



Kaleidescape Programming Manual for Savant

Rev. 1, December 2021

Strato Custom Profile version - 14.25

© 2021 Kaleidescape, Inc. All rights reserved.

Kaleidescape and the Kaleidescape logo are trademarks of Kaleidescape, Inc. and are registered in the United States and certain other countries. Other third party product and brand names are the trademarks of their respective owners and may be registered in certain jurisdictions.

This document is for informational purposes only. While Kaleidescape believes that the information contained herein is accurate as of the first date of publication (March 1, 2021), Kaleidescape makes no warranties, express or implied, regarding the accuracy or completeness of the information contained herein and Kaleidescape shall have no obligation to provide updates to this information in the future

Table of Contents

1. [Overview](#)
 - 1.1. Introduction
 - 1.2. Before You Begin
2. [Kaleidescape Player Setup](#)
 - 2.1. Network Setup
3. [Savant Deployment](#)
 - 3.1. Control Deployment
 - 3.2. Savant Pro App Deployment
4. [Savant Triggers and Automation](#)
 - 4.1. Savant Triggers Deployment - CurrentPlayStatus
 - 4.2. Savant Triggers Deployment - IsInEndingCredits
 - 4.3. Savant Triggers Deployment - IsInIntermission
 - 4.4. Savant Triggers Deployment - CurrentAspectRatio
5. [Custom Buttons](#)
 - 5.1. Adding a Custom button to the Pro App and Remotes
6. [Troubleshooting](#)
7. [Kaleidescape App Volume Control](#)
8. [Appendix](#)

1 Overview

1.1 Introduction

This programming manual provides an overview of the Kaleidescape control profile and touch panel UI designed for Savant controllers. It explains everything a Savant programmer needs to know to successfully integrate a Kaleidescape system into any Savant project. The profile described in this document is a custom profile based on the most recent Savant profile released. This profile includes custom functions, state variables and Kaleidescape App Volume Control. This profile may not be supported by Savant. If you have basic control issues use the Savant release of the profile to troubleshoot with Savant, if required.

1.2 Before You Begin

Before you begin, please read through this document and make sure you have all the necessary equipment and information ready to complete the deployment.

The following Savant documents may be useful while using this deployment guide:

1. Savant® Pro Host Deployment Guide - Document Number 009-1776-02
2. Savant® Rack Mountable Hosts Deployment Guide - Document Number 009-1914-00
3. Automator Best Practices - RacePoint Blueprint™ Programming Guide
4. State Triggers - RacePoint Blueprint Programming Guide

2 Kaleidescape Player Setup

In this example we will be using TCP/IP control from Savant. The Kaleidescape players will need to be set to a static IP address to ensure consistent control. The recommended setup would be using static IP reservations at the router level to avoid IP conflicts and DNS issues.

Network Requirements:

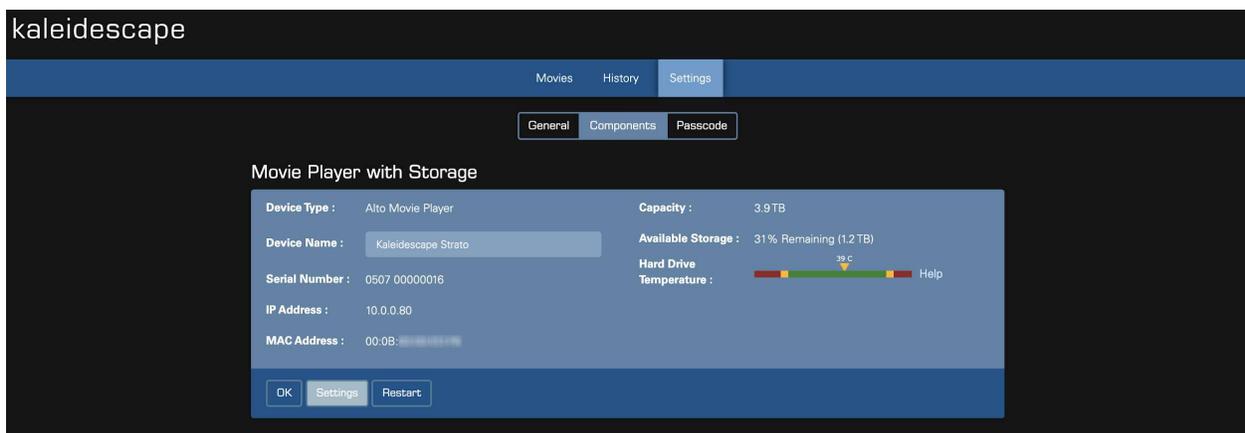
A Gigabit Ethernet network is recommended and is required when there is more than one Kaleidescape component. Connect Strato components using a shielded Ethernet cable. The built-in Wi-Fi may be used when a Strato S player is used as a single-component standalone system.

2.1 Network Setup via the Browser Interface

Wired network connection (recommended).

Note: if you assigned an IP address via a static reservation in your router, you can skip to step 3.

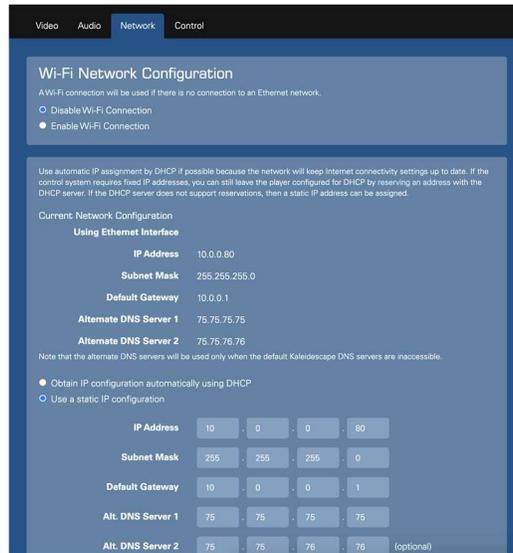
1. Select **Settings** from the top menu, then select **Components** and then **Settings** for the component you want to set up.



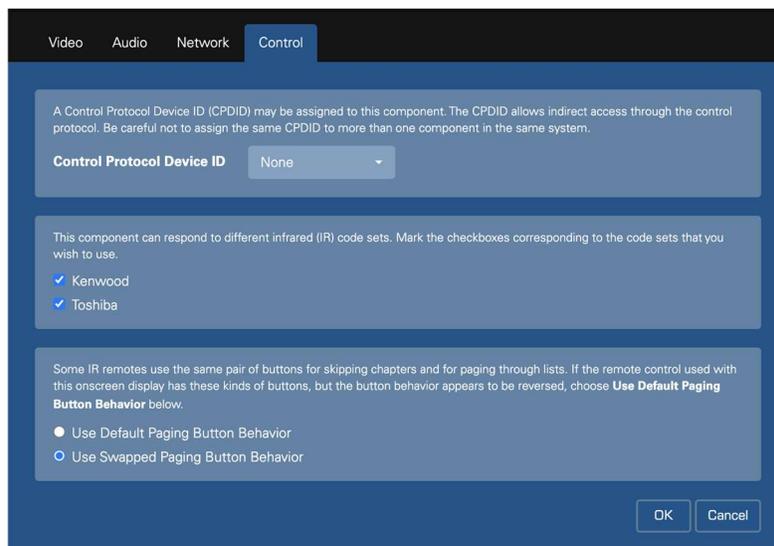
2. Select the **Network** tab from the menu. In most cases, the Strato player will automatically obtain an IP address from a DHCP server. Set your static IP, in this example the IP will be 10.0.0.80. Click OK.

Note - If you use this method to set a static IP address, please be aware of the following items:

- a. Make sure your static IP is unique on your network
- b. Be sure to set your Gateway and DNS properly - if not done properly the player may experience issues with accessing the internet.



3. Select the **Control** tab
4. The **Control Protocol Device ID** should be set to '**None**'.



Savant Profile notes:

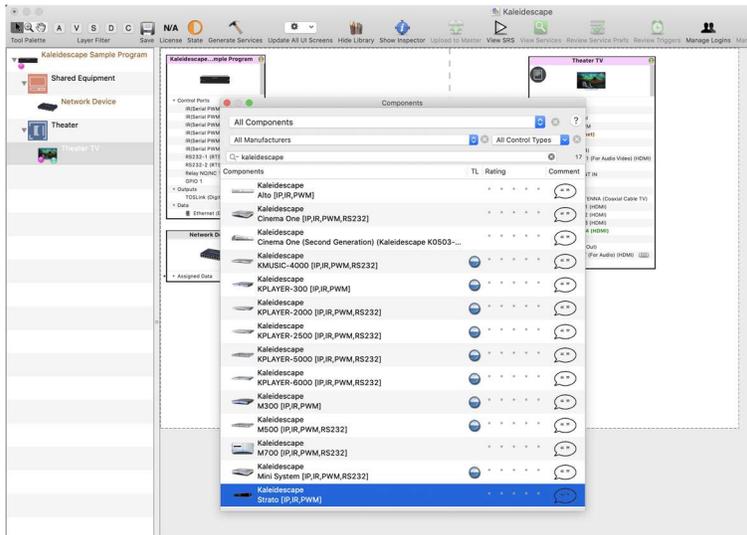
This device must have its Device ID set to 'None' or '1' depending on what your system allows (sometimes a restart of the Kaleidescape system is required for this to take FULL effect). This setting can be found in the Web Browser's Component Settings page. This profile assumes both feedback and control will come through as Device ID 1, so if a different Number (such as 3) begins each feedback message when looking at through RPMTerminal, it's likely this is set as the number you see.

3 Savant Deployment

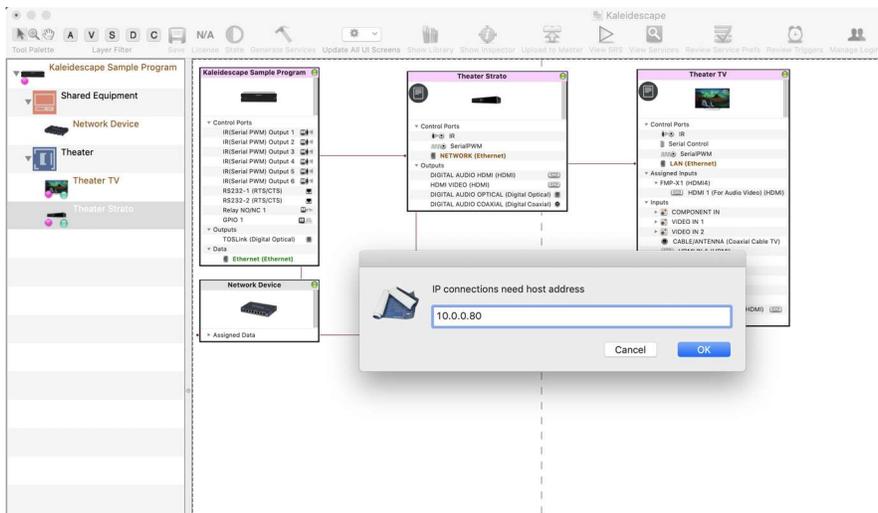
This section explains how to deploy a Kaleidescape Strato player for control using TCP/IP and how to customize the Savant Pro App and handheld remotes.

3.1 Control Deployment

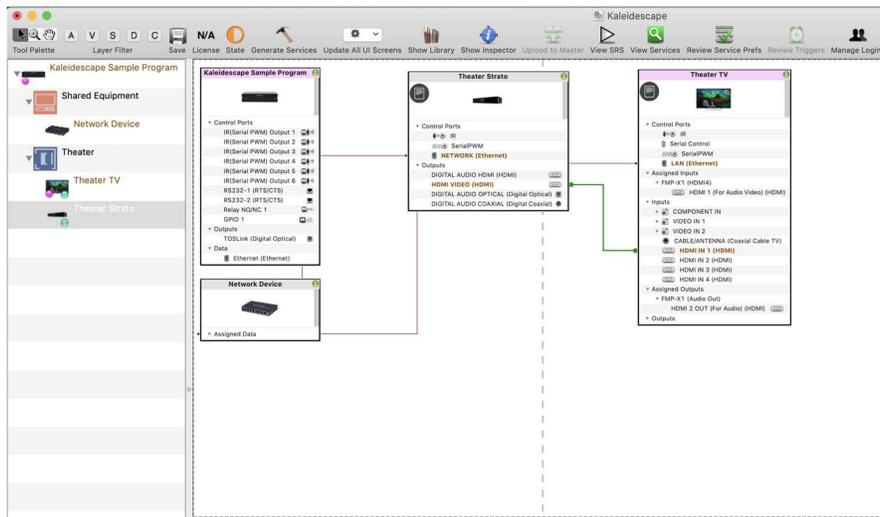
1. Open the library and search for *Kaleidescape*. Place the **Kaleidescape Strato** profile into your User Zone or Shared Equipment Zone, depending on your system configuration. Give the player a unique name. In this example we will be using a User Zone named "Theater," with a TV for the display and a name of "Theater Strato" for the player.



