all-in-one configuration management solution.

```
"Name": "Default BLUEPILL Configuration",
"ActiveIf"
Layers
ManualDescription
MasterBehaviors
Variables
FlagGroups
MetaData
ConstantSets
FeedbackDefault
Generator
Template
VirtualTriggers
```

Simulator

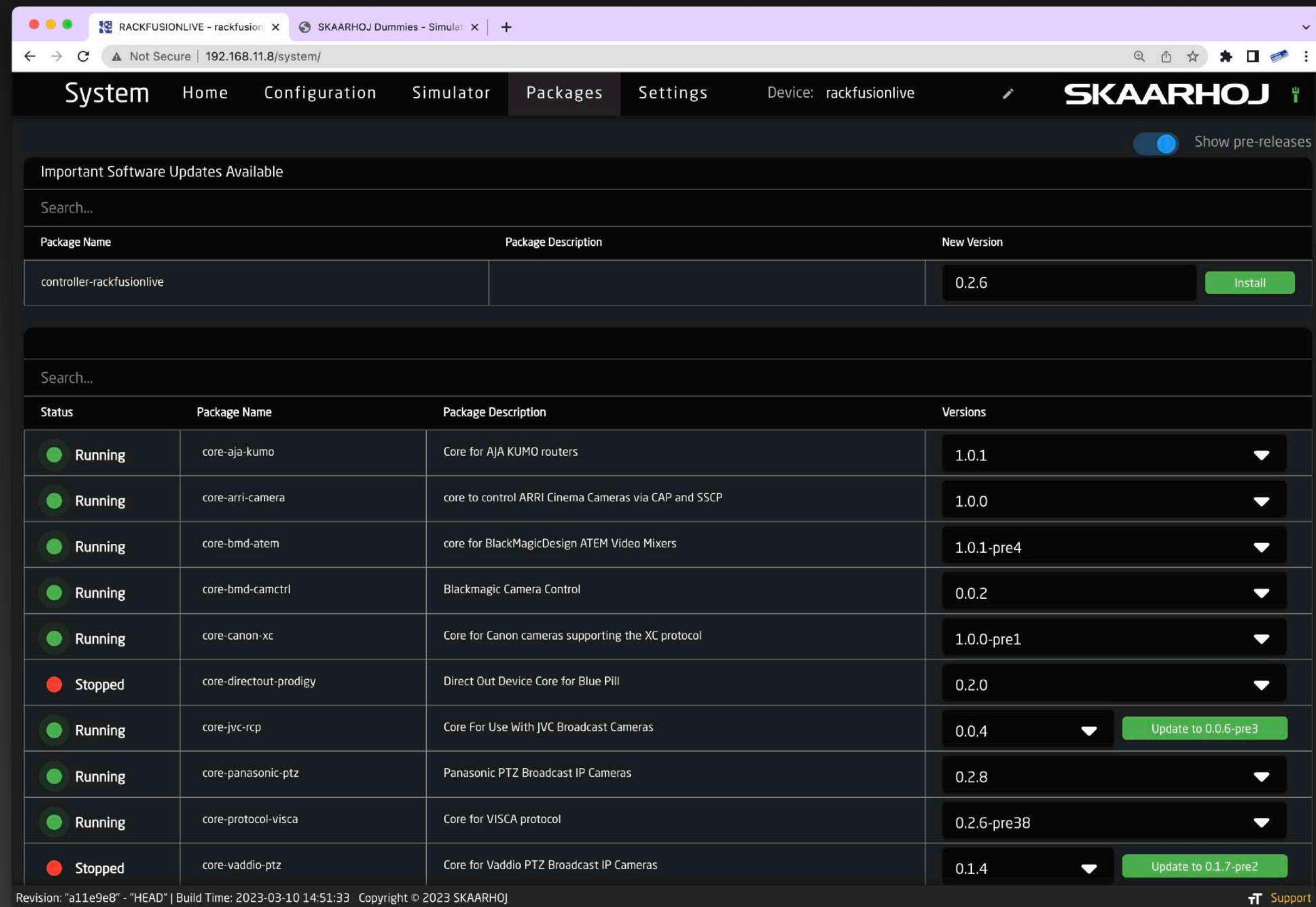
Reactor's advanced simulator provides a virtual, browser-based panel version for configuration, remote training, and occasional remote operation. The Simulator displays panels on their canvases, presenting modular controllers as a cohesive surface for seamless interaction.