2. Place the Strato Player on to the Blueprint layout and connect the **Network(ethernet)** under **Control Ports** to a network device. This will prompt you to enter the IP of the Strato Player that was statically assigned to the player.



3. Connect the **HDMI Video (HDMI)** to your TV or other display device or video matrix you may be using. The Audio HDMI, Optical and Coaxial outputs may be used for connections to an audio controller.

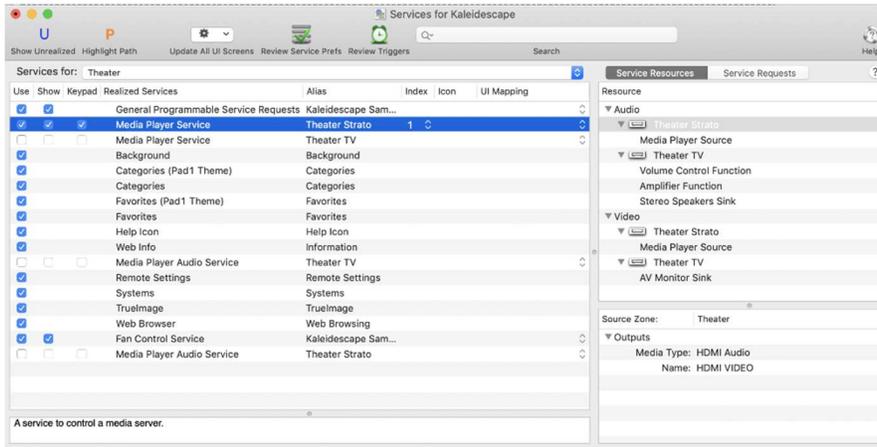


4. In Blueprint, go to **Tools > Branding Icons**. Select 'Kaleidescape' for the Strato Player. This will change the default logo to the Kaleidescape Logo on the Pro App UI. Save.

Component Name	Type	Brand
Theater TV	Multiple	-Select the Brand to Display-
Theater Strato	Media Server	Kaleidescape

These settings only affect the service icons that are displayed for the Cable, Satellite, or Media Server Services. The actual Provider will still need to be set in the Savant Pro app.

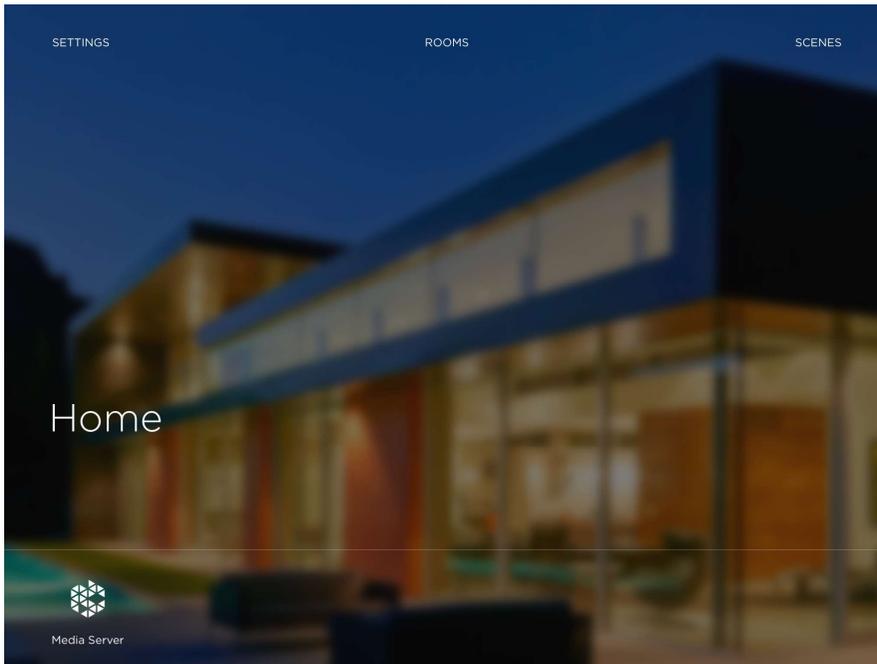
5. Generate Services and check your Strato services for correct service paths. Close this window.



6. Save and upload the configuration to the host.

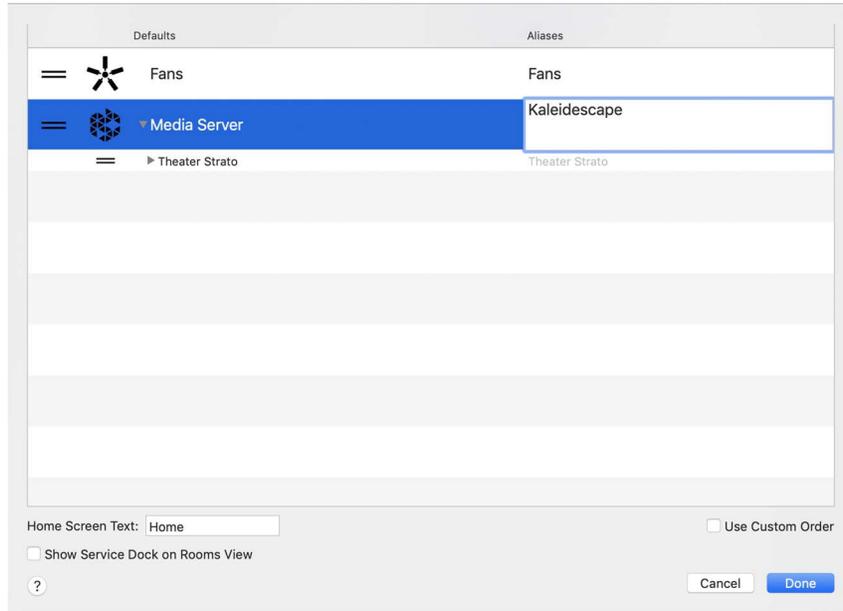
3.2 Savant Pro App Deployment

By default, Savant names the Kaleidescape service Media Server. To change this, modify the file after services have been generated.

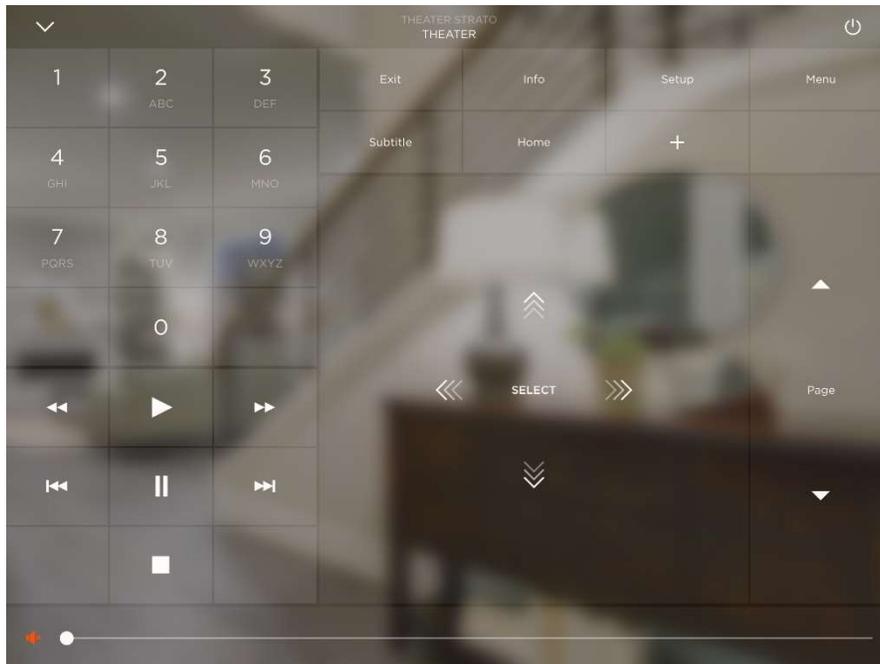


1. Go to **Tools > Review > Savant App Preferences...**

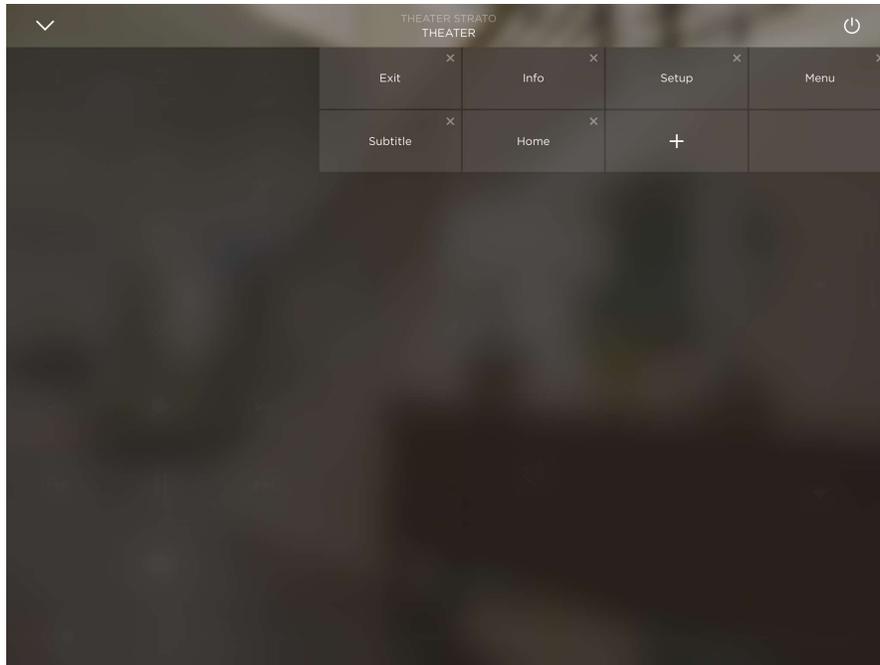
2. Change the Alias of the **'Media Server'** to what you want the client to see on the Pro App. Click done and upload your changes.



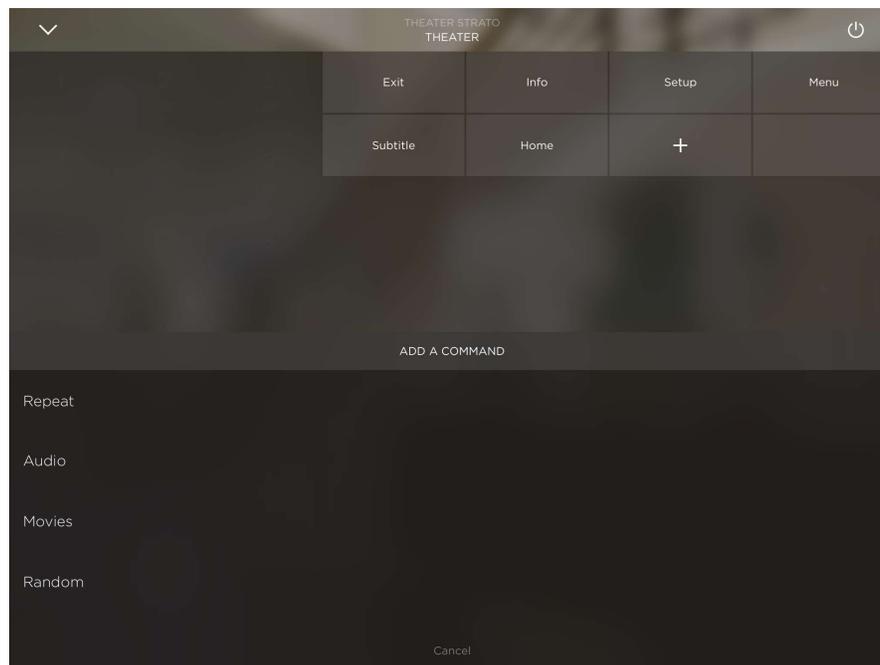
The default UI provides some buttons that may not be desirable to have. These buttons can be moved, deleted and added back directly on the UI.



1. To Move or Delete the default buttons, **press and hold** a button until all the buttons change state and are isolated. You can now move or delete them. Press and hold a button again to move it. Tap anywhere on the empty space to exit this mode.

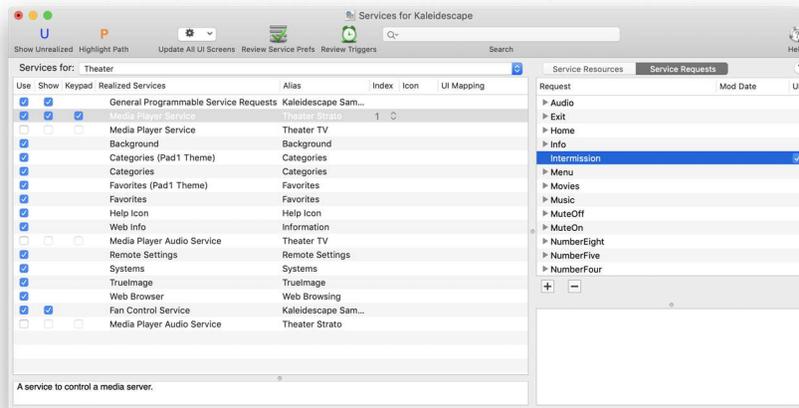


2. To add a button, press the '+' button. The current profile will allow the addition of the following buttons to the UI. Custom buttons can be added which are covered in [5.1 Adding a custom button to the Pro App and Remotes](#).
 - a. Repeat
 - b. Audio
 - c. Movies
 - d. Random
 - e. Music

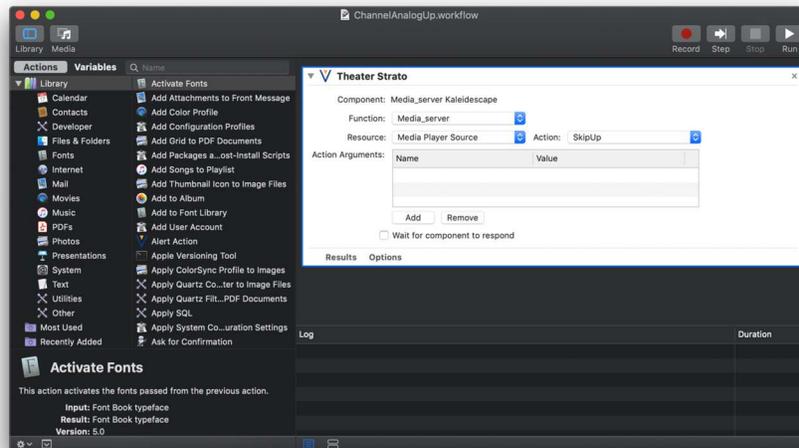


Savant handheld remotes: Some remotes do not have hard buttons for skip forward and skip back transport controls. To use buttons Channel Up and Channel Down on the remotes for these function follow these steps:

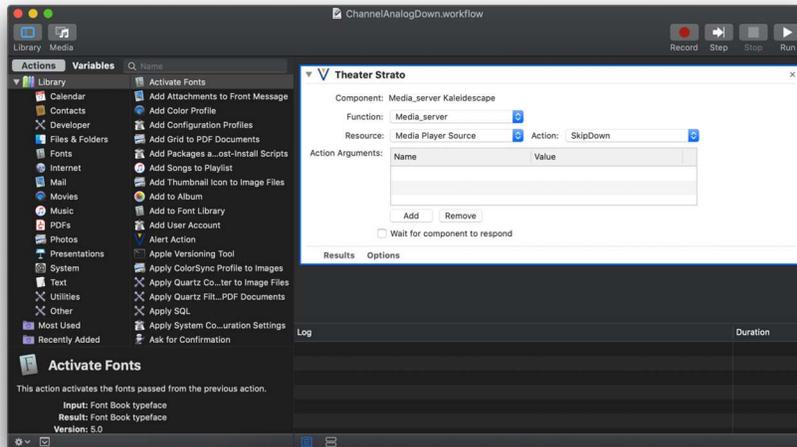
1. Open **View Services** for your blueprint project and go to your zone that has the Strato player. If View Services is not available you may have to generate services first.
2. Select your Strato player service and select **Service Requests** on the right side.
3. Add a new service request by clicking on the '+'.



4. Name the service request **ChannelAnalogUp**, case sensitive.
5. Double click your new service request to open it in Automator.
6. Search for your Strato player and double click on it or drag it to the workflow.
7. Select **'Media_server'** for the Function.
8. Select **'Media Player Source'** for the resource
9. Select **'SkipUp'** from the **Action** dropdown list.



10. Save your workflow and create a new service.
11. Name the new service request **ChannelAnalogDown**, case sensitive.
12. Double click your new service request to open it in Automator.
13. Search for your Strato player and double click on it or drag it to the workflow.
14. Select **'Media_server'** for the Function.
15. Select **'Media Player Source'** for the resource
16. Select **'SkipDown'** from the **Action** dropdown list.



17. Save and upload the changes.

4 Savant Triggers and Automation

This section demonstrates the use of the Savant State values from the Kaleidescape player profile to automate environmental elements such as lighting, masking and other 3rd party devices. Please refer to the **State Triggers - RacePoint Blueprint Programming Guide** on Savant's dealer portal for more detail on triggers and state values.

The following section is an overview of suggested automations for an immersive home cinema experience.

Customizing the Cinema Experience

1. **Start up the Theater** - Lighting zones are set such that physical navigation of the space is still relatively easy, and features and accents of the room are highlighted
 - a. Any lighting zone affecting the screen and perceived contrast brought to 10% or lower
 - b. Any step or pathway lighting zone brought to 70%
 - c. Any cove, column or accent lighting zone brought to 60%
 - d. All downlight/reading zones brought to 30%
 - e. Scene fades at 5-10 seconds
 - f. Close Shades if present
 - g. HVAC adjusted for movie watching
 - h. Kaleidescape user interface on screen
2. **Select and Start a movie** - Based on owner preference, this near dark experience allows family and friends to become lost in the movie, as the director intended; some minimal pathway lighting is provided
 - a. Any lighting zone affecting the screen and perceived contrast brought to 0%
 - b. Any step or pathway lighting zone brought to 10%
 - c. Any cove, column or accent lighting zone brought to 0%
 - d. All downlight/reading zones brought to 0%
 - e. Scene fades at 5-10 seconds
 - f. Screen Masking Adjustments if required
3. **Intermission On or Pause** - A quick break, this lighting scene again focuses on physical navigation of the space while maintaining most of the ambiance
 - a. Any lighting zone affecting the screen and perceived contrast brought to 10% or lower
 - b. Any step or pathway lighting zone brought to 60%
 - c. Any cove, column or accent lighting zone brought to 20%
 - d. All downlight/reading zones brought to 20%
 - e. Scene fades at 20 seconds
4. **Intermission Off or Un-Paused** - Quickly returns the room back to dark immersive experience
 - a. Any lighting zone affecting the screen and perceived contrast brought to 0%
 - b. Any step or pathway lighting zone brought to 10%
 - c. Any cove, column or accent lighting zone brought to 0%
 - d. All downlight/reading zones brought to 0%
 - e. Scene fades at 3 seconds
5. **Start of End Credits** - Slowly returns us back to lighting levels similar to our intermission look; credits and in-credits movie scenes are allowed to run in good contrast, physical navigation of the space is again made possible and features and accents of the room are highlighted
 - a. Any lighting zone affecting the screen and perceived contrast brought to 10% or lower

- b. Any step or pathway lighting zone brought to 60%
 - c. Any cove, column or accent lighting zone brought to 20%
 - d. All downlight/reading zones brought to 20%
 - e. Scene fades at 15 seconds or more
- 6. End of Movie and Return to On Screen Display** - Some additional lighting could be brought up while viewers decide whether to watch some special features or scenes from other movies. Masking/Lens may adjust, depending on CinemaScope mode.
- 7. Finished Watching Movies**
- a. Turn off Control System - Put the Kaleidescape Strato player into standby
 - b. Close the proscenium curtains
 - c. Open any shades
 - d. Adjust the HVAC
 - e. Lighting should adjust to pre-show levels - Some delay can be programmed to turn off lights, or a switch needs to be provided near the door

Using Savant State Values to Automate the Cinema Experience

Using the following Savant state values built into the profile, triggers can be set up to automate lighting, masking and other 3rd party devices. These examples are only a few of the many values that can be used in triggers to automate your experience.

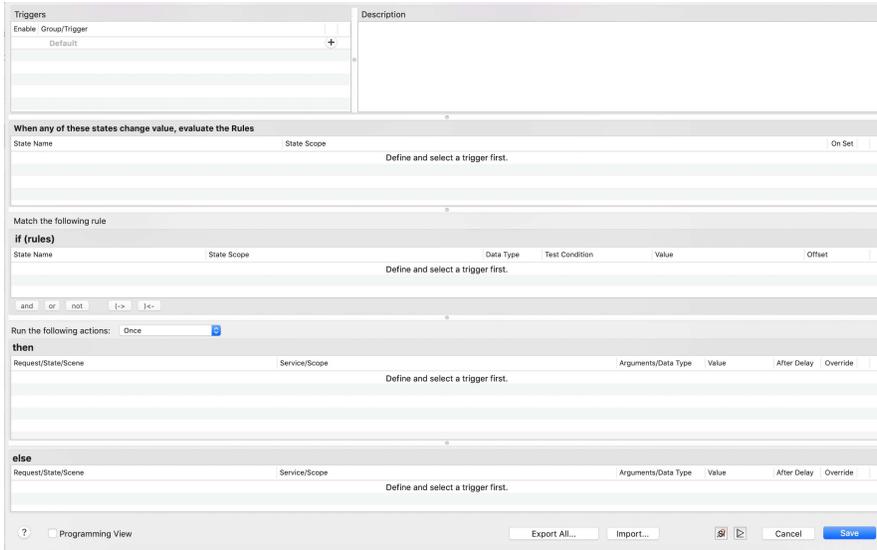
1. **CurrentPlayStatus** - state of movie playback - *string*
 - a. Play
 - b. Stop
 - c. Pause
2. **IsInEndingCredits** - Movie is in the Ending credits - *boolean*
3. **IsInIntermission** - Movie is in Intermission - *boolean*
4. **CurrentAspectRatio** - the aspect ratio of the video content - *string*
 - a. 1.33 - 4:3
 - b. 1.66 - European widescreen
 - c. 1.78 - 16:9
 - d. 1.85 - Widescreen
 - e. 2.35 - Cinemascope

Other Savant state variables useful for automation:

IsInKScapeInterface - state of the player on screen - *boolean*
 IsInMainContent - state of the player on screen - *boolean*
 Movie Title - Name of Movie playing - *string*
 CinemascopeMode - current Cinemascope mode - *string*
 CinemascopeMask - current aspect ratio of the masking - *string*
 CurrentTopMaskRelative_(0.1percent)
 CurrentBottomMaskRelative_(0.1percent)
 CurrentTopMaskAbsolute_(0.1percent)
 CurrentBottomMaskAbsolute_(0.1percent)
 CurrentTopMaskCalibrated_(0.1percent)
 CurrentBottomMaskCalibrated_(0.1percent)
 CurrentAspectRatio_Conservative

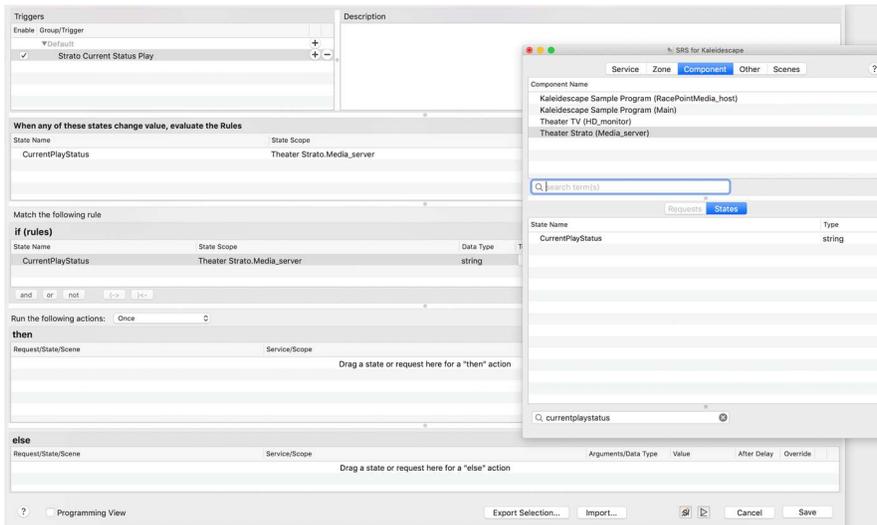
4.1 Savant Triggers Deployment - CurrentPlayStatus

1. Open **Review Triggers** in Blueprint. Click the '+' to add a new trigger and name it.

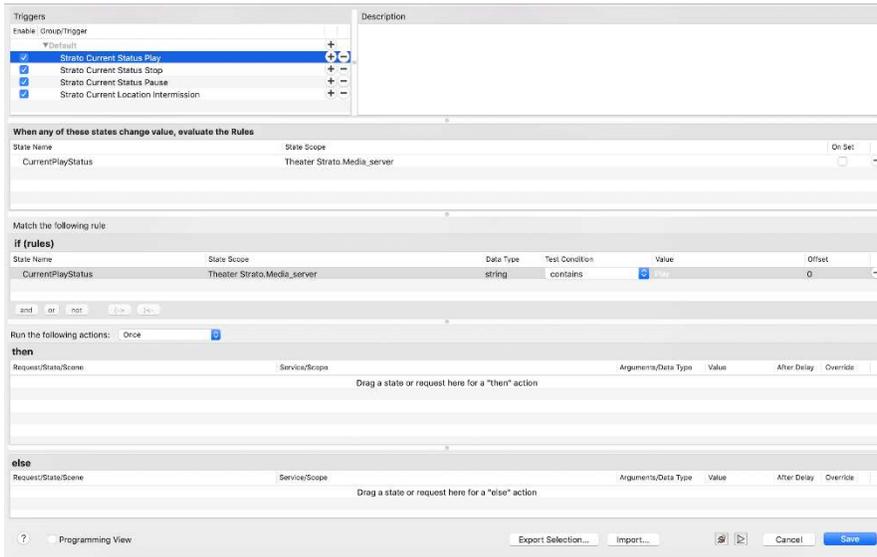


2. Click the **Services icon**  in the lower right corner to open the services window.

3. Locate the player under **Component** and select it. Use the lower search bar to search for **currentplaystatus** under State Name. Drag '**CurrentPlayStatus**' to the '**if (rules)**' section. This will automatically add an entry to the section above.