PTZ Extreme

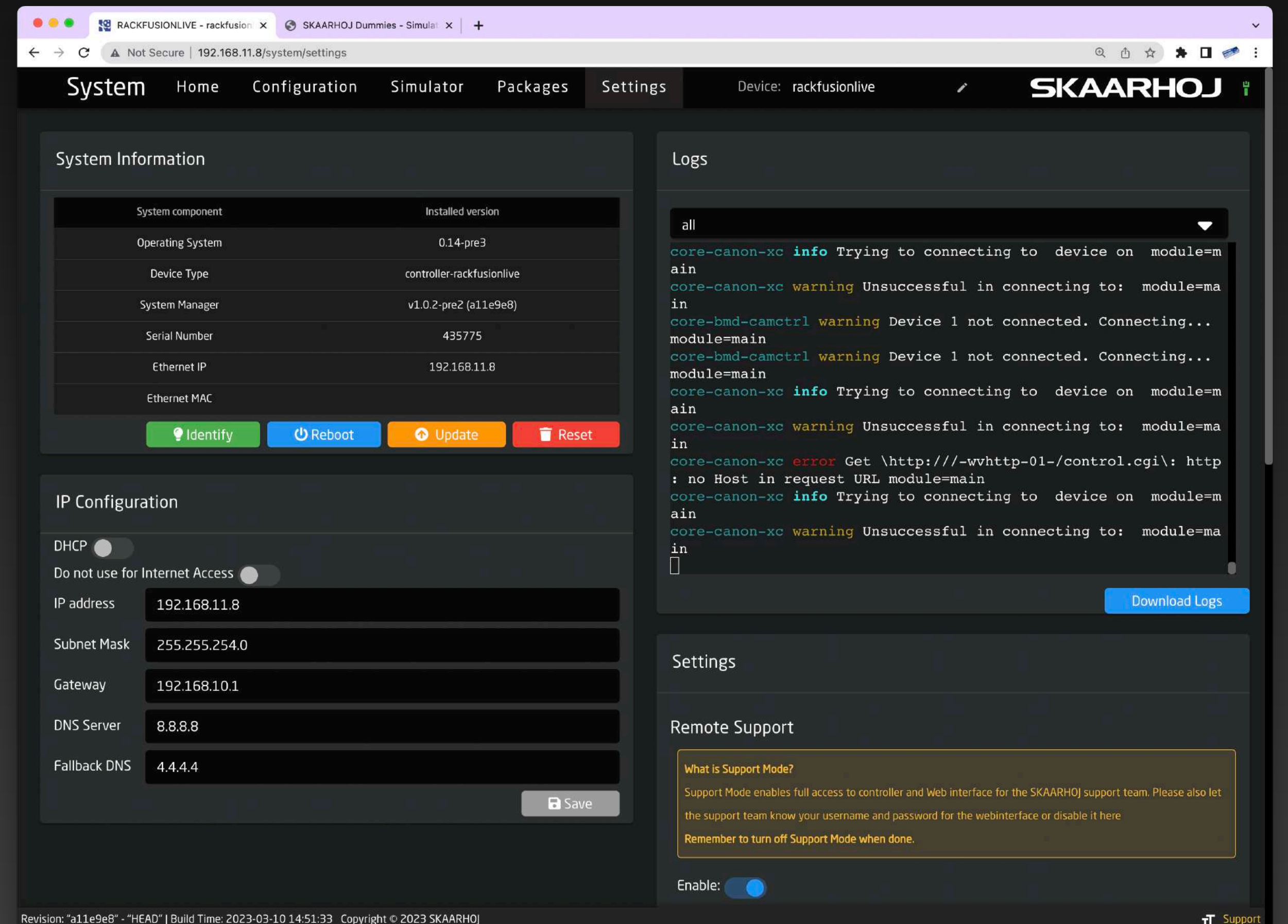
Packages

The Packages tab in a Blue Pill device, part of skaarOS, serves as a package manager to install and update software packages. Connecting to SKAARHOJ's online repository for easy access or accepting uploaded package files for offline panels, it streamlines the process of managing your panel's software.



Settings

The Settings tab, a feature of skaarOS, allows configuring the panel's IP address, viewing system logs, updating the operating system, and rebooting the device. It also enables WiFi access, activates Remote Support, and unveils expert features with Advanced Mode. The Settings tab offers a comprehensive and user-friendly interface for managing your panel's core settings and optimizing its performance.





RCP Pro



This Booklet