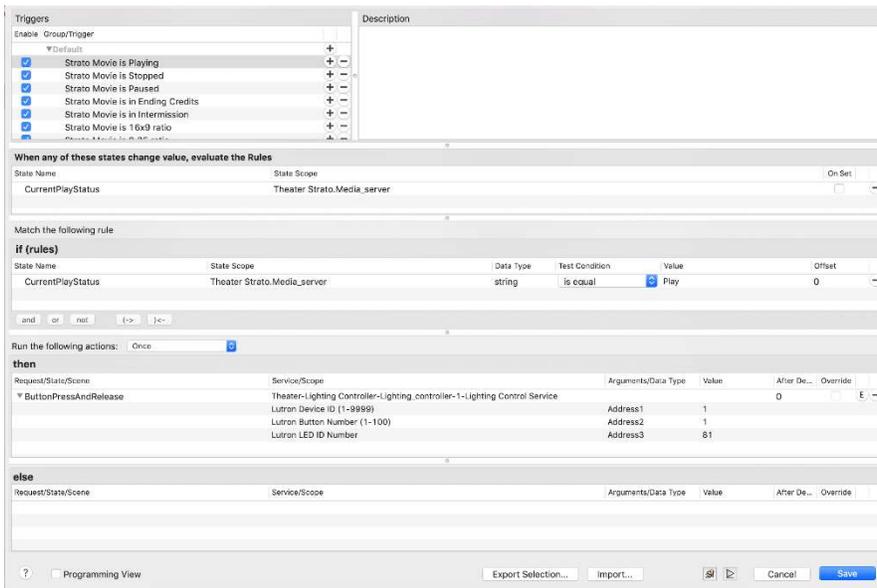


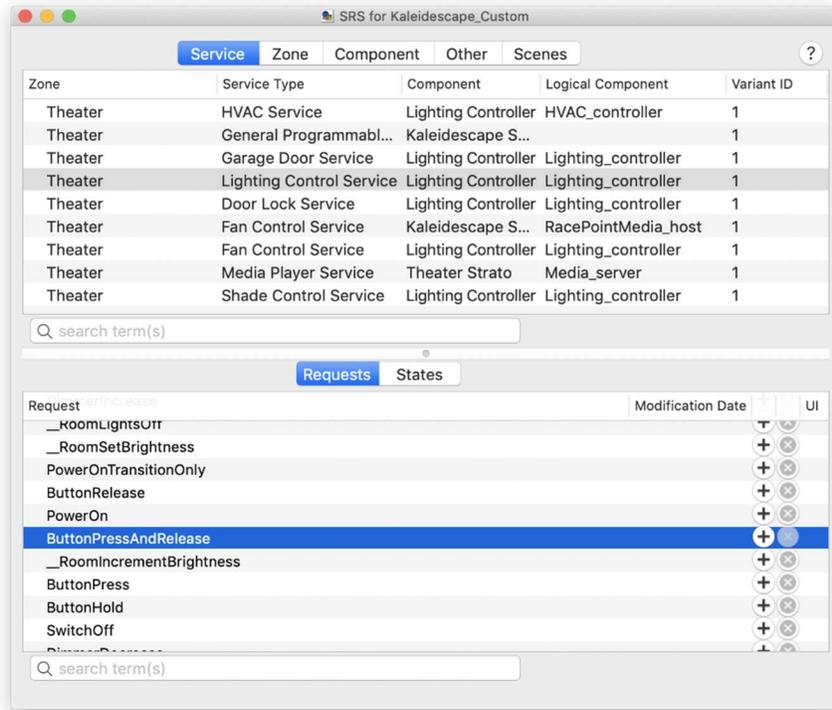
- Close the services window and change the test condition to **'contains'** or **'is equal'** and **"Play"** for the value.



- Place your lighting, shade or other 3rd party service in the **'Then'** section. In this example "Play" will send a command to a Lutron Lighting controller to press and release a Lutron button that is already programmed for the lighting scene desired for watching a movie.

- Click the **Services icon**  and search for your lighting controller under the **Service** tab. Drag the desired control request to the **'Then'** section. In this example, enter the values required for the lighting controller, Lutron Device ID, Lutron Button number and Lutron LED ID Number.

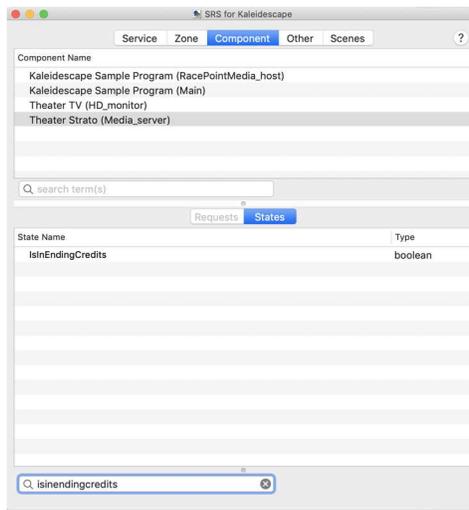




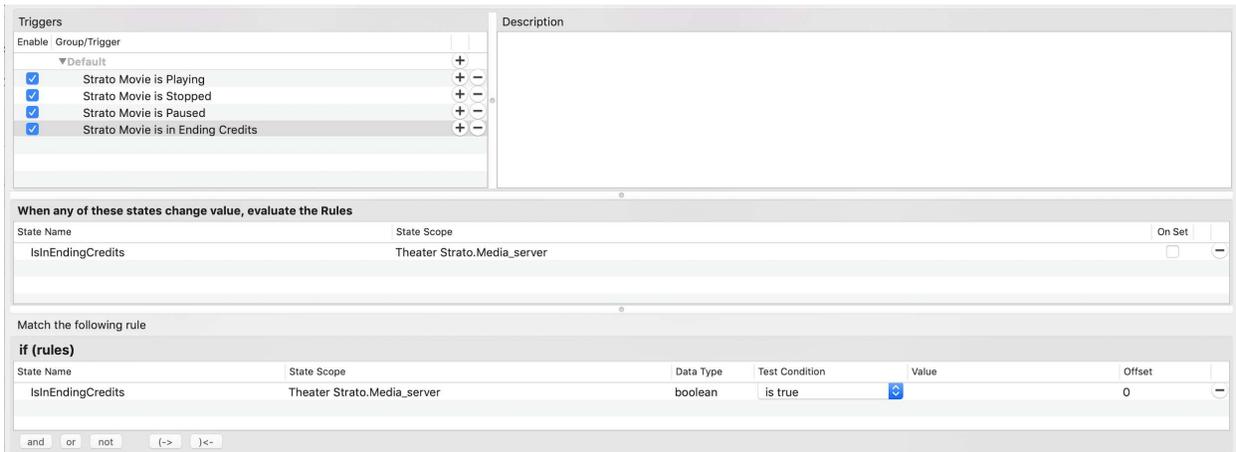
7. **CurrentPlayStatus** can be used for "Play", "Stop" and "Pause". Case sensitive values.

4.2 Savant Triggers Deployment - IsInEndingCredits

1. Follow steps 1 and 2 from above. Locate the player under **Component** and select it. Use the lower search bar to search for *IsInEndingCredits* under State Name. Drag '**IsInEndingCredits**' to the 'if (rules)' section. This will automatically add an entry to the section above.



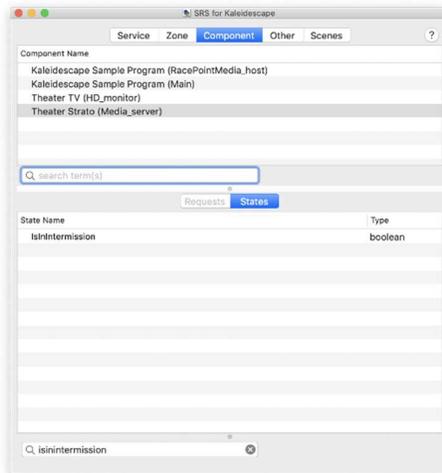
2. Close the services window and change the test condition to **'is true'**.



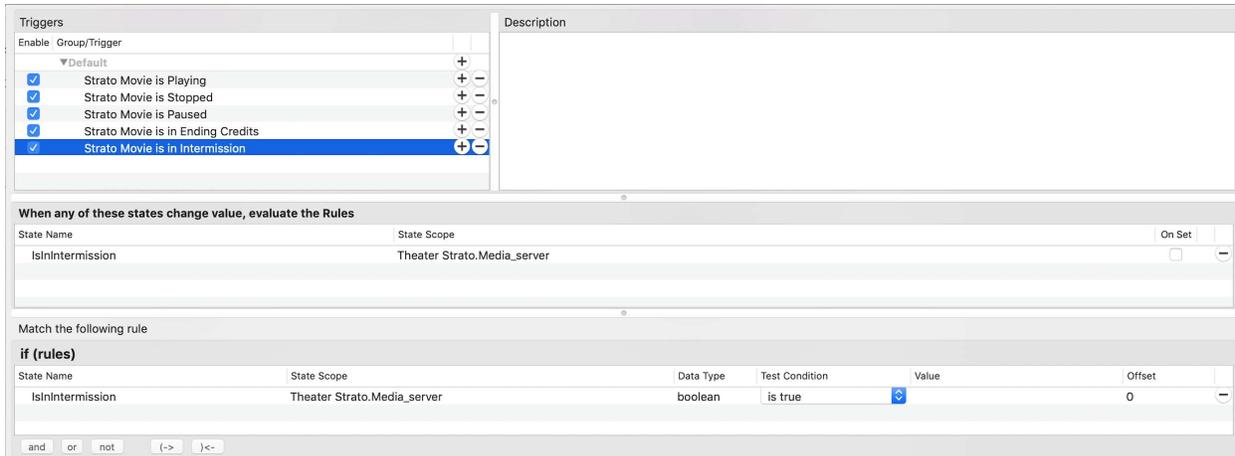
3. Place your lighting, shade or other 3rd party service in the **'Then'** section.

4.3 Savant Triggers Deployment - IsInIntermission

1. Follow steps 1 and 2 from above. Locate the player under **Components** and select it. Use the lower search bar to search for **IsInIntermission** under State Name. Drag **'IsInIntermission'** to the **'if (rules)'** section. This will automatically add an entry to the evaluation section above.

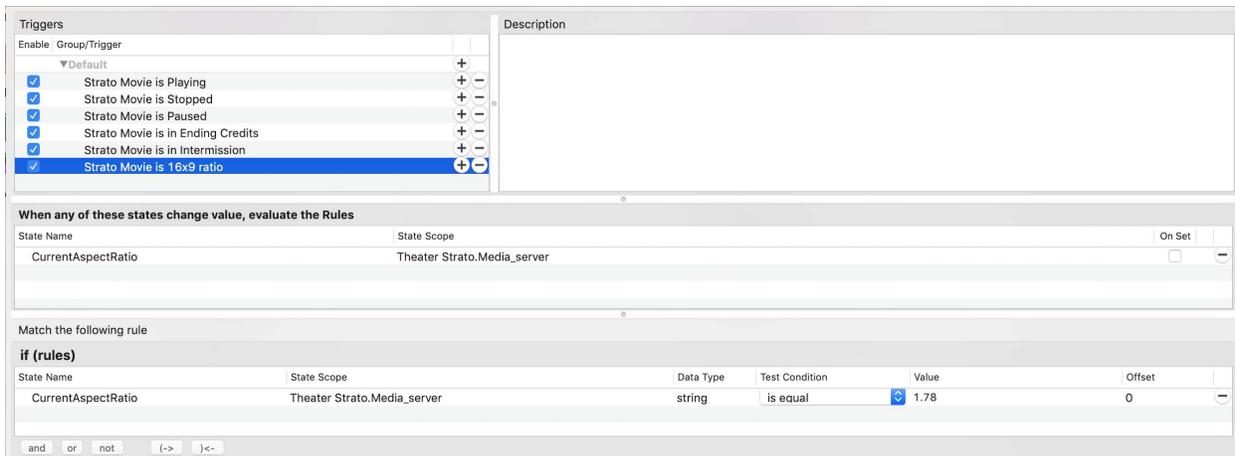


2. Close the services window and change the test condition to **'is true'**.
3. Place your lighting, shade or other 3rd party service in the **'Then'** section.
4. This trigger will work from the Intermission button on the Kaleidescape App or if you add a custom button to Savant for Intermission Control. Reference [5.1 Adding a custom button to the Pro App and Remotes](#).



4.4 Savant Triggers Deployment - CurrentAspectRatio

1. **Note:** If you are using the CinemaScope feature, use **CinemascopeMask** instead of **CurrentAspectRatio**, but the rest of these steps apply.
2. Follow steps 1 and 2 from above. Locate the player under Components and select it. Use the lower search bar to search for **currentaspectratio** under State Name. Drag **'CurrentAspectRatio'** to the **'if (rules)'** section. This will automatically add an entry to the section above. Close the services window and change the test condition to **'contains'** or **'is equal'** and **'1.78'** for the value.
3. Place your lighting, shade or other 3rd party service in the **'Then'** section.



4. Create a trigger for each of the following to track the ratio for each movie. Use this trigger to automate your masking and lens if applicable.
 - a. 1.33 - 4:3
 - b. 1.66 - European widescreen
 - c. 1.78 - 16:9
 - d. 1.85 - Widescreen
 - e. 2.35 - Cinemascope

Other Savant state values that can be used for masking automation

CinemascopeMode - Identifies the CinemaScope mode currently active

- a. No Mode
- b. 2.35 Anamorphic
- c. 2.35 Letterbox
- d. 2.35 Native

CinemascopeMask - Provides aspect ratio when a player is in CinemaScope mode

- a. 1.33
- b. 1.66
- c. 1.78
- d. 2.37
- e. 2.40

CurrentTopMaskRelative_(0.1percent)

CurrentBottomMaskRelative_(0.1percent) - Indicates top and bottom trim values, relative to the aspect ratio. Positive values indicate adjustment towards the center of the screen, negative toward the edge. For example, +010 means adjust a mask inward by 1% of the screen height, and -005 means adjust the mask outward by 0.5% of the screen height. The value always includes a plus or minus sign. These values do not apply when using CinemaScope.

CurrentTopMaskAbsolute_(0.1percent)

CurrentBottomMaskAbsolute_(0.1percent) - Describes the position for the top and bottom masks in absolute terms, measured from the top and bottom of the screen, respectively. For example, a value of 0200, means that the corresponding mask should be located 20% from the top of the screen. These values do not apply when using CinemaScope.

CurrentTopMaskCalibrated_(0.1percent)

CurrentBottomMaskCalibrated_(0.1percent) - Contains masking positions calculated by the movie zone based on values entered in the Calibrate Masking page accessed through the **CalibrateMasking** command. These fields return the top and bottom trim values as a zero-padded, five-digit number between 00000 and 99999.

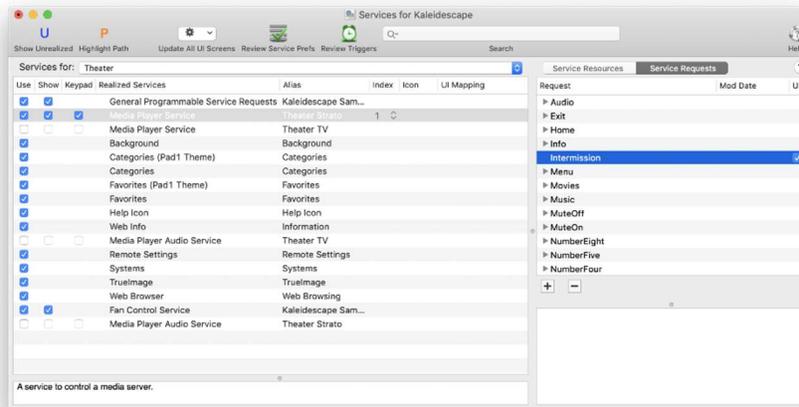
CurrentAspectRatio_Conservative - Has the same possible values as the **currentaspectratio** state value, but represents a more conservative estimate of the image aspect ratio. This value never goes too far into the actual picture. **Note:** This value is designed for 16:9 and 4:3 screens that have top and bottom masks without trim capability. This field should not be used on 2.35:1 screens with side masks, because it can cause the masks to move too far into the picture.

5 Custom Buttons

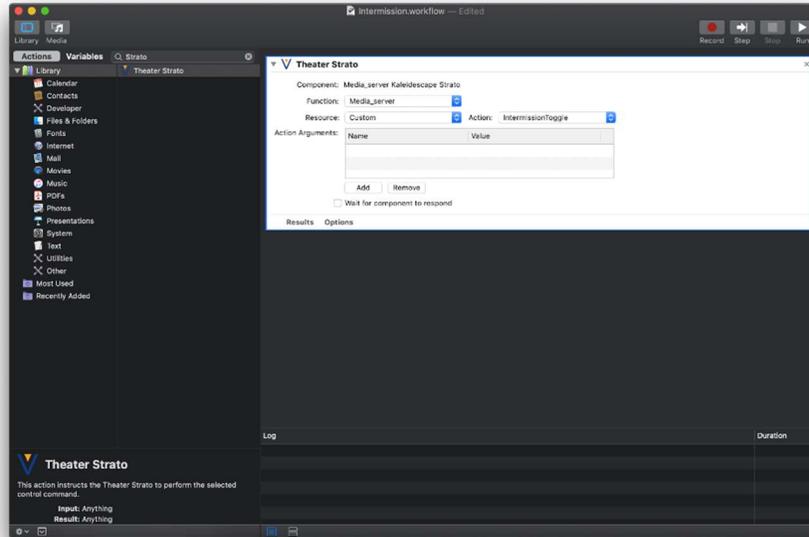
This section covers adding a custom button that is not on the UI by default. Intermission, for example, can be added as a custom button to the UI. Complete this section and then refer back to [3.2 Savant Pro App Deployment](#) to add the buttons to the UI.

5.1 Adding a Custom Button to the Pro App and Remotes

1. Open **View Services** for the Blueprint project and go to your zone that has the Strato player. If View Services is not available, you may have to generate services first.
2. Select your Strato player service and select **Service Requests** on the right side.
3. Add a new service request by clicking on the '+'. Name your new service request what you want it to display as on the Pro App and handheld remotes.
4. Check the UI checkbox. This will add this new button to the '+' list on the Kaleidescape control UI. The buttons will automatically be added to the handheld remotes.



5. Double-click the new service request to open it in Automator.
6. Search for your Strato player and double click on it or drag it to the workflow.
7. Select **'Media_server'** for the Function.
8. Select **'Custom'** for the resource
9. Select your new button from the **Action** dropdown list.



- Save your workflow and upload. You can now add the new button on the Pro App once your Strato service is activated. These custom buttons will show up on the handheld remotes once you select the service.

Current buttons available on Savant Pro App

Savant

Exit
Info
Setup
Menu
Subtitle
Home

Repeat
Audio
Movies
Random
Music

Kaleidescape Control Protocol Command

CANCEL:
DETAILS:
GO_SYSTEM_STATUS:
DVD_OR_KALEIDESCAPE_MENU:
SUBTITLES_NEXT:
KALEIDESCAPE_MENU_TOGGLE:

MUSIC_REPEAT_TOGGLE:
AUDIO_NEXT:
GO_MOVIE_LIST:
MUSIC_RANDOM_TOGGLE:
GO_MUSIC_LIST:

Other commands in the Savant profile that may be useful

Action name:

IntermissionToggle
IntermissionOn
IntermissionOff
AlphabetizeCoverArt
GoMovieCollection
GoMovieCoverArt
ShuffleCoverArt
SendCharacter
Store

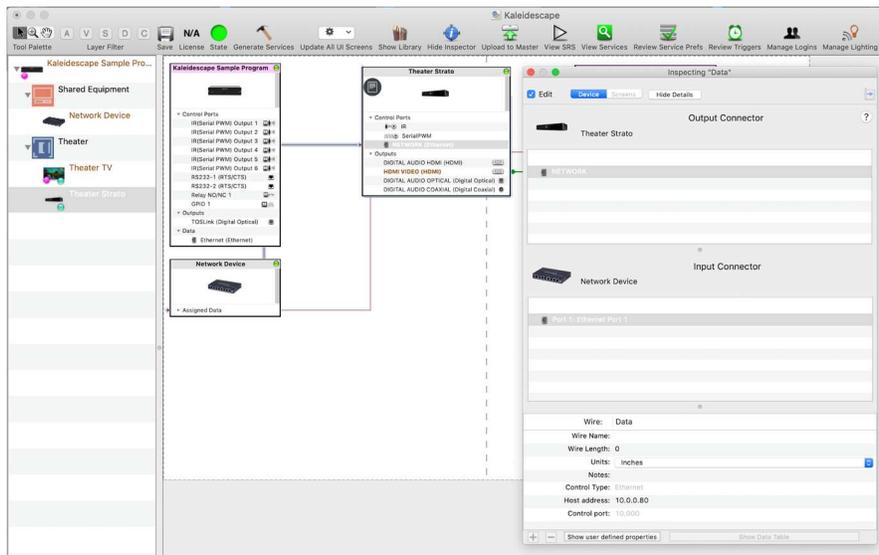
INTERMISSION_TOGGLE:
INTERMISSION_ON:
INTERMISSION_OFF:
ALPHABETIZE_COVER_ART:
GO_MOVIE_COLLECTIONS:
GO_MOVIE_COVERS:
SHUFFLE_COVER_ART:
KEYBOARD_CHARACTER: (from 'a' to 'Z' , use \: for '!')
GO_MOVIE_STORE

6 Troubleshooting

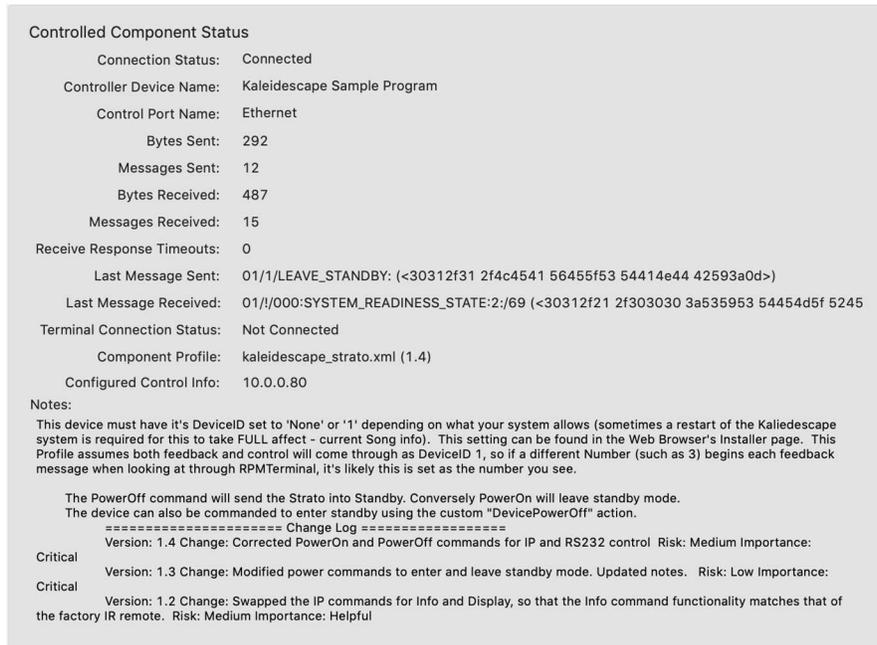
This section covers troubleshooting Kaleidescape player controls and triggers using System Monitor and RPMterminal.

6.1 No control over IP

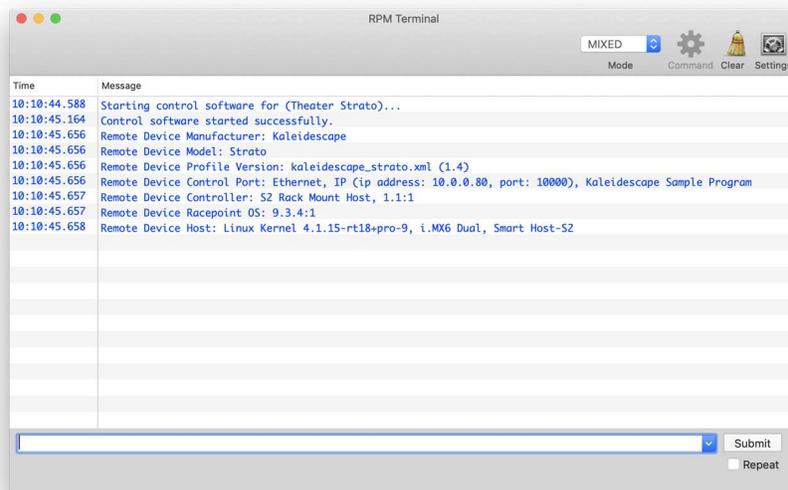
1. If you do not have control over IP, make sure the player IP is the same as in Blueprint. In Blueprint, highlight the **Network(ethernet)** connection and select **Show Inspector**. Check the Host address field for your IP address. If you do not see the IP information make sure to drag open the sections using the dots.



2. If the IP is correct, login to the player web interface and make sure the Device ID is set to **'None'** or **'1'**.
3. If this is all correct, the next step would be to check the RPMterminal and make sure you are sending and receiving data.
 - a. Open System Monitor.
 - b. Double click on your host or highlight and click Connect.
 - c. Go to **Component Status** on the left menu.
 - d. Select your player from the list of devices.



- e. On the right side, make sure **Connection Status** says **Connected**. If Connection Status is Connected but your controls still do not work, continue to the next step.
- f. While you still have your player selected click on **Launch Terminal** in the lower right to open the RPMterminal.



- g. Make sure the player is responding to Savant commands. Using RPMTerminal you can watch the commands being sent and the Kaleidescape player responding. You can also check the IP once the RPMTerminal opens. If you still do not have control, use the Kaleidescape App to control the player and watch the terminal for responding information. Make sure the player is using ID 1, "01/1/000:" If not make sure the player ID is correct.
- h. In RPMterminal the gray strings are commands being sent from Savant to the player and the orange strings are coming from the player to Savant.

7 Kaleidescape App Volume Control

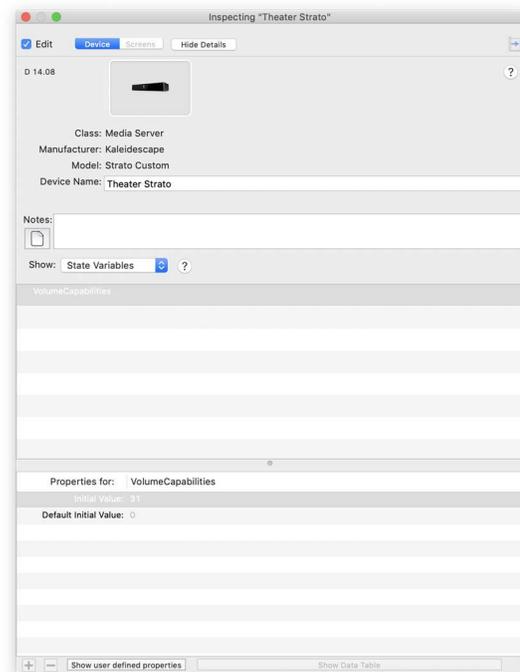
This section covers programming triggers and workflows to allow the use of the Kaleidescape App to control volume and mute of the Savant system. The audio controller for the Savant zone will require IP or RS-232 control due to the *SetVolume* command being used. If the system is not capable of IP or 232, a volume up and volume down command can be used for simple volume ramping of the audio controller. You will need to adjust the *VolumeCapabilities* state variable to accommodate the setup needed.

Each profile that is placed in Blueprint will create its own state variables to use for each player. Follow the example to create User State Variables and new triggers to allow for the Kaleidescape App to control the Savant zone volume and mute associated with each Strato player. Additional triggers will be added to send feedback to the Kaleidescape app for volume level feedback and mute status.

7.1 Player State Variables

By default this profile will not allow volume control from the Kaleidescape App. The default *VolumeCapabilities* state variable will need to be set for each player that will use Kaleidescape App volume controls.

1. Inspect the Strato player
2. Drop down the '**Show:**' menu and select **State Variables**.
3. Change the *initial value* of the **VolumeCapabilities** to **30**. See below for values.



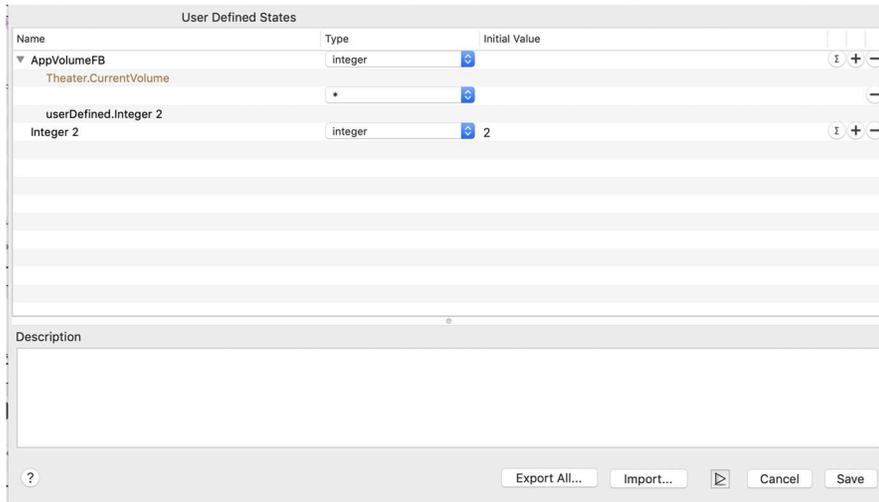
The sum of capabilities of the audio system:

- 0 No volume control or feedback
- 1 Volume up/down control
- 2 Mute Control
- 4 Volume level feedback
- 8 Mute feedback
- 16 Volume level set

For example, if the control system supports adjusting the volume level up and down, supports mute, and can report feedback for the volume level and mute, the flag would be set to: $1+2+4+8 = 15$

7.2 User State Variables

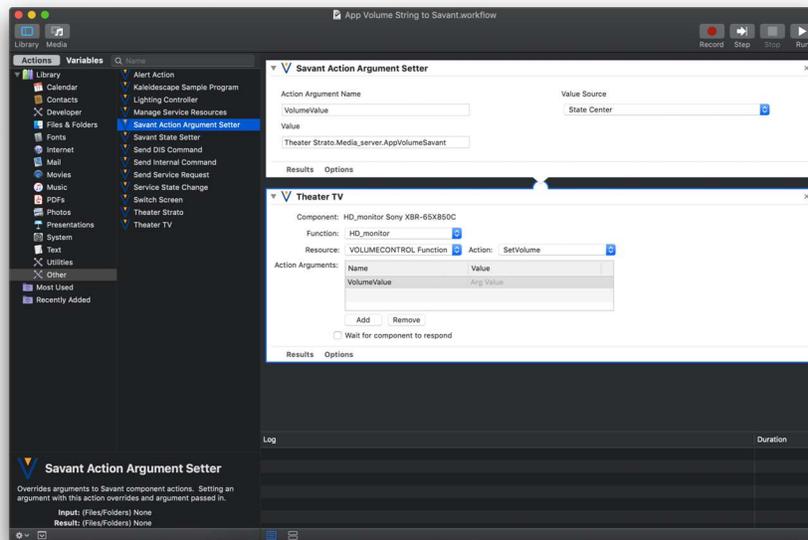
1. In Blueprint, go to **Tools > Review > User State Variables...**
2. Create 2 new states. **AppVolumeFB** and **integer 2**.



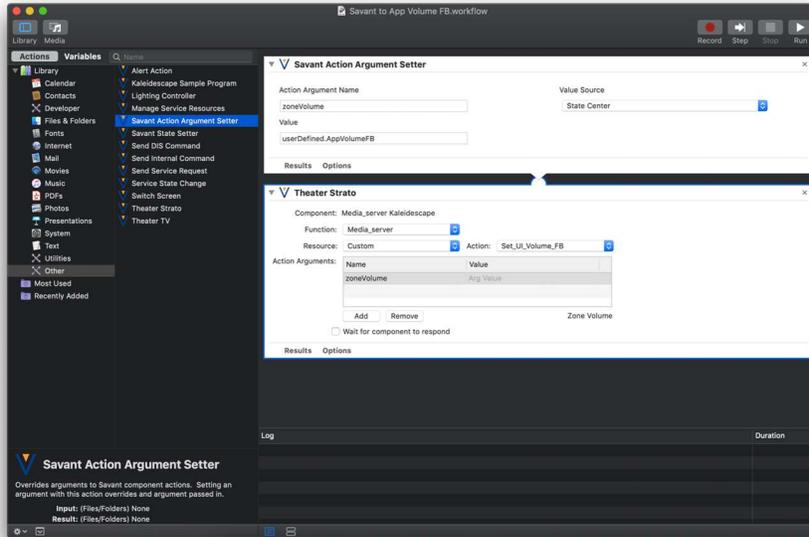
3. Integer 2 will be a **'Type'** of **integer** and **'Initial Value'** will be **2**.
4. AppVolumeFB will be a **'Type'** of **integer**. It will also need to be calculated, press the icon  to add new states.
5. Using the services icon  to find the state **CurrentVolume** for the Savant Strato zone. Drag it to the first state of the AppVolumeFB.
6. Drag the **Integer 2** variable from the list to the second state of the AppVolumeFB.
7. Change the calculation type to **'*'**.

7.3 Custom Workflows

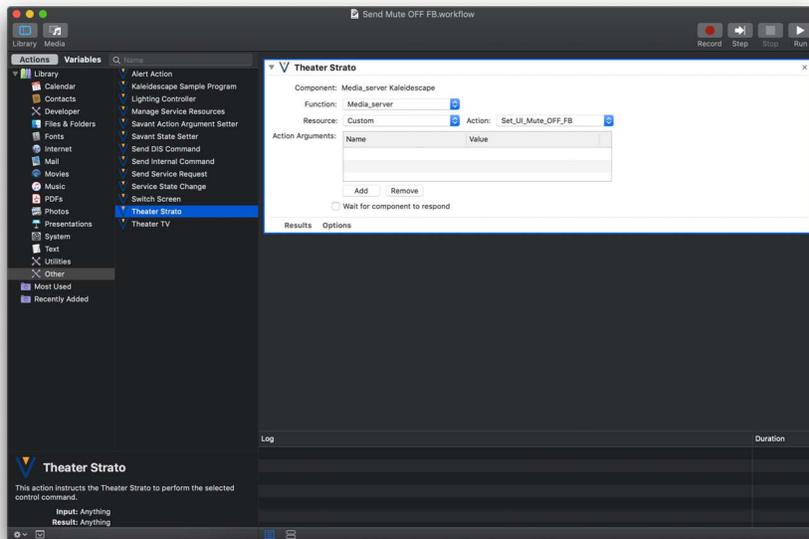
1. Create a new workflow. In this example it is named **App Volume String to Savant**. This workflow will only be needed if your VolumeCapabilities is set to use the *Volume Level Set*.
 - a. Add a **'Savant Action Argument Setter'**
 - i. *Action Argument Name* - **VolumeValue**
 - ii. *Value* - **Theater Strato.Media_server.AppVolumeSavant** (this will vary on how you named your Strato in Blueprint. This can be found in System States using System Monitor)
 - b. Add the device from Blueprint that is controlling the volume for the Savant zone.
 - i. *Function* - **HD_Monitor**
 - ii. *Resource* - **VOLUMECONTROL Function**
 - iii. *Action* - **SetVolume**
 - iv. Save and close this window



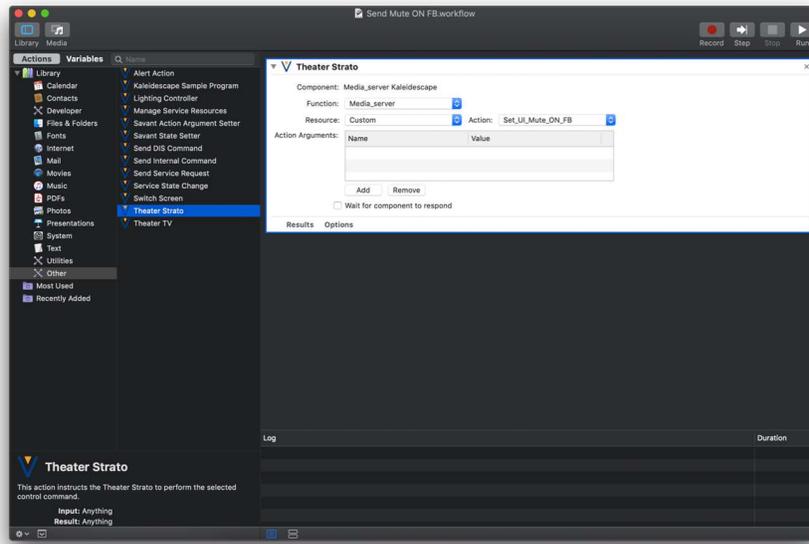
2. Create a new workflow. In this example it is named **Savant to App Volume FB**. This workflow will only be needed if your VolumeCapabilities is set to use the *Volume Level Feedback*.
 - a. Add a **'Savant Action Argument Setter'**
 - i. *Action Argument Name* - **ZoneVolume**
 - ii. *Value* - **userDefined.AppVolumeFB** (this is the user state variable)
 - b. Add your Strato device.
 - i. *Function* - **Media_server**
 - ii. *Resource* - **custom**
 - iii. *Action* - **Set_UI_Volume_FB**
 - iv. Save and close this window



3. Create a new workflow. In this example it is named **Send Mute OFF FB**. This workflow will only be needed if your VolumeCapabilities is set to use *Mute Feedback*.
 - a. Add the Strato device.
 - i. *Function* - **Media_server**
 - ii. *Resource* - **Custom**
 - iii. *Action* - **Set_UI_Mute_Off_FB**
 - iv. Save and close this window

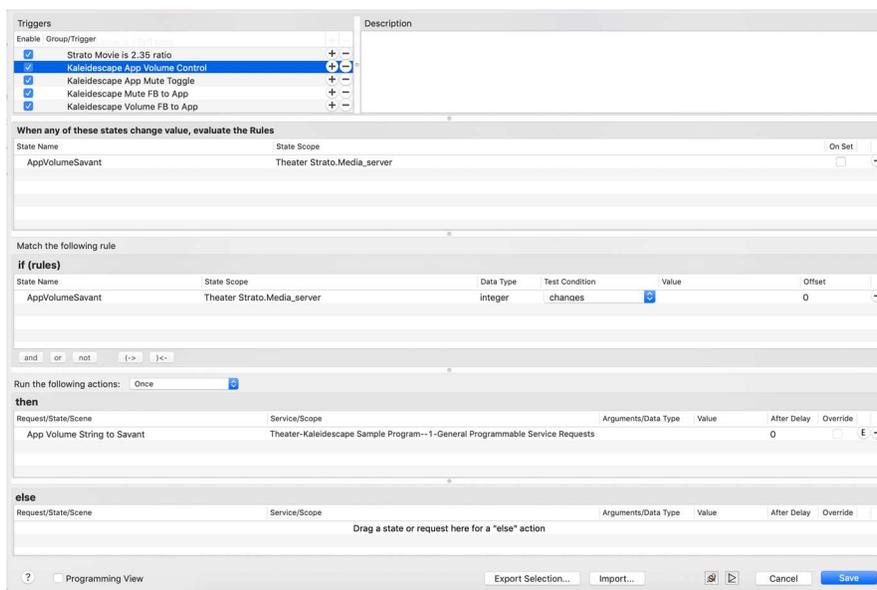


4. Create a new workflow. In this example it is named **Send Mute ON FB**. This workflow will only be needed if your VolumeCapabilities is set to use *Mute Feedback*.
 - a. Add the Strato device.
 - i. *Function* - **Media_server**
 - ii. *Resource* - **Custom**
 - iii. *Action* - **Set_UI_Mute_On_FB**
 - iv. Save and close this window



7.4 Triggers

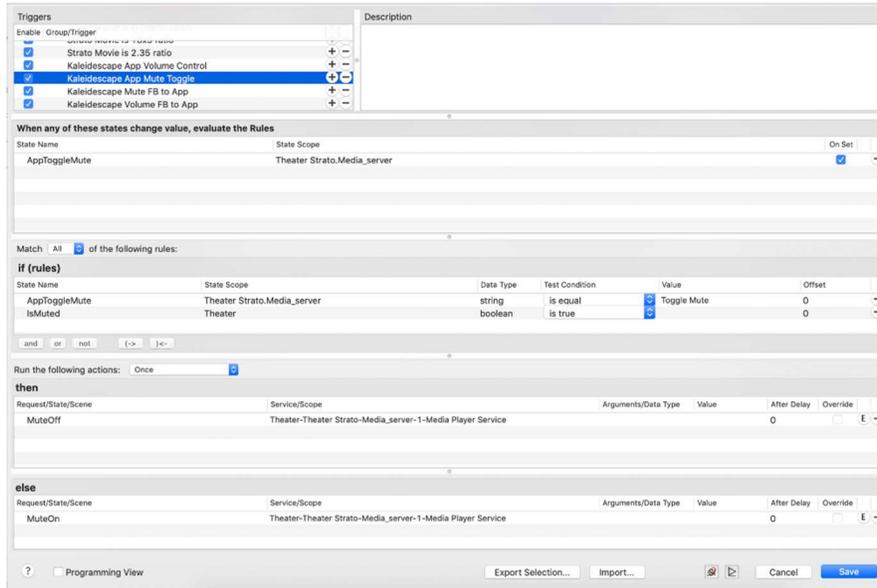
1. Create a new trigger. In this example it is named **Kaleidescape App Volume Control**. This trigger will monitor the volume from the Kaleidescape App. Once it changes state Savant will use the current volume value and send it to your audio controller for the Savant zone. This trigger will only be needed if your VolumeCapabilities is set to use *Volume Level Set*.
 - a. **Evaluate - Theater Strato.Media_server.AppVolumeSavant**
 - b. **If - Theater Strato.Media_server.AppVolumeSavant / Test Condition - changes**
 - c. **Then - App Volume String to Savant** (custom workflow)



2. Create a new trigger. In this example it is named **Kaleidescape App Mute Toggle**. This trigger will monitor the mute toggle command from the Kaleidescape App. Once it changes

state Savant will use the current mute status to mute or unmute the audio controller. This trigger will only be needed if your VolumeCapabilities is set to use *Mute Control*.

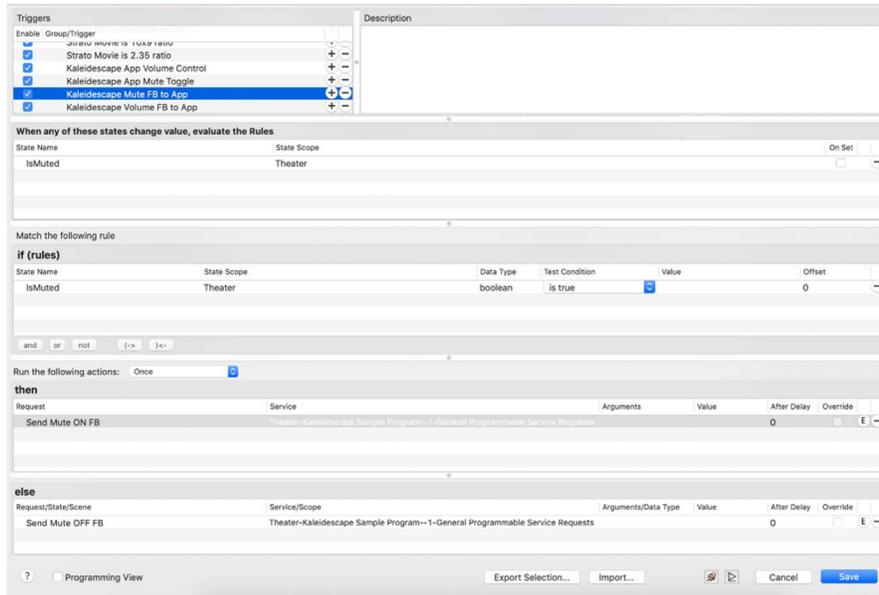
- a. *Evaluate* - **Theater Strato.Media_server.AppToggleMute** / *Check* - **On Set**
- b. *If* - **Theater Strato.Media_server.AppToggleMute** / *Test Condition* - **is equal / Value Toggle Mute & Theater IsMuted** / *test Condition* - **is true**
- c. *Then* - **MuteOff** (direct audio controller mute off workflow)
- d. *Else* - **MuteOn** (direct audio controller mute on workflow)



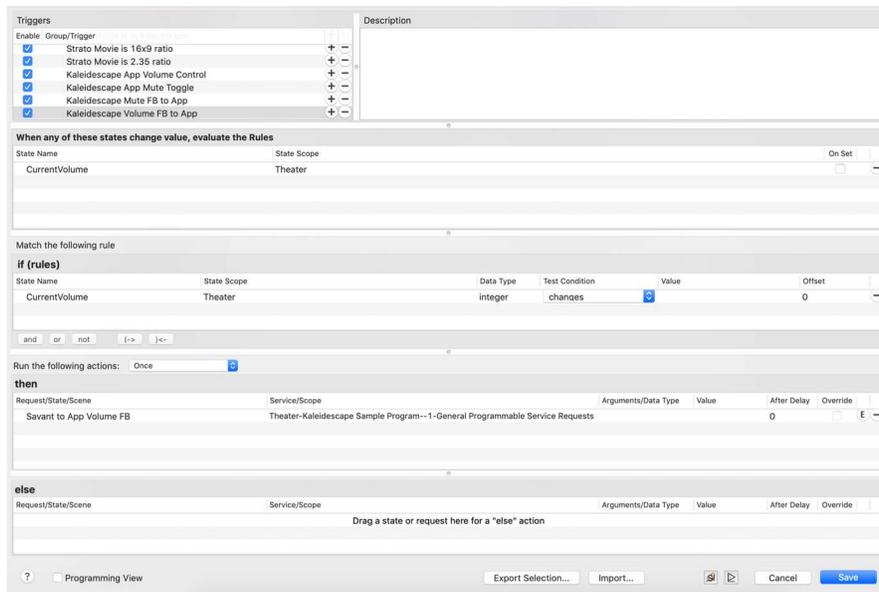
3. Create a new trigger. In this example it is named **Kaleidescape Mute FB to App**. This trigger will monitor the mute status of the Savant zone. Once it changes state Savant will use the current mute status to send the Kaleidescape App mute feedback. This trigger will only be needed if your VolumeCapabilities is set to use *Mute Feedback*.

- a. *Evaluate* - **Theater IsMuted**
- b. *If* - **Theater IsMuted** / *Test Condition* - **is true**
- c. *Then* - **Send Mute ON FB** (custom workflow)

d. **Else - Send Mute OFF FB** (custom workflow)



4. Create a new trigger. In this example it is named **Kaleidescape Volume FB to App**. This trigger will monitor the volume status of the Savant zone. Once it changes state Savant will use the current volume status to send the Kaleidescape App volume feedback. This trigger will only be needed if your VolumeCapabilities is set to use *Volume Level Feedback*.
 - a. **Evaluate - Theater.CurrentVolume**
 - b. **If - Theater CurrentVolume / Test Condition - changes**
 - c. **Then - Savant to App Volume FB** (custom workflow)



5. Create a new trigger. In this example it is named **Kaleidescape Volume Up**. This trigger will monitor the volume up command from the Kaleidescape App. This trigger will only be needed if your VolumeCapabilities is set to use *Volume up/down control*.
 - a. *Evaluate* - **Theater Strato.Media_server.AppVolumeUp** / *Check* - **On Set**
 - b. *If* - **Theater Strato.Media_server.AppVolumeUp** / *Test Condition* - **is equal / Value Up**
 - c. *Then* - **VolumeUp** (direct audio controller volume increase workflow)

The screenshot shows the configuration for the 'Kaleidescape Volume Up' trigger. In the 'Triggers' list, 'Kaleidescape Volume Up' is selected. Below, the 'When any of these states change value, evaluate the Rules' section shows a rule for 'AppVolumeUp' with state scope 'Theater Strato.Media_server' and 'On Set' checked. The 'Match the following rule' section shows an 'if (rules)' condition where 'AppVolumeUp' is checked and the test condition is 'is equal' with a value of 'Up'. The 'Run the following actions' section shows a 'then' action 'VolumeUp' with a service scope of 'Theater-Theater Strato-Media_server-1-Media Player Service' and 'After De...' set to 0.

6. Create a new trigger. In this example it is named **Kaleidescape Volume Down**. This trigger will monitor the volume down command from the Kaleidescape App. This trigger will only be needed if your VolumeCapabilities is set to use *Volume up/down control*.
 - a. *Evaluate* - **Theater Strato.Media_server.AppVolumeDown** / *Check* - **On Set**
 - b. *If* - **Theater Strato.Media_server.AppVolumeDown** / *Test Condition* - **is equal / Value Down**
 - c. *Then* - **VolumeDown** (direct audio controller volume decrease workflow)

The screenshot shows the configuration for the 'Kaleidescape Volume Down' trigger. In the 'Triggers' list, 'Kaleidescape Volume Down' is selected. Below, the 'When any of these states change value, evaluate the Rules' section shows a rule for 'AppVolumeDown' with state scope 'Theater Strato.Media_server' and 'On Set' checked. The 'Match the following rule' section shows an 'if (rules)' condition where 'AppVolumeDown' is checked and the test condition is 'is equal' with a value of 'Down'. The 'Run the following actions' section shows a 'then' action 'VolumeDown' with a service scope of 'Theater-Theater Strato-Media_server-1-Media Player Service' and 'After De...' set to 0.

8 Appendix

Savant Command - *resource_component_actions*

Kaleidescape Control Command

Home	01/1/KALEIDESCAPE_MENU_TOGGLE:
Movies	01/1/GO_MOVIE_LIST:
Music	01/1/GO_MUSIC_LIST:
Audio	01/1/AUDIO_NEXT:
Menu	01/1/DVD_OR_KALEIDESCAPE_MENU:
Setup	01/1/GO_SYSTEM_STATUS:
Info	01/1/STATUS_AND_SETTINGS:
Subtitle	01/1/SUBTITLES_NEXT:
PageUp	01/1/PAGE_UP:
PageDown	01/1/PAGE_DOWN:
ScanUp	01/1/SCAN_FORWARD:
ScanDown	01/1/SCAN_REVERSE:
SkipUp	01/1/NEXT:
SkipDown	01/1/PREVIOUS:
Play	01/1/PLAY:
Pause	01/1/PAUSE:
Stop	01/1/STOP:
NumberZero	01/1/KEYBOARD_CHARACTER:0:
NumberOne	01/1/KEYBOARD_CHARACTER:1:
NumberTwo	01/1/KEYBOARD_CHARACTER:2:
NumberThree	01/1/KEYBOARD_CHARACTER:3:
NumberFour	01/1/KEYBOARD_CHARACTER:4:
NumberFive	01/1/KEYBOARD_CHARACTER:5:
NumberSix	01/1/KEYBOARD_CHARACTER:6:

NumberSeven	01/1/KEYBOARD_CHARACTER:7:
NumberEight	01/1/KEYBOARD_CHARACTER:8:
NumberNine	01/1/KEYBOARD_CHARACTER:9:
Repeat	01/1/MUSIC_REPEAT_TOGGLE:
Random	01/1/MUSIC_RANDOM_TOGGLE:
OSDCursorUp	01/1/UP:
OSDCursonDown	01/1/DOWN:
OSDCursorLeft	01/1/LEFT:
OSDCursorRight	01/1/RIGHT:
Select	01/1/SELECT:
Exit	01/1/CANCEL:

Savant Command - *custom_component_actions*

Kaleidescape Control Command

DevicePowerOff	01/1/ENTER_STANDBY:
DevicePowerOn	01/1/LEAVE_STANDBY:
PowerToggle	01/1/PAUSE:
PauseOn	01/1/PAUSE_ON:
PauseOff	01/1/PAUSE_OFF:
KaleidescapeMenuOn	01/1/KALEIDESCAPE_MENU_ON:
KaleidescapeMenuOff	01/1/KALEIDESCAPE_MENU_OFF:
DVDMenu	01/1/DVD_MENU:
DVDTopMenu	01/1/DVD_TOP_MENU:
DVDResume	01/1/DVD_RESUME:
NextAngle	01/1/ANGLE_NEXT:
Aspect	01/1/GET_ASPECT_RATIO:
Replay	01/1/REPLAY:
ShuffleCoverArt	01/1/SHUFFLE_COVER_ART:

ParentalControlSafeLevel	01/1/SAFE_LEVEL:
ParentalControlDefaultLevel	01/1/DEFAULT_LEVEL:
SetFavoriteSceneStart	01/1/SET_FAVORITE_SCENE_START:
SetFavoriteSceneEnd	01/1/SET_FAVORITE_SCENE_END:
GoMovieCoverArt	01/1/GO_MOVIE_COVERS:
GoMovieCollection	01/1/GO_MOVIE_COLLECTIONS:
GoParentalControl	01/1/GO_PARENTAL_CONTROL:
IntermissionToggle	01/1/INTERMISSION_TOGGLE:
IntermissionOn	01/1/INTERMISSION_ON:
IntermissionOff	01/1/INTERMISSION_OFF:
GoScreenSaver	01/1/GO_SCREEN_SAVER:
StopScreenSaver	01/1/STOP_SCREEN_SAVER:
Display	01/1/DETAILS:
DisplayOff	01/1/DETAILS:
DisplayToggle	01/1/DETAILS:
GoMusicCovers	01/1/GO_MUSIC_COVERS:
GoMusicCollections	01/1/GO_MUSIC_COLLECTIONS:
GoNowPlaying	01/1/GO_NOW_PLAYING:
Req-ProtocolVersion	01/1/GET_PROTOCOL:
Req-Time	01/1/GET_TIME:
Req-NetworkSetting	01/1/GET_NETWORK_SETTINGS:
SetNetworkSetting	01/1/SET_NETWORK_SETTINGS:
Req-SystemCapabilities	01/1/GET_SYSTEM_CAPABILITIES:
Req-AvailableDevices	01/1/GET_AVAILABLE_DEVICES:
Req-DeviceInfo	01/1/GET_DEVICE_INFO:
Req-EnableEvents	01/1/ENABLE_EVENTS:
Req-DisableEvents	01/1/DISABLE_EVENTS:

Req-FriendlyName	01/1/GET_FRIENDLY_NAME:
SetFriendlyName	01/1/SET_FRIENDLY_NAME:
OSDCursorUp_Release	01/1/UP_RELEASE:
OSDCursorDown_Release	01/1/DOWN_RELEASE:
OSDCursorLeft_Release	01/1/LEFT_RELEASE:
OSDCursorRight_Release	01/1/RIGHT_RELEASE:
OSDCursorUp_Press	01/1/UP_PRESS:
OSDCursorDown_Press	01/1/DOWN_PRESS:
OSDCursorLeft_Press	01/1/LEFT_PRESS:
OSDCursorRight_Press	01/1/RIGHT_PRESS:
PageUp_Release	01/1/PAGE_UP_RELEASE:
PageDown_Release	01/1/PAGE_DOWN_RELEASE:
PageUp_Press	01/1/PAGE_UP_PRESS:
PageDown_Press	01/1/PAGE_DOWN_PRESS:
SetPosition	01/1/POSITION_SELECT:
SendCharacter	01/1/KEYBOARD_CHARACTER:
Backspace	01/1/BACKSPACE:
Req_FilterList	01/1/FILTER_LIST:
AlphabetizeCoverArt	01/1/ALPHABETIZE_COVER_ART:
GoParentalControl	01/1/GO_PARENTAL_CONTROL:
PlayScript	01/1/PLAY_SCRIPT:
Store	01/1/GO_MOVIE_STORE:
CalibrateMasking	01/1/GO_CALIBRATE_MASKING:
CalibrateMaskingOverscan	01/1/GO_CALIBRATE_MASKING_OVERSCAN:
SetScreenMask	01/1/SET_SCREEN_MASK:
PreviousAngle	01/1/ANGLE_PREVIOUS:
StatusAndSettings	01/1/STATUS_AND_SETTINGS:

ShowNavigationOverlay	01/1/SHOW_NAVIGATION_OVERLAY:
StartChapterEntry	01/1/START_CHAPTER_ENTRY:
StartDVDTitleEntry	01/1/START_DVD_TITLE_ENTRY:
StartSendNumberToDVDEntry	01/1/START_SEND_NUMBER_TO_DVD_ENTRY:
StopOrCancel	01/1/STOP_OR_CANCEL:
PageUpOrPrevious	01/1/PAGE_UP_OR_PREVIOUS:
PageUpOrPreviousPress	01/1/PAGE_UP_OR_PREVIOUS_PRESS:
PageUpOrPreviousRelease	01/1/PAGE_UP_OR_PREVIOUS_RELEASE:
PageDownOrNext	01/1/PAGE_DOWN_OR_NEXT:
PageDownOrNextPress	01/1/PAGE_DOWN_OR_NEXT_PRESS:
PageDownOrNextRelease	01/1/PAGE_DOWN_OR_NEXT_RELEASE:
PageUpOrNext	01/1/PAGE_UP_OR_NEXT:
PageUpOrNextPress	01/1/PAGE_UP_OR_NEXT_PRESS:
PageUpOrNextRelease	01/1/PAGE_UP_OR_NEXT_RELEASE:
PageDownOrPrevious	01/1/PAGE_DOWN_OR_PREVIOUS:
PageDownOrPreviousPress	01/1/PAGE_DOWN_OR_PREVIOUS_PRESS:
PageDownOrPreviousRelease	01/1/PAGE_DOWN_OR_PREVIOUS_RELEASE:
GetCameraAngle	01/1/GET_CAMERA_ANGLE:
GetMovieLocation	01/1/GET_MOVIE_LOCATION:
GetPlayStatus	01/1/GET_PLAY_STATUS:
GetPlayingTitleName	01/1/GET_PLAYING_TITLE_NAME:
SetStatusCuePeriod	01/1/SET_STATUS_CUE_PERIOD:
GetUIStatus	01/1/GET_UI_STATE:
GetHighlightedSelection	01/1/GET_HIGHLIGHTED_SELECTION:
GetUserInput	01/1/GET_USER_INPUT:
GetScreenMask	01/1/GET_SCREEN_MASK:
GetVideoMode	01/1/GET_VIDEO_MODE:

GetContentDetails	01/1/GET_CONTENT_DETAILS:
Set_UI_Volume_FB	01/1/SEND_EVENT:VOLUME_LEVEL=
Set_UI_Mute_ON_FB	01/1/SEND_EVENT:MUTE_ON_FB:
Set_UI_Mute_OFF_FB	01/1/SEND_EVENT:MUTE_OFF_FB:
GetCinemascopeMask	01/1/GET_CINEMASCAPE_MASK:
GetCinemascopeMode	01/1/GET_CINEMASCAPE_MODE:
SetCinemascopeMode	01/1/SET_CINEMASCAPE_MODE:
CHILD_OSDCursorDown_Press	01/1/CHILD_DOWN_PRESS:
CHILD_OSDCursorDown_Release	01/1/CHILD_DOWN_RELEASE:
CHILD_OSDCursorDown	01/1/CHILD_DOWN:
CHILD_OSDCursorUp_Press	01/1/CHILD_UP_PRESS:
CHILD_OSDCursorUp_Release	01/1/CHILD_UP_RELEASE:
CHILD_OSDCursorUp	01/1/CHILD_UP:
CHILD_OSDCursorLeft_Press	01/1/CHILD_LEFT_PRESS:
CHILD_OSDCursorLeft_Release	01/1/CHILD_LEFT_RELEASE:
CHILD_OSDCursorLeft	01/1/CHILD_LEFT:
CHILD_OSDCursorRight_Press	01/1/CHILD_RIGHT_PRESS:
CHILD_OSDCursorRight_Release	01/1/CHILD_RIGHT_RELEASE:
CHILD_OSDCursorRight	01/1/CHILD_RIGHT:
CHILD_OSDCursorSelect	01/1/CHILD_SELECT:
CHILD_ShuffleCoverArt	01/1/CHILD_SHUFFLE_COVER_ART:
CHILD_Play	01/1/CHILD_PLAY:
CHILD_Pause	01/1/CHILD_PAUSE:
CHILD_Stop	01/1/CHILD_STOP:
CHILD_GetChildModeState	01/1/GET_CHILD_MODE_STATE:
CHILD_EnterChildMode	01/1/ENTER_CHILD_MODE:
CHILD_LeaveChildMode	01/1/LEAVE_CHILD_MODE:

