August 2025

Immersive Workflow Guide

Blackmagicdesign

DaVinci Resolve

Leading the Creative Revolution

Contents

Introduction

This manual serves as a practical guide for filmmakers exploring the frontier of immersive content creation in the Blackmagic URSA Cine Immersive camera and Apple Vision Pro ecosystem. With stereoscopic 8K video capture, spatial audio integration, and advanced post-production workflows in DaVinci Resolve Studio, immersive filmmaking unlocks a new chapter in visual storytelling.

Beyond technical specifications and software settings, this guide encourages creative considerations essential for producing compelling immersive video content. It offers tips for optimizing viewer comfort, adapting familiar editing workflows to immersive timelines, and delivering technically compliant content. Whether you're preparing your first stereoscopic shoot or using it as a repeat reference for project setup, this manual ensures you're equipped to craft deeply engaging immersive experiences.

Blackmagic URSA Cine Immersive

Blackmagic URSA Cine Immersive

The Blackmagic URSA Cine Immersive is the world's first commercial camera system engineered specifically for capturing immersive content for the Apple Vision Pro. It features dual 8K sensors that capture 3D video with a 180-degree field of view for true stereoscopic depth. With 16 stops of dynamic range and support for 90 fps recording, it enables filmmakers to create ultra-realistic, high-fidelity cinematic experiences. Recordings are made in the stereoscopic Blackmagic RAW Immersive format, allowing for accurate color management and HDR post-production in DaVinci Resolve Studio.

Apple Vision Pro

Apple Vision Pro

The Apple Vision Pro is a mixed-reality headset that blends virtual and augmented reality with real-world interaction for a seamless experience.

Immersive Video and Spatial Audio

Immersive Video is a next-generation cinematic format designed to envelop viewers in lifelike, three-dimensional experiences. Unlike traditional 360° or VR formats, Immersive Video focuses on realism and depth captured using dual lenses and sensors for each eye with a fidelity that approaches human 20/20 vision. Specialized encoding formats, MV-HEVC and AIVU, ensure smooth playback and hyper-realistic motion. Combined with Spatial Audio, this format creates the sensation of being physically present in the scene.

Spatial Audio is a transformative sound technology that creates a three-dimensional listening experience. It uses advanced algorithms and object-based mixing to position individual sounds in a virtual space. This allows for precise placement of voices, instruments, and effects, enhancing immersion whether watching a movie, listening to music, or playing a game. With head tracking, the soundscape dynamically adjusts to the viewer's movements, anchoring audio to the active environment for a realistic experience.

DaVinci Resolve Studio

DaVinci Resolve Studio is a comprehensive post production software that seamlessly integrates editing, color correction, visual effects, motion graphics, and audio post-production. It supports workflows tailored to the Blackmagic RAW Immersive format's unique characteristics – including ultra high resolution stereoscopic video, 90 fps frame rates, and Spatial Audio mixing. DaVinci Resolve Studio is fully equipped to handle the production, monitoring, and delivery of Apple Immersive Video, making it a powerful tool for immersive content creators.

Workstation Recommendations

Hardware recommendations

- Mac Studio M3 Ultra Desktop
- High refresh rate monitor (minimum 60Hz)

Minimum system requirements

- macOS 15 or later
- 64 GB of system memory
- For monitoring, Blackmagic Design Desktop Video 12.9 or later
- Apple Silicon based computer or GPU which supports Metal

Immersive monitoring requirements

- Apple Vision Pro
- High-speed wireless network (Wi-Fi 5 minimum; Wi-Fi 6 preferred)

Project Setup

Before ingesting Blackmagic URSA Cine Immersive raw media, it is good practice to go through DaVinci Resolve's project settings to ensure the media is correctly debayered and processed for optimal visual output and ease of editing.

Master Settings

It is crucial to indicate that you will be creating content for the Apple Vision Pro when setting up a project. This will ensure that all the correct settings and workflow options are visible to you as you prepare, assemble and deliver your project.

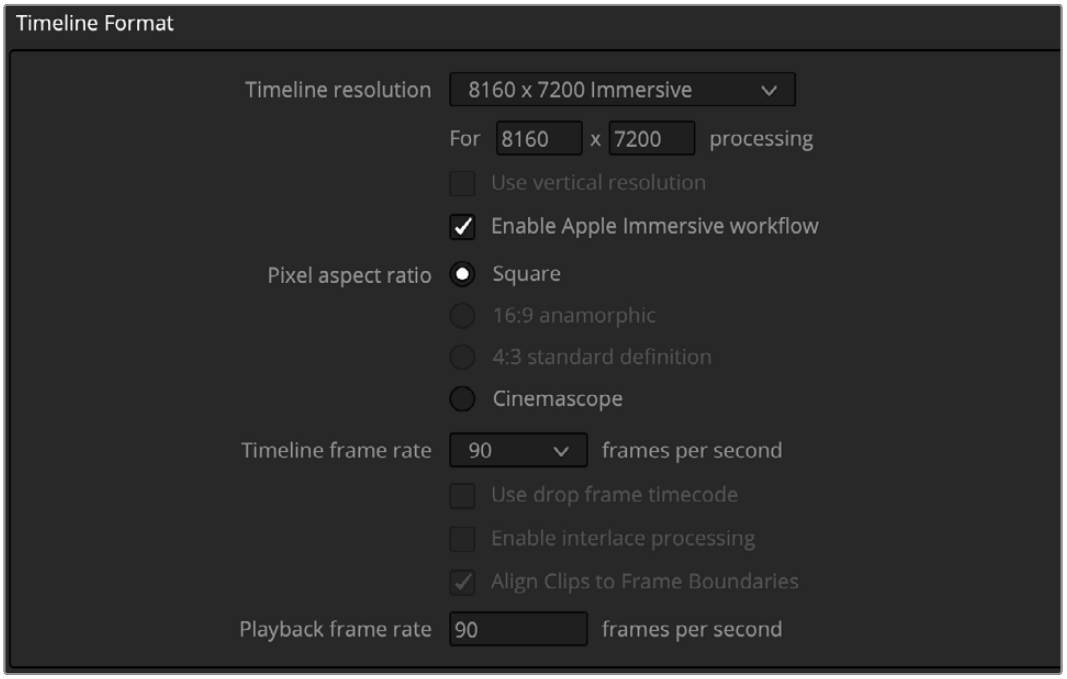
1. Launch DaVinci Resolve Studio.
2. In the lower right corner of the Project Manager window, click New Project.
3. Enter a project name, set the Media Location (if needed) and click Create.
4. Open the Project Settings by clicking the gear icon in the lower right corner of DaVinci Resolve Studio.

   First, you will set the resolution and frame rate of the project.
5. Select Master Settings in the sidebar.
6. In the Timeline Format section, select 'Enable Apple Immersive workflow'.

   This will configure multiple DaVinci Resolve settings to best accommodate immersive workflows. When creating an immersive project, ensure you always select this setting first.
7. Change the 'Timeline resolution' to '8160 x 7200 Immersive' to process the clips in their native sensor resolution.

   If working with limited processing power, choose '4080 x 3600 Immersive' to half the resolution for faster playback.
8. The 'Timeline frame rate' should be set to 90 frames per second. This parameter is automatically configured when you enable Apple Immersive workflows.
9. Set the 'Playback frame rate' to 90 frames per second for the most accurate reproduction of your Immersive video content.

   If working with limited processing power (or a monitor with a lower refresh rate), lower the frame rate for more stable playback. It is recommended you choose a frame rate that is a fraction of the native frame rate, such as 30fps or 60fps.



Video Monitoring

If you are using an external monitor (via a capture/playback device like Decklink or UltraStudio), configure the video monitoring settings based on your monitor's specifications.

1. In the Video Monitoring section, set the Video Resolution to match your monitor resolution. E.g. 1920 x 1080 HD or 3840 x 2160 Ultra HD.

   Though the native frame rate of immersive video is 90 fps, it is not a frame rate most broadcast monitors support.

2. Set the Format to match your monitor frame rate. E.g. 30p.

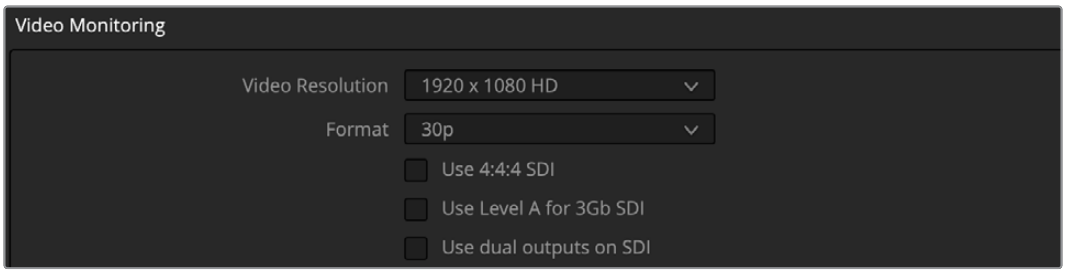


Image Scaling

The Image Scaling settings will ensure that the resolution and aspect ratio of the immersive media is correctly displayed in the viewer. These parameters would have automatically been configured after you enabled the Apple Immersive workflow option in the Master Settings, but you should still verify that they are correct.

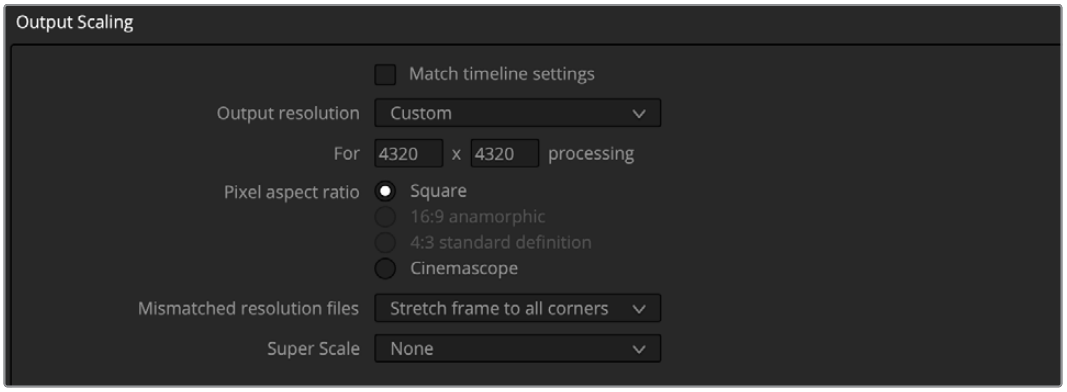
1. Select Image Scaling in the Project Settings sidebar.
2. In the Output Scaling section, ensure 'Match timeline settings' is deselected.

3 Ensure the ‘Output resolution’ is Custom and set to **4320 x 4320**.

Half the resolution (**2160 x 2160**) for smaller displays and/or faster playback.

These square aspect ratio resolutions are required for playing back Apple Immersive video content within DaVinci Resolve Studio and for streaming to the Apple Vision Pro (via wifi or developer strap).

4 Ensure ‘Mismatched resolution files’ is set to ‘Stretch frame to all corners’.



Color Management

Due to the need for custom control over the maximum output nit luminance, color management is encouraged with the use of Color Space Transform (CST) nodes in the color page. The goal of the color management settings in the Project Settings is to specify the timeline (or ‘working’) color space of the color page and to set the color space metadata that will be associated with the rendered file.

1 Select Color Management in the Project Settings sidebar.

2 Leave ‘Color science’ as DaVinci YRGB.

3 Set the ‘Timeline color space’ as DaVinciWG/Intermediate.

This is a future-proof wide gamut color space that supports HDR workflows, making it ideal for Blackmagic URSA Cine Immersive media and Apple Vision Pro deliverables.

4 Set the ‘Output color space’ as ‘P3-D65 ST2084 1000 nit.’

This color space features the appropriate gamut for monitoring the DaVinci Resolve viewers on a Mac Studio Desktop, which is typically Display P3. If you are using a grading monitor or a different Apple ICC profile, the output color space should be changed to match the color standard of your display.

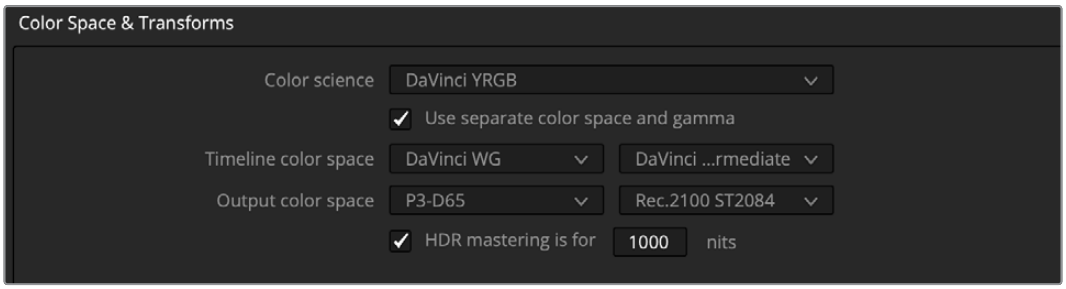
However, you will need to change the gamma (transfer function) to correctly map to the ICC profile.

5 Click ‘Use separate color space and gamma’ above the color space settings.

This splits the color space and gamma dropdown fields.

6 Set the output gamma (in the right column) as ‘Rec.2100 ST2084.’

7 Select ‘HDR mastering is for 1000 nits.’



To spend less time setting up your next Apple Immersive project, you can save the current settings as a preset.

8 Go to the three-dot options menu in the upper right corner of the Project Settings window and choose ‘Save Current Settings as Preset’.

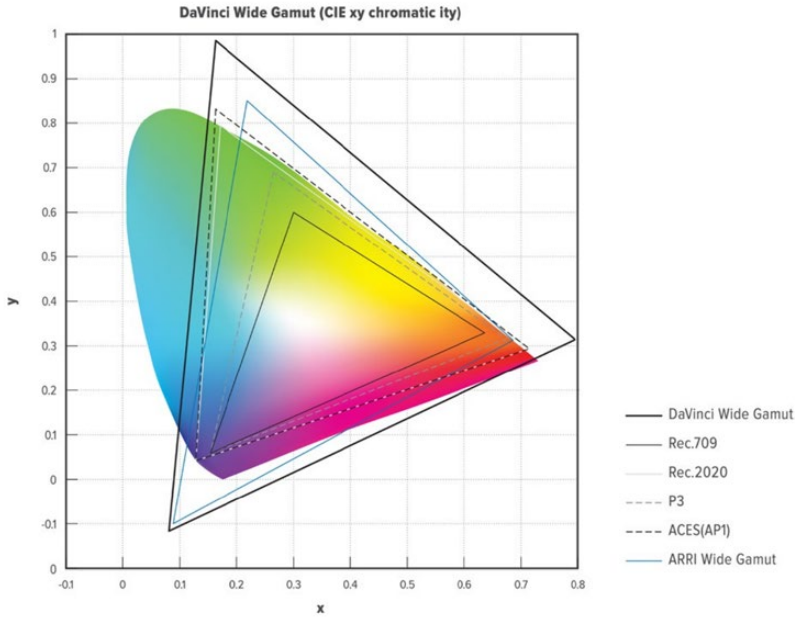
9 Name the preset and click Ok.

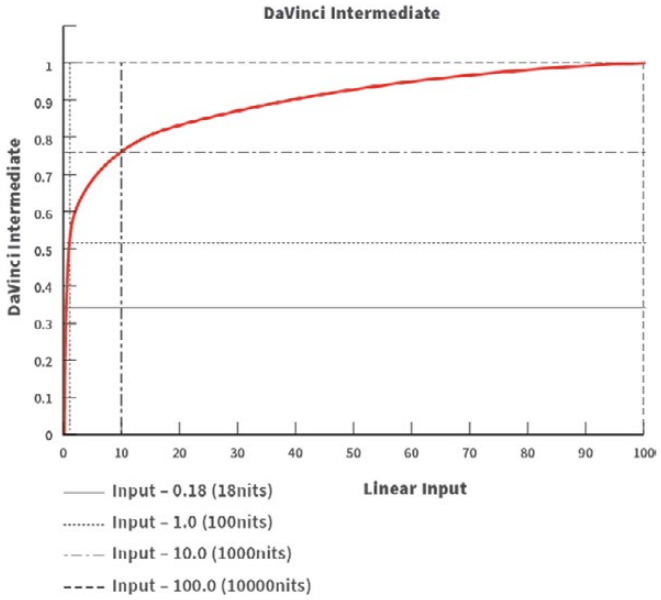
To quickly configure the project settings next time you work with Immersive media, open the Project Settings window, click the options menu and choose the listed preset.

10 Click Save to close the Project Settings.

About DaVinci Wide Gamut

DaVinci Wide Gamut is Blackmagic Design’s advanced internal color space developed for DaVinci Resolve, offering an exceptionally broad and precise environment for color grading and image processing. Paired with the DaVinci Intermediate gamma, it forms a unified working space that simplifies color management across mixed camera formats, ensuring consistent tone and saturation mapping from input to output. Ideal for working with HDR and Immersive media, this wide gamut allows colorists to make nuanced adjustments without risk of banding or tonal clipping when mastering for high-fidelity delivery formats.





Apple Immersive Processing

In the Project Settings sidebar, the Apple Immersive section allows you to set custom ILPD (Immersive Lens Profile Data) to modify how the immersive content is calibrated and projected.

1. Select Apple Immersive in the Project Settings sidebar.
2. Click Open Calibration Folder.
3. Add or update a custom ILPD file.
4. Click Update Lists to refresh the ILPD content accessible in the Media Pool.
5. Click Save to close the Project Settings.
6. In the Media Pool, right-click an immersive clip.
7. In the contextual menu, navigate to Apple Immersive Calibration and choose either the source ILPD (‘No Apple Immersive Calibration File Selected’), the custom ILPD, or the gap ILPD.

By default, the custom ILPD Calibration folder is located in the ProIM folder, alongside the default LUT folder.

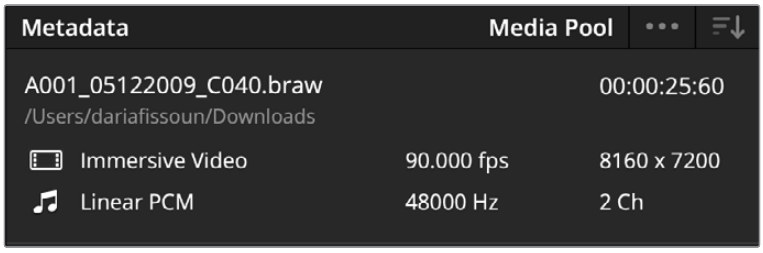
Ingesting

When ingesting raw immersive content, both eyes and all associated immersive metadata travel with the imported .braw clip.

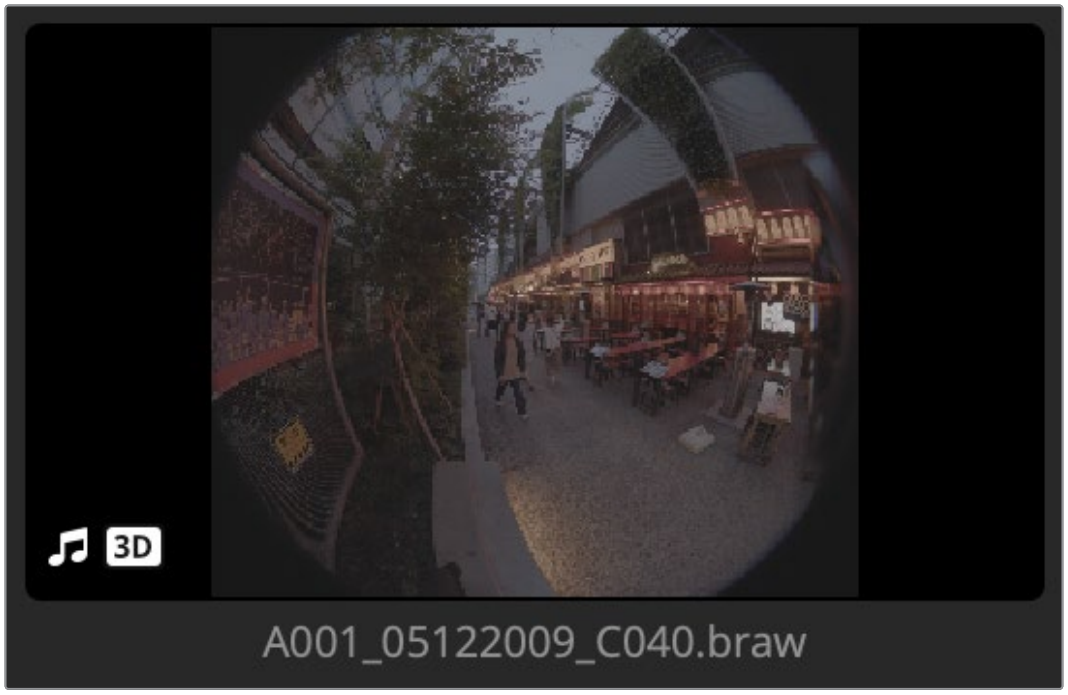
1. Go to the media page.
2. In the Media Storage panel, navigate to the location of the immersive video content.
3. Drag and drop the media into the Media Pool underneath.

TIP: To retain the original folder structure, drag folders from the Media Storage panel into the Media Pool sidebar. When dragged into the Master bin (or any bin you have created), folders and subfolders are retained.

When an Apple Immersive video clip is selected, the codec parameter in the header of the Metadata panel will indicate that it is an Immersive Video, alongside its native frame rate and resolution.



Additionally, a 3D icon in the lower left corner of the video thumbnail also confirms its status as a 3D clip.



NOTE: When you drag and and drop immersive media into the Media Pool for the first time, a dialogue window might prompt you to choose whether you wish to Keep Settings or Change Settings. Choosing Change Settings will set the Project Settings to automatically adopt the frame rate and resolution of the source media.

Editing

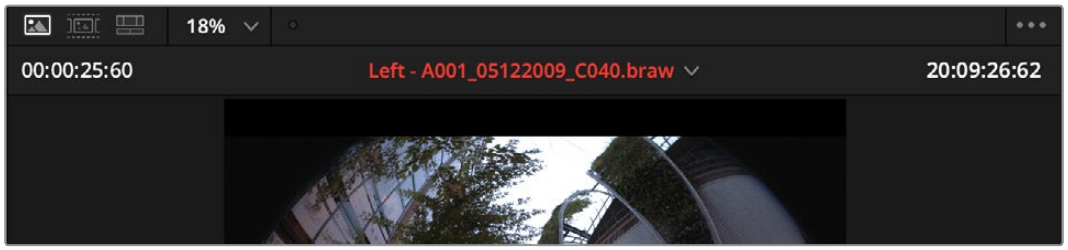
When working with immersive content, editing tools and actions remain mostly the same as when working with conventional 2D video formats. The following set of instructions describe edit features or actions that are unique to immersive video workflows.

Assembly

When previewing clips during timeline assembly, you have the ability to change which eye (left or right) you see in the source viewer. Otherwise, the process of assembling a cut on the timeline remains unchanged from standard NLE workflows.

1 Go to the edit page.

2 Double-click a clip to load it in the source viewer on the left. The label at the top of the viewer indicates which eye you are viewing.



3 To change the eye in the viewer, right-click the source clip in the Media Pool and choose Stereoscopic 3D > Stereoscopic 3D Mode > Left Eye/Right Eye/Both Eyes.

TIP: You can also hover over the thumbnails in the Media Pool to scrub clips in the source viewer if you have Live Media Preview enabled in the source viewer options.

4 Use I and O shortcuts to define a range in the source viewer.

5 Press Insert Clip in the toolbar (or use the F9 shortcut) to insert clips onto the timeline.

TIP: You can also drag and drop clips from the Media Pool into the timeline panel.

6 Use your mouse (in Selection or Trim Edit mode) to trim clips on the timeline.

Backdrops

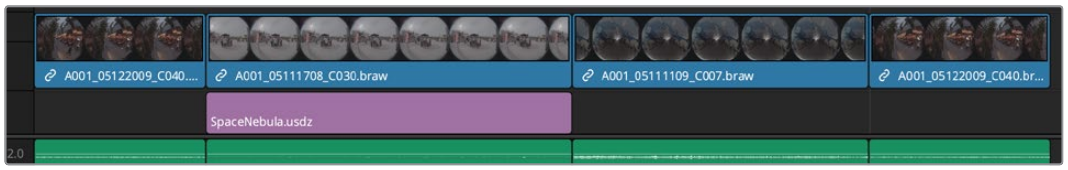
When you select ‘Enable Apple Immersive workflow’ in the Project Settings, a Backdrops track (‘BD’) automatically appears under the Video 1 track.

Backdrops refer to the area that the observer will see when they look beyond the perimeter of the immersive video content, which is typically limited to a 180–210 degree field of vision.

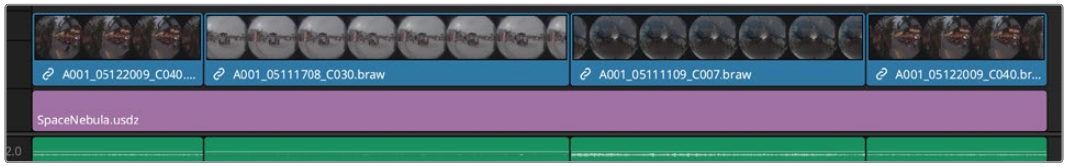
By default, this area is black, but you can insert custom background imagery (in the .usdz format) using the backdrops track.

1 Drag a .usdz backdrop file onto the Backdrops track to add it to the timeline.

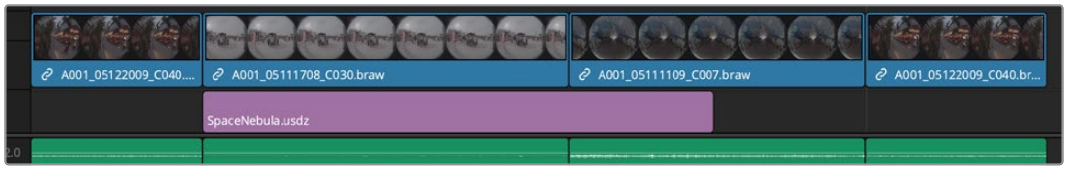
The duration of backdrop images must align to the clip boundaries in the video tracks. A single backdrop file can span one clip, multiple clips, or the entire timeline, but it cannot cut in the middle of an immersive .braw clip (aka 'L-cut').



One clip – correct ✓

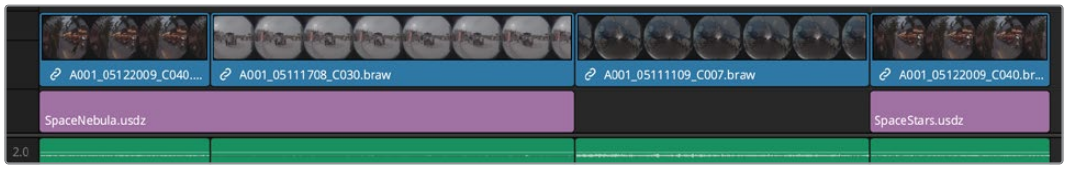


Multiple clips – correct ✓



L-cut – wrong ✗

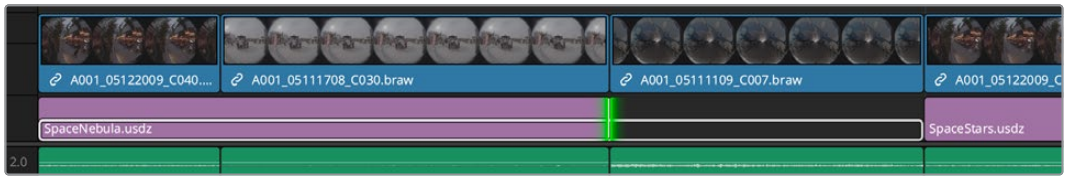
You can opt to have multiple backdrop files within a single timeline, or not add an image at all for the default black backdrop.



Different images and gaps – correct ✓

2 Use standard trimming controls to adjust the length of the backdrop file.

Drag the edge of the .usdz file (in Selection or Trim Edit mode) to adjust the length.



You may also use the , (comma) and . (period) commands to nudge the trim edge one frame at a time.

TIP: To ensure backdrops are always aligned to the Immersive Video edits, enable Snap mode in the edit page toolbar.

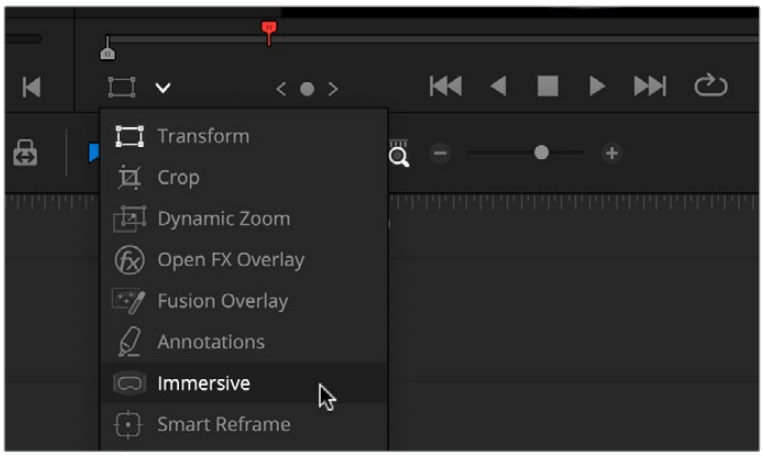
Using the Viewer

By default, immersive clips are displayed in the DaVinci Resolve page viewers in the circular ‘lens space’. The visual lens calibration used to project the image originates from the camera and is unique to every clip.

You also have the option to change the view mode to LatLong or Viewport.

1 In the lower left corner of the timeline viewer, click the dropdown arrow next to the onscreen controls button.

2 Select Immersive.

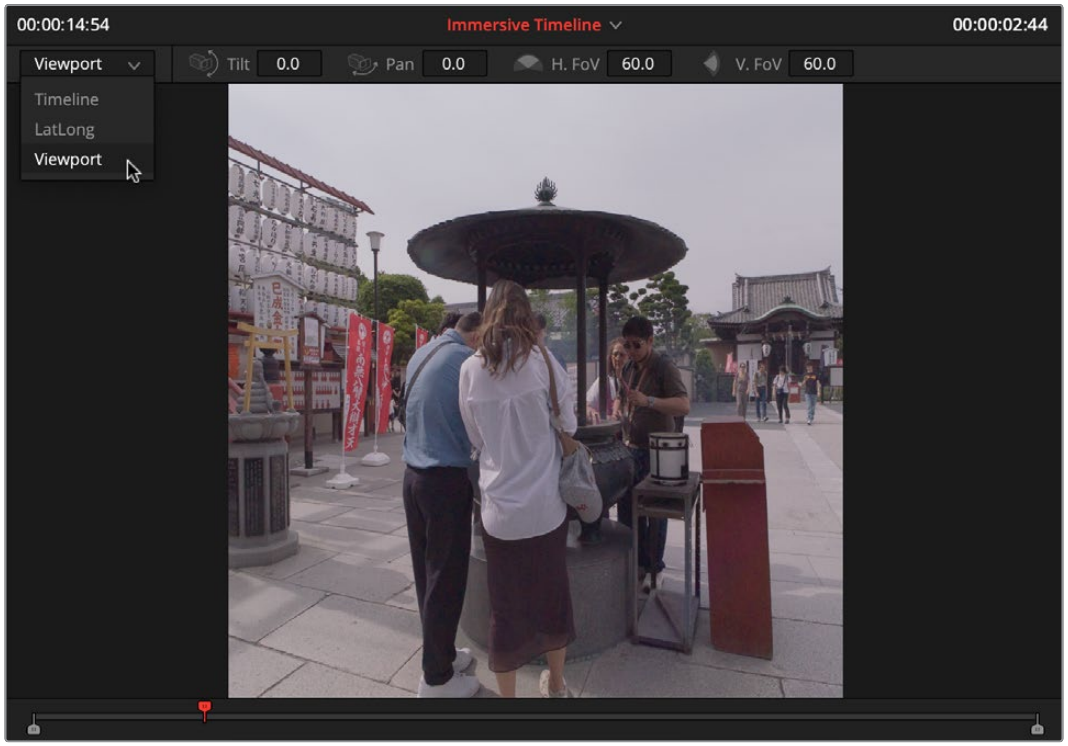


A new toolbar appears above the viewer with additional view options.

3 Click LatLong to see an equirectangular projection of the image. This mode warps the image to more closely resemble traditional 2D media proportions which eases certain post production processes, such as tracking and compositing.

4 Click Viewport to see a center cutout of the immersive scene in a square frame. This mode helps you clearly see the correctly-proportioned details of the scene on a standard monitor, though you will need to be mindful about not missing any off-frame action.

You can use the transform controls in the immersive toolbar at the top to tilt and pan the stereoscopic image in the frame, as well as to adjust the horizontal and vertical fields of view.



Transitions

To transition between two immersive video clips, you can either perform a direct cut, use the handles on either side of the clip to fade in or fade out, or use a Dip To Color Dissolve effect. Transitions should not be baked into the footage, but are rather rendered by the Apple Vision Pro itself during playback.

1 Open the Effects panel.

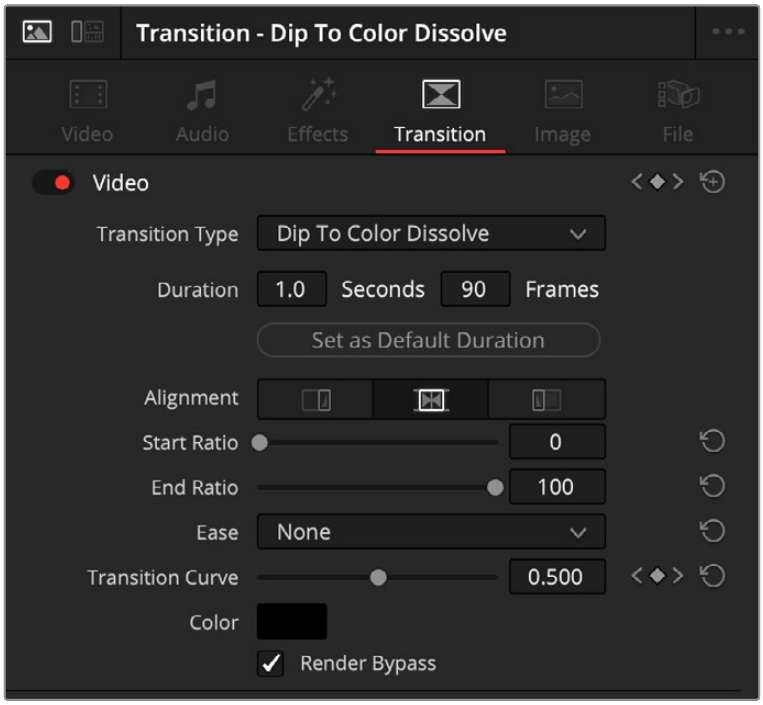
2 In the sidebar, navigate to Toolbox > Video Transitions.

3 Drag Dip To Color Dissolve onto a cut point between two clips.

NOTE: You must have enough trim data beyond the cut points of both clips to apply a transition. If there is insufficient trim data ('handles'), the transition will not apply.

4 In the Inspector panel, navigate to the Transition tab and select Render Bypass. This will ensure that the dissolve is correctly rendered by the Apple Vision Pro.

5 You may then use the remaining parameters in the Dip To Color Dissolve Inspector to modify the duration, alignment, animation speed, and color of the transition.

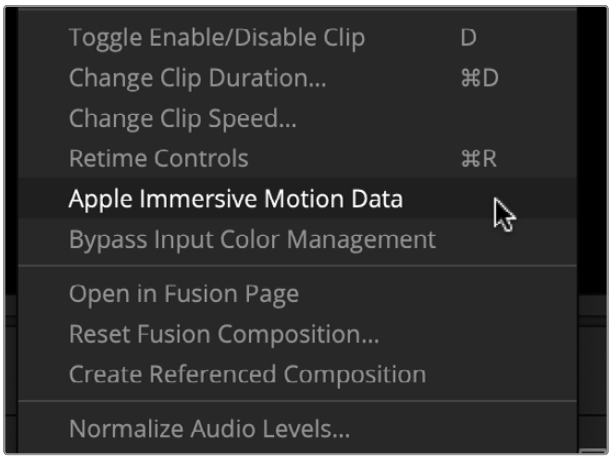


Apple Immersive Motion Data

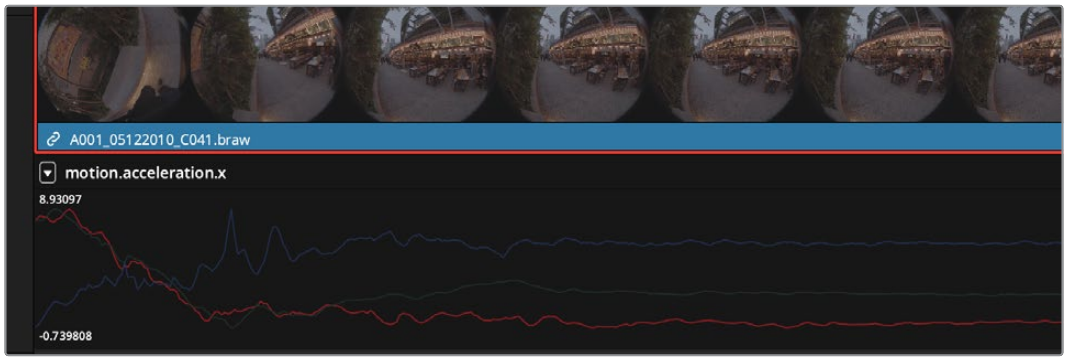
When working with immersive content, quick or sudden camera movement can be disorienting to the viewer. To ensure viewer comfort, avoid using takes with quick pans/tilts, rotating motion, or uneven horizons.

The Apple Immersive Motion Data feature graphs the x, y, and z acceleration and gyroscope position values to objectively demonstrate where camera motion and movement may be too strong. Editors can use this information to avoid using disorienting sections of footage.

1. Right-click an immersive clip on the timeline.
2. Choose Apple Immersive Motion Data.

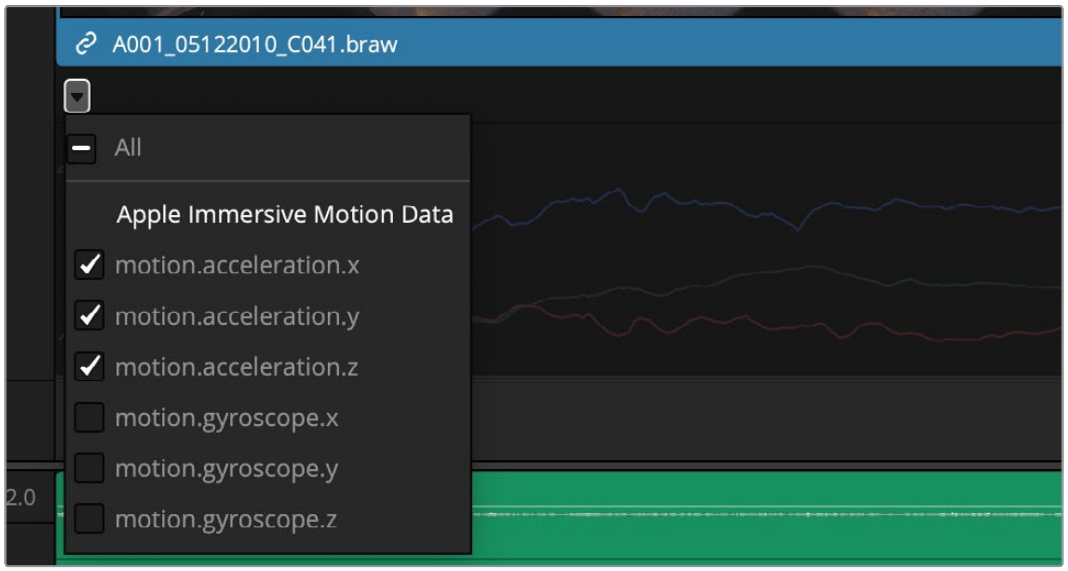


A graph appears in the video track showing the x, y, z acceleration rates.



In this example, the camera was recording as the camera operator was setting it up. The spikes in the graph indicate sudden movement as the camera was hoisted, rotated and placed on his shoulder. Ideally, you'd want to use the latter section of the video with the flatter readouts, after the camera has been steadied.

3 In the upper left corner of the graph, click the displacement arrow.



You can enable or disable readouts based on the content of the shot and the nature of the camera motion.

Titles

In the edit page, you can add text to an immersive scene using any of the title tools in the Effects panel – Text, Text+, MultiText, Scroll, and any of the lower third varieties.

When placed above an immersive clip, the title generator will use the clip's ILPD (Immersive Lens Profile Data) to accurately distort the text to match the lens warp of the scene.

If no ILPD is available (for example, when adding text over a black backdrop or immersive CGI scene), default lens distortion will be applied to ensure the text still appears appropriately warped when viewed in the Apple Vision Pro.

1 In the Effects panel, navigate to Toolbox > Titles.

2 Drag the Text generator into the timeline. Place it on a video track above an immersive clip to incorporate it into the scene.

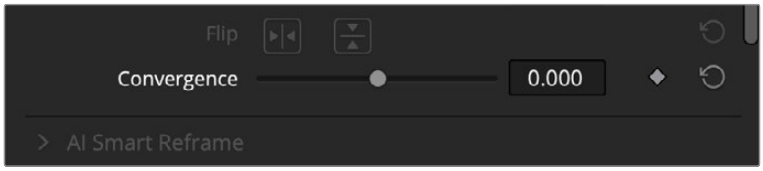
As with backdrops, Text generators cannot cross clip boundaries. As each clip is rendered with an individual lens space, any overlapping titles risk being warped between shots.

3 Trim the title to the video edit point(s).

4 Use the Inspector to write and format the title text.

5 Go to the Settings tab and use the Convergence slider to position the text in 3D space.

Moving the slider to the left pushes the title back in space, while moving it right brings it forward.



Convergence is best observed when monitoring on the Apple Vision Pro. If using other monitoring means, you can change how the text convergence will appear in the viewer by changing the Vision mode of the underlying video layer in the color page.

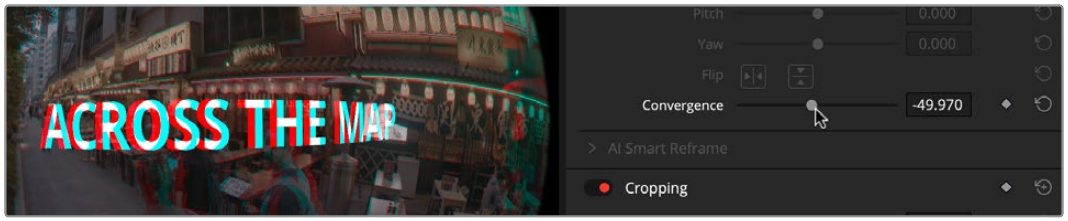
6 Go to the color page.

7 Open the 3D palette.

The Vision column in the centre of the palette allows you to change the stereoscopic standard used to project the immersive media.

8 Select the Stereo tab.

9 Change the Out mode to Anaglyph (Color).



Changing the text generator's Convergence parameter in the edit page's Inspector panel will now appear more intuitive if viewed through anaglyph (or anachrome) glasses.

Creative Considerations

During these early days of Blackmagic URSA Cine Immersive camera and Apple Vision Pro production, filmmakers are encouraged to experiment with the format to identify best practices, establish new rules and discover workflows that traditional filmmaking had not previously imagined.

With that in mind, some aspects of immersive production have already been flagged for special consideration to ensure maximum audience absorption and visual comfort.

1 Editing should begin in pre-production. Due to the special considerations for shot length and audience motion comfort, filmmakers need to capture scenes with the final edit in mind.

2 Avoid or limit using clips with rapid camera movements, as these may disorient the viewer. Instead, opt for locked shots, or slow pans and dollies.

3 Off-balance or extreme camera angles can also be disorienting – opt for takes that are level to the subject, and at a comfortable distance (not too close).

4 New environments take longer to absorb in immersive media than when viewing a 2D scene. Cuts should be slower to allow the viewer time to adapt to new surroundings.

5 When cutting between clips, be mindful of where the viewer would have been looking in the previous shots. Important action shouldn't be taking place on the left side of the frame if the audience's attention was previously towards the right.

6 Spatial Audio drives much of the storytelling – treat it as another character in immersive productions.

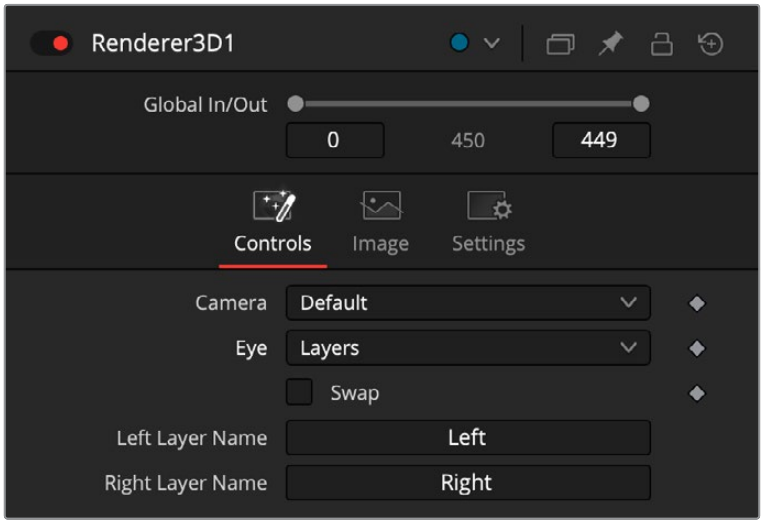
VFX Compositing

When ‘Apple Immersive workflow’ is activated in the Project Settings, dedicated Fusion parameters allow for seamless stereoscopic workflows and immersive-compliant compositing. MediaOut nodes feature left and right eye output options and the Immersive Patcher tool allows for equirectangular projection and composition.

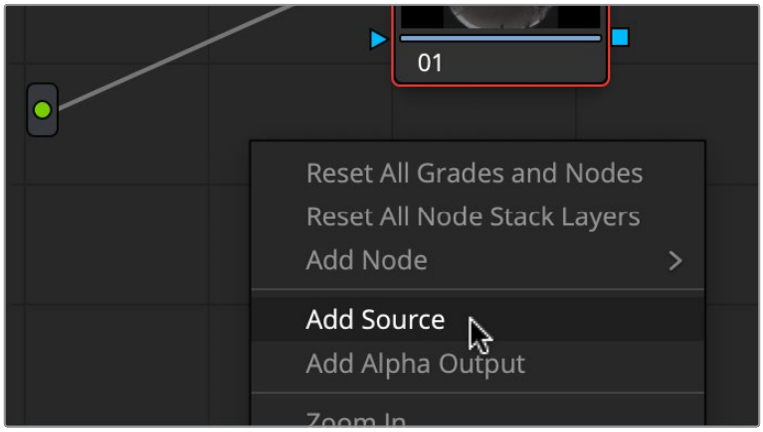
MediaOut nodes

When you first enter the Fusion page while working on an immersive clip, the Nodes graph contains two MediaIn nodes connected to two corresponding MediaOut nodes.

In the MediaOut Inspector, the Stereo Eye parameter can be used to route the eye signal output.



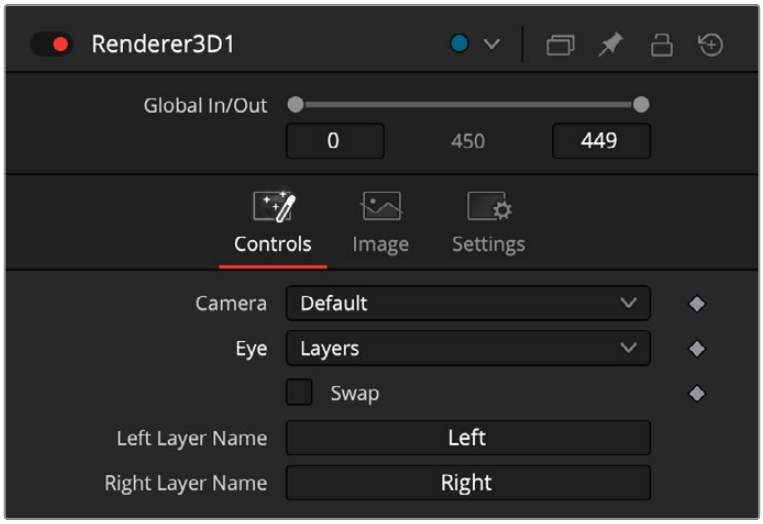
By default, MediaOut1 is set to the Left eye and MediaOut2 is set to Right. This configuration allows you to immediately start assembling stereo-compliant composites and graphic overlays. The dual MediaOut nodes also result in two signal variants being sent to the color page’s node editor for further image refinement. You can access the MediaOut2 signal in the color page by right-clicking in the node editor and choosing ‘Add Source’.



When the Stereo Eye parameter is set to Auto, the MediaOut node combines the signals and assigns them to the appropriate eyes on the timeline. This allows you to use Fusion's multilayer workflow to send left and right signals to a single MediaOut node.

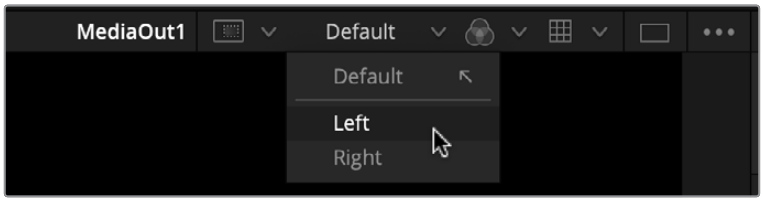
Renderer3D

When working with 3D graphics or CGI on an immersive timeline, the Renderer3D node features multiple eye setups, and the ability to process left and right layers.



This makes it possible to create complex 3D titles or entire CGI scenes and send them directly to a stereoscopic timeline in immersive projects.

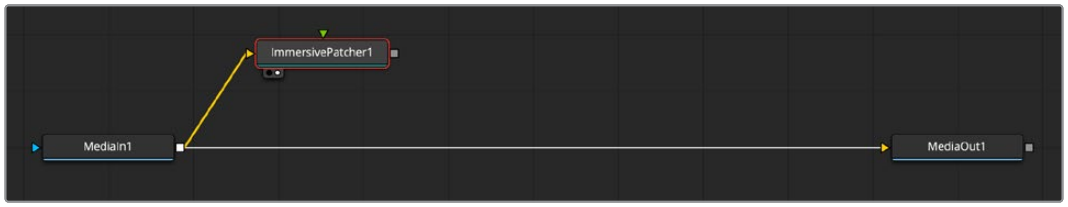
When working with this setup, you can use the drop down menu in the upper right corner of the Fusion viewers to alternative between viewing the left and right eye of your composition.



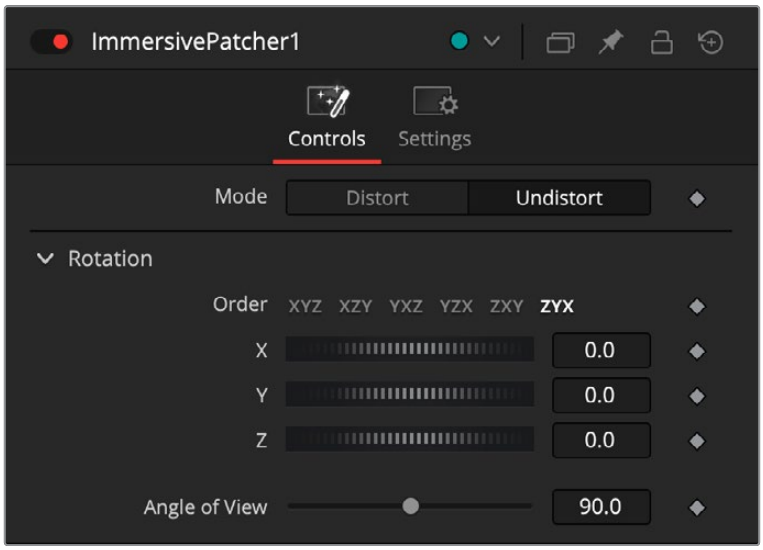
Compositing with the Immersive Patcher

The Immersive Patcher tool allows you to temporarily ‘undistort’ a section of an immersive clip to allow for easier tracking, painting and compositing on a flat plane. Once finished, you can use a second Immersive Patcher to re-distort the immersive signal and integrate the newly composited elements into the scene.

1 Add Immersive Patcher to the node pipeline.



2 In the Inspector, click Undistort.



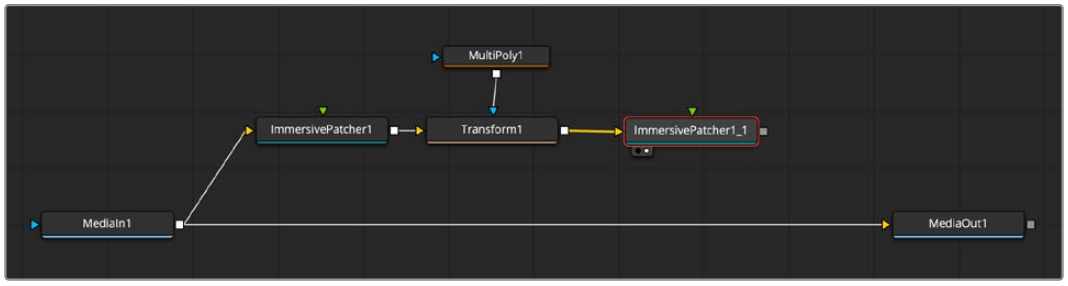
3 Use the X, Y, and Z rotation parameters to choose a section of the immersive image to flatten.

4 Proceed to composite as you would a standard 2D video signal. This includes (but is not limited to) using standard paint methods to remove unwanted objects, sampling sections of the frame with transforms and polygons for set extension, or applying mattes, text, and external images/video layers to replace signs or screens.

Once finished, you need to re-distort the flat image to match the original immersive warp.

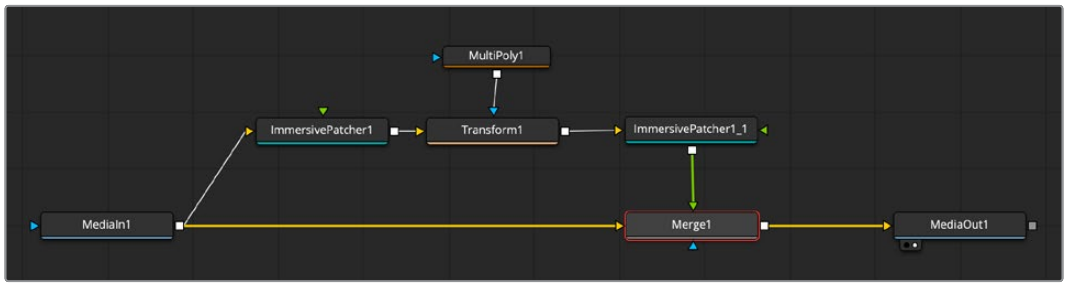
5 Copy the Immersive Patcher to retain the original rotation and angle-of-view information.

6 Paste the Immersive Patcher after the composite nodes and connect to the end of the composition pipeline.



7 In the Inspector, click Distort.

8 Merge the composited patch to the original clip to integrate the effect into the immersive scene.



If working collaboratively, you can exit the Fusion page and continue to edit, mix and grade the immersive timeline in DaVinci Resolve.

If performing immersive turnovers, render out the individual eyes as EXRs in ACES AP0 to retain maximum properties for immersive delivery.

Color Grading

This section covers the optical, color, and 3D tools that you can use when working with immersive media in DaVinci Resolve's color page.

Due to the unique nature of immersive content, not all tools will be possible to use during post-production. Other tools are currently unavailable, but are in the process of being adapted to support immersive content.

Noise Reduction

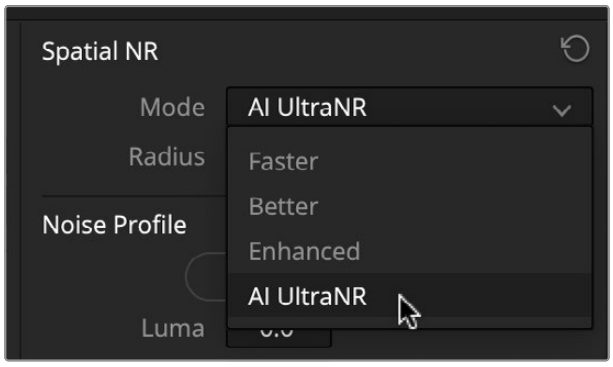
Noise reduction is possible when working with immersive video. In the Motion Effects palette, Spatial NR analyzes the noise pattern of the source media in order to maximally reduce it while retaining original image detail.

1 In the color page toolbar, open the Motion Effects palette.

In the Spatial NR column, you can select which noise reduction mode you wish to use.

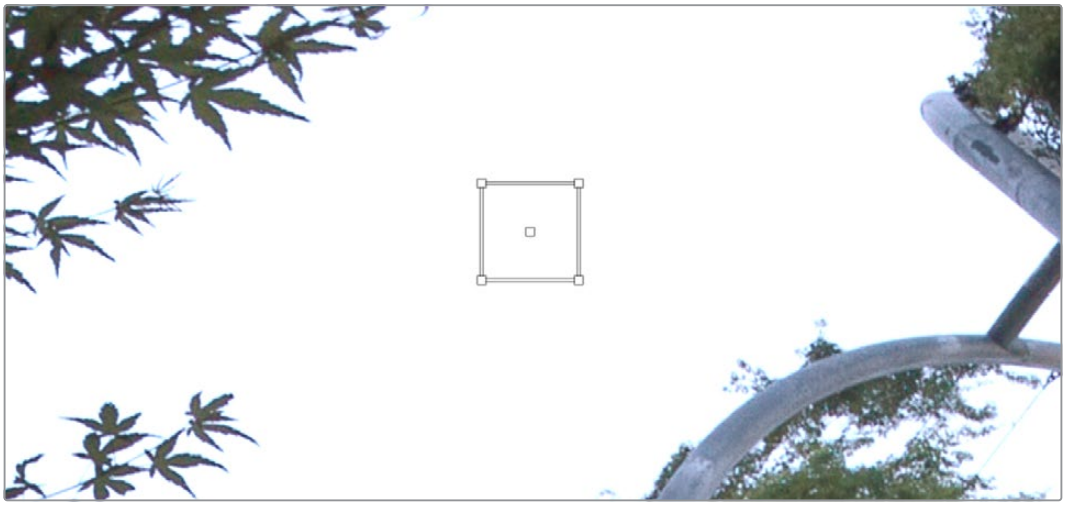
- *Faster*, *Better*, and *Enhanced* modes will all run their pattern analysis within a given radius size and configure their output based on the set threshold. They are listed in order of processing complexity.
- *AI UltraNR* is an AI-driven noise pattern identifier that will extract the noise pattern from the signal by locating and analyzing the "flattest" section of the frame for pure noise.

2 Set the spatial noise reduction mode to AI UltraNR.



3 Click Analyze.

A bounding box appears in the viewer, identifying the “emptiest” area of pixel data in the given frame. With no tangible visual content, the data within the square is ideal for extracting the digital noise produced by the camera sensor.



You can reposition and resize the bounding box in the viewer if you wish, but it is generally not advised—AI UltraNR is better at detecting optimal analysis regions than our eyes.

You can also manually adjust the Noise Profile Luma and Chroma Threshold settings, but again, AI UltraNR tends to automatically achieve the ideal mix of thresholds for clean noise reduction.

Primary Grading

Primary color grading operates the same as it does with regular video footage. Use the same considerations for normalization, balancing, and matching when using primaries color and log wheels, the HDR palette, and the HSL Curves. The Color Slice and Color Warper palettes also work as expected.

Secondaries

Pixel-based selection methods, like the Qualifier, work as expected with immersive media.

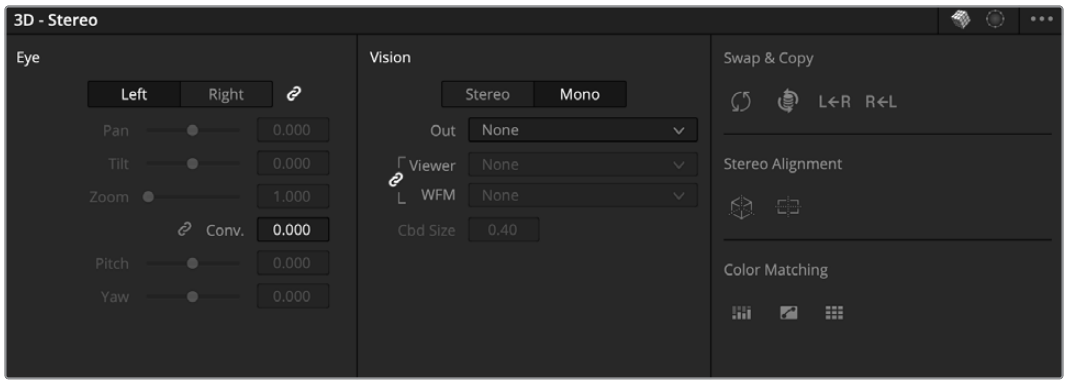
Object- or window-based selection can only work accurately in a flat equirectangular (or viewport) environment, meaning that most secondary methods that require precision in qualification shape are currently unavailable. This includes Power Windows (and, by extension, the Tracker palette) and the Magic Mask tool.

Sizing

There are no sizing controls (i.e. zooming and panning) available for immersive content.

3D Palette

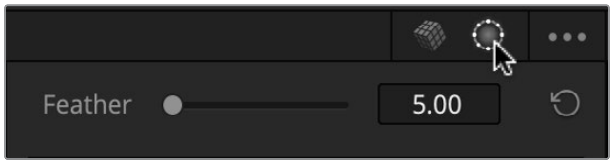
The color page's 3D palette features controls for managing a variety of 3D camera and graphics formats. Unlike with other stereoscopic formats, you do not need to adjust the alignment of the eyes, because the Blackmagic URSA Cine Immersive camera lenses, focal length and irises are fixed.



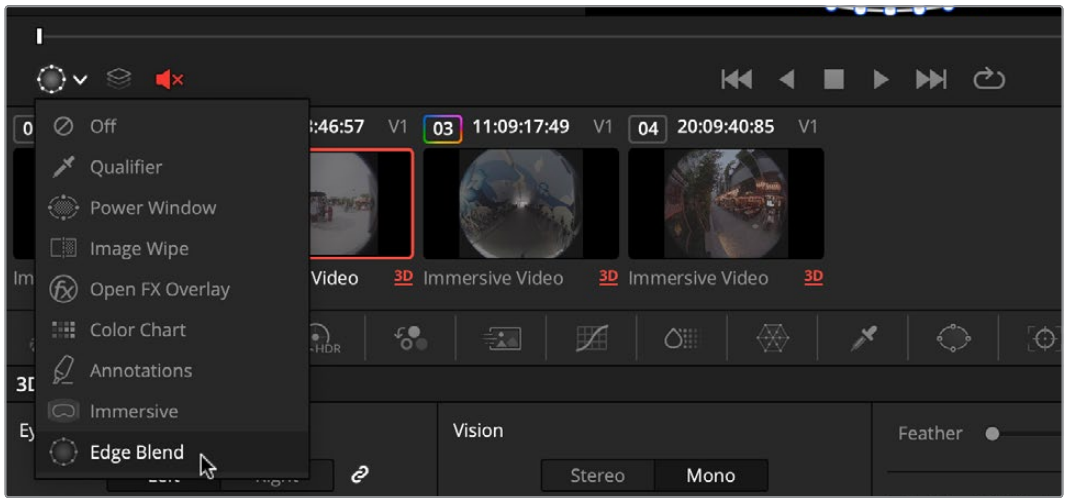
The Edge Blend mode is exclusive to immersive content and allows you to adjust and soften the perimeter of the immersive video frame. This can be used to hide visible film equipment and to limit the observer's field of vision. Beyond this edge, the observer will see the backdrop.

1 Open the 3D palette.

2 In the upper right corner, change the mode to Edge Blend.



3 Ensure the viewer's onscreen controls are set to Edge Blend.



You should see a circle of control points around the perimeter of the video frame, with an anchor handle in the center.

4 Drag a control point to change its position.

Or right-click to delete a point.

5 In the 3D palette's right column, use the Feather parameter to soften the edge of the frame.

6 Select 'visionOS Effect' to ensure the new edge data is correctly processed during playback on the Apple Vision Pro. If not selected, the edge will be baked into the rendered video file.

7 In the left column, use the Left and Right tabs to switch between eyes in the viewer.

Note that many of the controls in the 3D palette are greyed out. Parameters throughout DaVinci Resolve will appear grey when they are not applicable to the media you are actively working on, which in this case is the immersive content.

8 In the center column, use the Stereo and Mono tabs to switch between single and double lens viewers of the immersive video clip.

CST Color Management

Using the Color Space Transform (CST) effect on nodes in the color page's node editor will allow for accurate color mapping and custom control over the maximum output nit luminance of the timeline.

The recommended workflow involves four CST nodes:

1 INPUT CST – camera to working color space.

Color correction occurs between these two nodes.

2 OUTPUT CST – working color space to HDR Mac display.

3 MONITOR CST – 1000 nits to 250 nits for HDR monitoring.

4 AVP CST – 250 nits to 108 nits for Apple Vision Pro monitoring and delivery.

CST Timeline Mapping

First, you will map from the camera source input color space to the 'working' timeline color space. The Blackmagic URSA Cine Immersive camera operates in the Blackmagic Design Wide Gamut Gen 4/5 color gamut with the DaVinci Intermediate log curve.

1 Open the Effects panel.

2 Drag Color Space Transform onto an empty node (01) in the node editor.

When mapping from the input color space to the timeline color space, this node should be among the first nodes in the node tree, before any grading operations.

In cases where you are mapping from the input color space directly to the monitoring or delivery color space, this node should be among the last nodes in the node tree, after grading operations.

3 Label node 01 as **INPUT CST**.

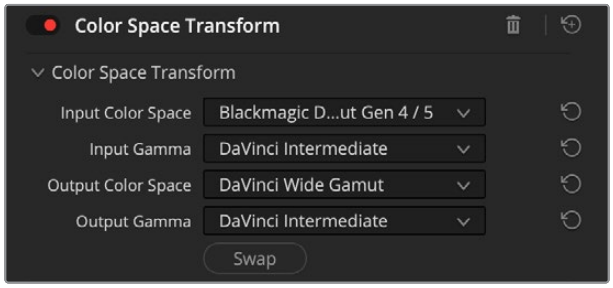
NOTE: The node labels used in these instructions are made for ease of reference to the documentation and screenshots. You may use whichever naming convention works best for you, or bypass labelling altogether.

4 Set the Input Color Space to Blackmagic Design Wide Gamut Gen 4/5.

5 Set the Input Gamma to DaVinci Intermediate.

6 Set the Output Color Space to DaVinci Wide Gamut.

7 Set the Output Gamma to DaVinci Intermediate.



8 Create a new serial node (02).

9 Copy and paste the first node (01) onto the new node (02).

10 Label node 02 as **OUTPUT CST**.

You will want the output color space and gamma to be the input in this new serial node to ensure that the timeline color space of the node tree is consistent.

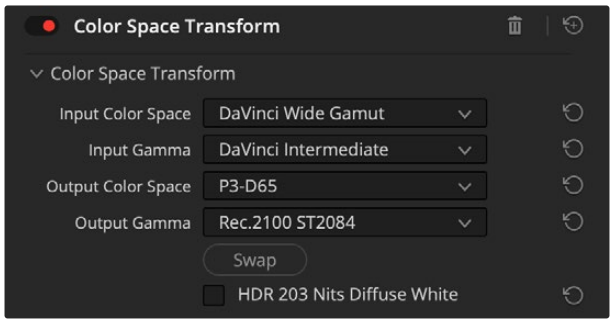
11 Press the Swap button to switch the input and output parameters.

The Input Color Space and Gamma are now appropriate for this CST node.

12 Set the Output Color Space to P3-D65. This is the current default gamut of Mac displays.

13 Set the Output Gamma to Rec.2100 ST2084, which is the HDR standard for a 1000-nit signal.

All grading operations will occur between these two nodes. But as you are working with node-based color management, you also have the unique ability to perform operations outside the perimeters of the timeline color management, which many colorists take advantage of to apply specific grading operations, effects or third-party elements.



ICC Profiles

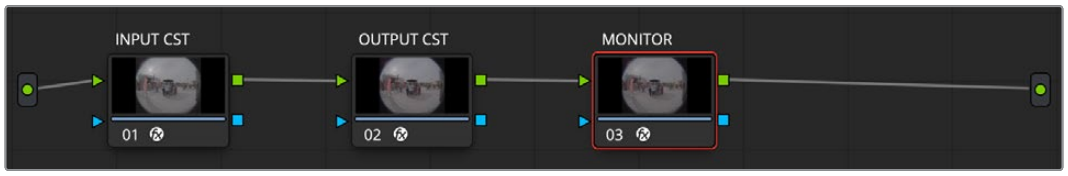
To further ensure that the Blackmagic URSA Cine Immersive signal is correctly mapped to your Mac display, go to the System Settings > Displays and verify that the color profile Preset is set to Apple XDR Display (P3-1600 nits).

CST Grading Monitor Mapping

You can use additional CSTs to temporarily limit the maximum nit output values for monitoring during grading.

1 Create a new serial node (03) after the color management OUTPUT CST (02).

2 Label node 03 as **MONITOR**.



3 Add the Color Space Transform effect to the MONITOR node (03).

The input values have to match the outgoing signal from the previous CST.

4 Set the Input Color Space to P3-D65.

5 Set the Input Gamma to Rec.2100 ST2084.

The purpose of this CST node is only to map the luminance of the image to a smaller nit value, which means the input and output parameters can remain the same.

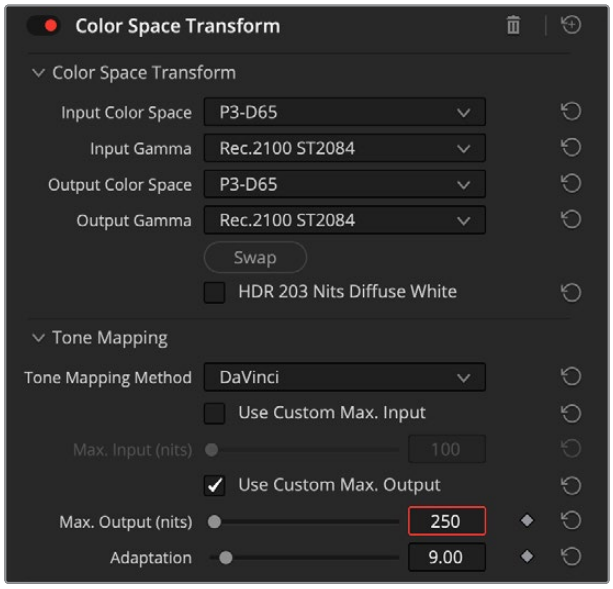
6 Set the Output Color Space to P3-D65.

7 Set the Output Gamma to Rec.2100 ST2084.

NOTE: In some circumstances, it is unnecessary to set the color space or gamma of a CST if the input and output are identical (as no mapping occurs). In this case, it is a necessity because the application of the custom nit mapping is affected by the value and shape of the signal's log curve.

8 Select 'Use Custom Max. Output'.

9 Set the Max. Output (nits) to **250**.



The 1000 nit signal will be now mapped (using the DaVinci Tone Mapping Method) to 250 nits.

This will result in an image that will closely match the rendered Apple Vision Pro signal on an HDR grading monitor (such as Flanders XMP310, Sony BVM-HX310, etc).

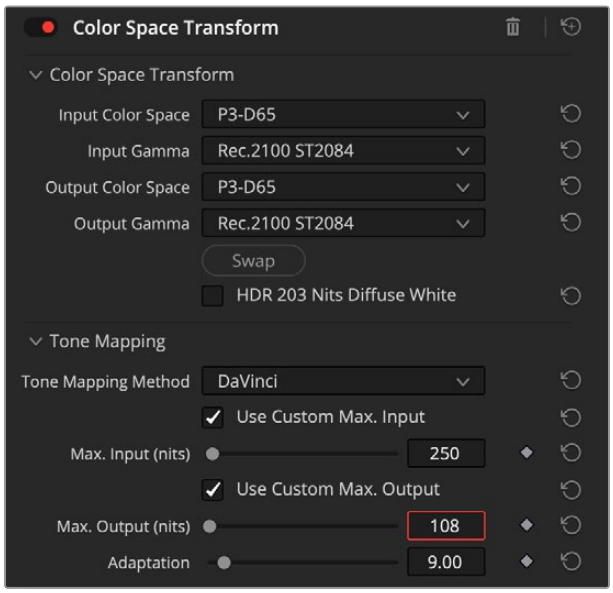
However, to monitor your project on an Apple Vision Pro, you will need to perform one more tone mapping operation.

CST Apple Vision Pro Monitoring

A final CST node can be used to allow for live monitoring, preview and final delivery to Apple Vision Pro devices.

1. Create a new serial node (04) after the MONITOR node (03).
2. Copy and paste MONITOR node (03) onto the node 04. You will be using the same input and output parameters, and only changing the nit count.
3. Label node 04 as **AVP**.

4. Select ‘Use Custom Max. Input’.
5. Set the Max. Input (nits) to **250** to correctly interpret the mapped signal from the MONITOR node.
6. Set the Max. Output (nits) to **108**, which is the maximum nit value for monitoring and delivery to the Apple Vision Pro.



You may now proceed with the grade, disabling the CST nodes if (and when) needed based on your workflow. For example, during your initial grading pass, in which you will be using your external grading monitor, you should disable the AVP node (04) to ensure you are correctly seeing the 250-nit signal in your HDR grading monitor. When switching to previewing in the Apple Vision Pro or for final delivery, re-enable the AVP node (04).

Can the CSTs be placed in the timeline node editor?

No. At the moment, clip tonal mapping relies on the accurate application of the Color Space Transform Adaptation parameter, which must occur on a clip-by-clip basis.

Audio

The Apple Vision Pro and DaVinci Resolve support spatial and ambisonic audio, AVP (Apple Vision Pro) audio, and AIA (Apple Immersive Audio) formats.

Many aspects of audio post production remain unchanged when working with spatial audio. One of the major differences is that rather than monitoring with high-end speakers, audio engineers should instead use headphones or Airpods for true spatial immersion and for a better reproduction of the audience's listening environment.

Whereas in 2D mixes, the majority of dialogue is mixed to the center channels, spatial audio needs to be mixed to the area where the sound is originating. Ducking, equalization and balancing must be used to mix the audio stems in a way that does not overwhelm or confuse the listener.

Immersive spatial elements can be combined with a regular stereo stem, allowing certain sounds to remain constant no matter where the head is moved. This is helpful for keeping non-diegetic sound, such as the music score and voiceovers, clear and easy to interpret.

Environment sounds tend to be interpreted more quietly in spatial audio, so will often need to be mixed more aggressively to be picked up. However, this should not be done at the expense of disorienting the listener.

Reverb contributes greatly to immersion as it is something that listeners experience naturally every day – every sound produced within a given environment bounces off the surfaces that surround the listener, resulting in a 'blooming' reverb quality. Thus, reverb is encouraged in spatial audio design to ensure full immersion.

The Apple Immersive Audio (AIA) format is a container for several formats:

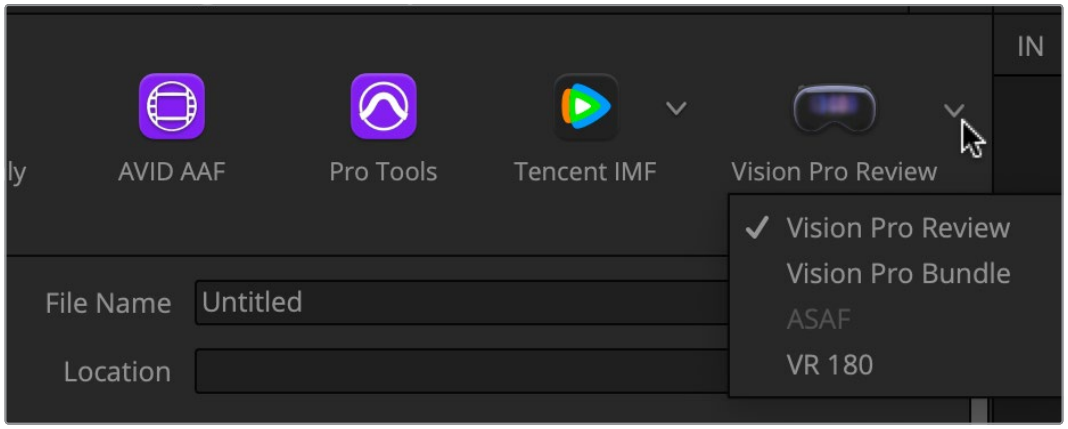
- **Apple Spatial Audio Format (ASAF)** is the raw working audio format.
- **Apple Positional Audio Codec (APAC)** is the encoded .mp4 format playable on APAC-supported devices, like Airpods, or from an AIVU file on the Apple Vision Pro.

Apple Spatial Audio Format (ASAF)

The Apple Spatial Audio Format (ASAF) is an immersive audio format developed primarily for VR and AR applications in the Apple Vision Pro. Exclusive to macOS, ASAF is supported through ProTools (via plugins) and DaVinci Resolve (built-in). ASAF features advanced technical capabilities, including built-in support for Ambisonics sources (called Scenes), and routing via bus assigns. Mixing is virtual, with no physical bus, and users can choose from various monitoring formats, including binaural with headtracking via direct Apple AirPod integration. Panning is handled through DaVinci Resolve's existing tools like the pan dialog, automation, SpaceView, and in-viewer pan tools, with secondary controls in the Inspector. These allow for the precise spatial placement of sources, including automatic tracking against lat-long video. ASAF mixes can be exported in WAV or MP4 formats (with accompanying metadata) via the Immersive Audio > Export Master File option, though they cannot currently be imported or played back within DaVinci Resolve.

Delivery

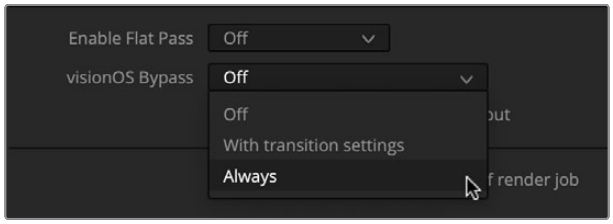
In the deliver page, two render settings allow you to render Apple Vision Pro files for review and final delivery.



- **Vision Pro Review** – creates a small MV-HEVC package for preview purposes. Apple 'share' the resulting file to the Apple Vision Pro via Wi-Fi and watch in the Apple Immersive Video Utility (AIVU) app, which is available in the App store.
- **Vision Pro Bundle** – creates a large ProRes file bundle which can be archived or processed in Apple Compressor to prepare it for playback on Apple TV.
- **ASAF** – renders an ASAF audio mix in WAV or MP4 formats.
- **VR180** – generates a VR180-compliant stereoscopic video file.

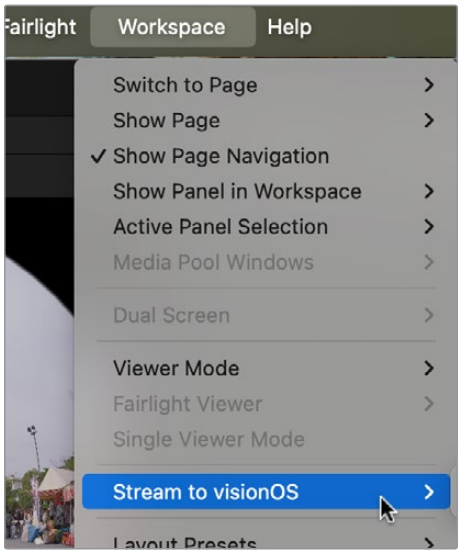
Bypassing Transitions

If a transition is added to an immersive timeline, it will be pre-rendered and turned into a single playback track. The XML generated for immersive content describes the process of switching between these rendered streams (footage, then pre-rendered transition, then back to footage). In the Advanced Settings of the Render Settings panel, you have the option to bypass visionOS transitions (Always), apply them according to their status in the timeline ('With transition settings'), or to include them as they appear in the timeline (Off).



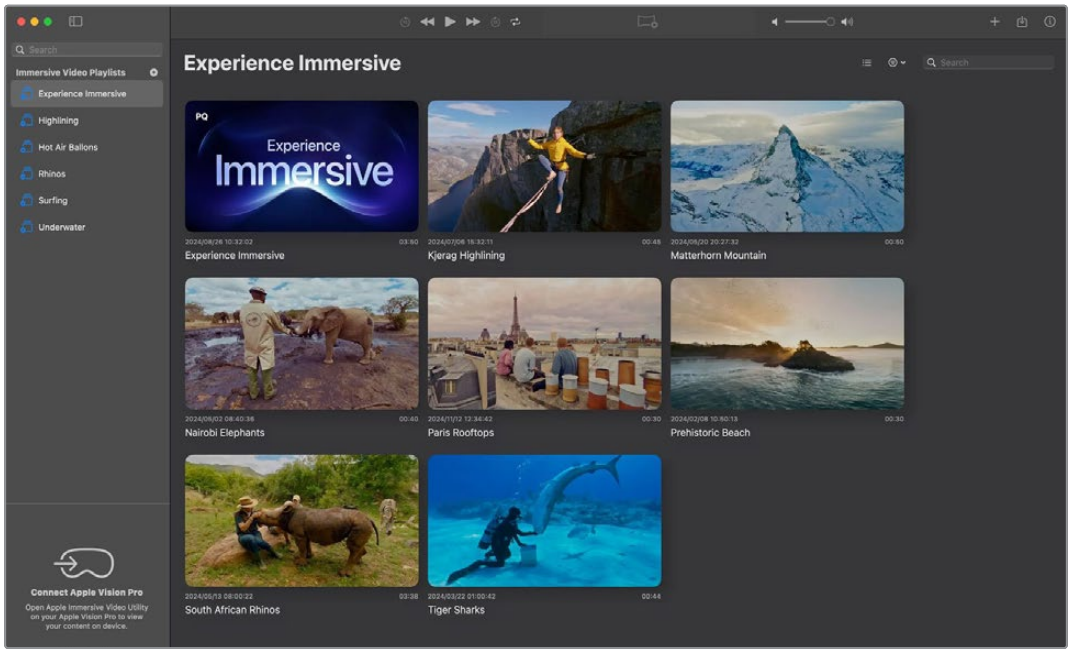
Streaming to the Apple Vision Pro

It is possible to stream to the Apple Vision Pro from the edit, color, and deliver pages in DaVinci Resolve by going to the Workspace menu and choosing ‘Stream to visionOS’ and selecting your AVP device. A fast wireless network (Wi-Fi 5 minimum; Wi-Fi 6 preferred) is recommended for smooth streaming.



About Apple Immersive Video Utility

The Apple Immersive Video Utility is a specialized macOS and visionOS app designed to streamline the creation, management, and playback of Apple Immersive Video content. Tailored for professional filmmakers and immersive media creators, the utility allows users to import, organize, package, and review 8K stereoscopic video files with a 180-degree field of view and Spatial Audio. It supports the creation of AIVU (Apple Immersive Video Universal) files, enabling metadata inspection and modification for post-production workflows. Users can build playlists, sort and search their libraries, and even stream content directly to one or more Apple Vision Pro devices.



URSA Cine Immersive

Congratulations on your purchase or rental of the Blackmagic URSA Cine Immersive.

Blackmagic URSA Cine Immersive is designed to shoot 180 degree stereoscopic VR footage, which is commonly called immersive video. In particular, it has been optimized to shoot Apple Immersive Video for the Apple Vision Pro, the highest quality Immersive experience currently available.

This guide is designed to be an introduction to Blackmagic URSA Cine Immersive and how to shoot with it, focusing on the camera's basic operation and the unique format that is designed to deliver, Apple Immersive Video. An understanding of the format will help you to better understand the camera itself.

In the following pages, you will find a mix of technical and creative advice that will help you not just to use the camera, but to understand the benefits of the immersive format.

What is Immersive?

Immersive is a three dimensional spatial video format. You're likely aware that other similar technologies have existed for some time. To understand what makes immersive different, here's a broad overview of other technologies:

- **Traditional 3D Cinema** allows the audience to experience depth within the frame by capturing each scene with two cameras with an interaxial distance that mimics the distance between human eyes. While stereoscopic images allow the audience to experience depth, the audience's perspective is controlled by the cinematographer's composition.
- **Spherical Video** combines the principles of 3D and 360 degree cameras to allow audiences to look in any direction while experiencing depth. 360 degree images require very high resolutions to ensure the small portion of the image that the audience is looking at has sufficient fidelity. As human physiology only allows us to turn our head from shoulder to shoulder, to be able to look behind, the audience is typically standing or seated on a swivel chair.
- **Volumetric Video** allows the audience not just to experience depth and look around, but to move and change their perspective. While camera arrays can capture volumetric video most virtual reality or augmented reality makes use of dynamically rendered 3D assets and requires a safe environment for participants.

Immersive combines the best of traditional cinema and 3D interactive formats. In its simplest form, it is a high frame rate, high resolution 180 degree stereoscopic format.

Apple Immersive Video

Blackmagic URSA Cine Immersive is designed specifically to capture Apple Immersive Video. This format is used with Apple Vision Pro and supports high resolution, stereoscopic 3D video in a 180 degree field of view.

Apple Immersive Video has the following features:

- 90 frames per second playback.
- Can be distributed at resolutions up to 4320 x 4320 per eye.
- A unique metadata system for mapping the original camera image into the 180 environment.
- Apple Spatial Audio Format to provide 3D sound.

Apple Immersive Video requires hardware capable of high resolution stereoscopic 180 degree playback. Apple Vision Pro is currently the only device that supports this format.

Apple Vision Pro

The headset features dual 3660 x 3200 pixel micro OLED displays with a 7.5 micron pixel pitch. It includes 13 tracking cameras, a LiDAR scanner, internal measurement units, and eye tracking sensors to align imagery with head and eye movement.

An M2 processor with an 8 core CPU and 10 core GPU drives playback at up to 100 frames per second.

URSA Cine Immersive

Blackmagic URSA Cine Immersive is designed for capturing Apple Immersive Video for Apple Vision Pro.

Blackmagic URSA Cine Immersive

URSA Cine Immersive uses two matched 58 megapixel RGBW sensors. The sensors are nearly square to capture a full 180 degree field of view, both horizontally and vertically. Each sensor is paired with a 210 degree fisheye lens. The lens centers are placed 64mm apart.

Built in ND filters provide up to 8 stops of light control. The lens aperture is fixed at approximately f4.5.

The camera records two 8K image streams at 90 frames per second. This results in significantly higher data rates than standard 4K 24p recording.

Getting Started

Getting started with your Blackmagic URSA Cine Immersive is as simple as powering your camera and turning the camera on. Blackmagic Media Module 8TB is loaded into the camera's module bay, formatted and ready to record clips straight away!

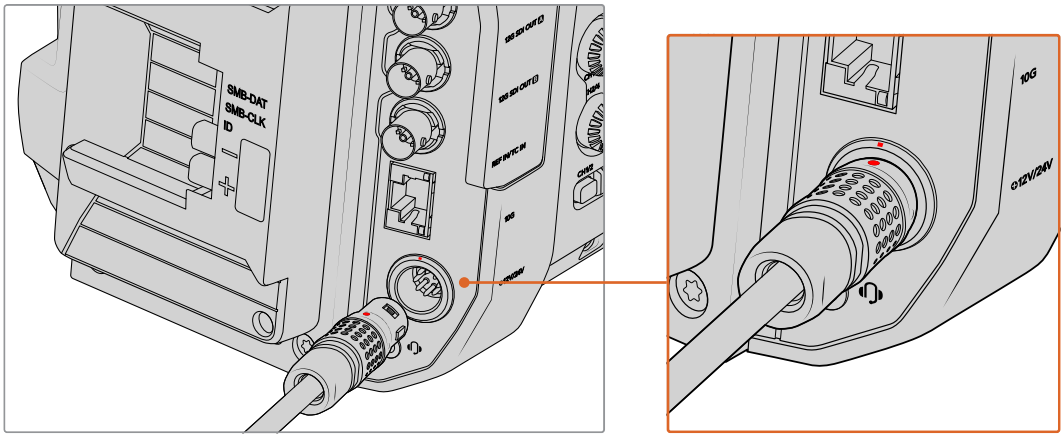
Powering your Camera

The quickest way to power your camera is to connect external power using the supplied power adapter.

To plug in external power:

1. First remove the rubber protective dust cap and store it somewhere safe. Connect the AC to 24V DC adapter plug to your mains power socket using a standard IEC C13 mains power cable.
2. Connect the adapter's 8 pin connector to the 24V DC power connector on the camera's rear panel.

If you have both external and battery power connected, the camera will draw from the external power source. Removing external power while a charged battery is connected will switch your camera to battery power without interruption.



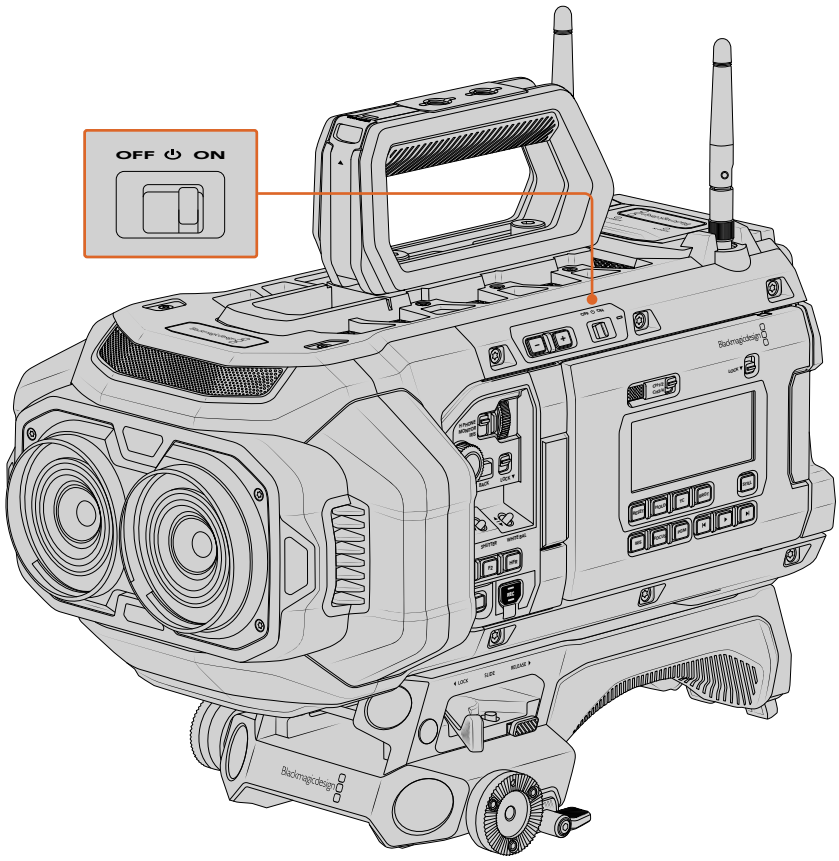
To plug in power, align the red dots and gently push the connector in until you feel it lock into place. To remove, hold the connector's metal sleeve and gently pull away from the camera. This will pull the sleeve back, releasing the connector from the socket.

The battery plate included on your camera is a B mount plate designed for 24 volt B mount batteries. Operating in the 24 to 34 voltage range is ideal for use with URSA Cine Immersive as the higher voltage reduces thermal losses allowing for high frame rate recording and powering multiple accessories from your camera.

Optional VLock and Gold Mount plates, compatible for 24v operation and Gold Mount Plus plates for 26v third party batteries are also available.

NOTE When running URSA Cine Immersive with 12 to 18 volt power supplies or batteries, powering accessories via the 3 pin RS connector or 2 pin lemo style connector is not possible.

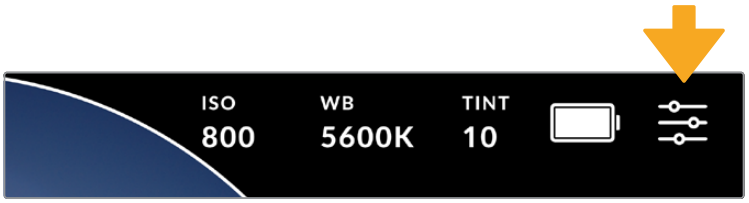
Once you have mounted a battery or connected external power, turn your URSA Cine Immersive on. Move the switch at the top of your camera to the ‘on’ position.



Move the power switch to ‘on’ to power your camera

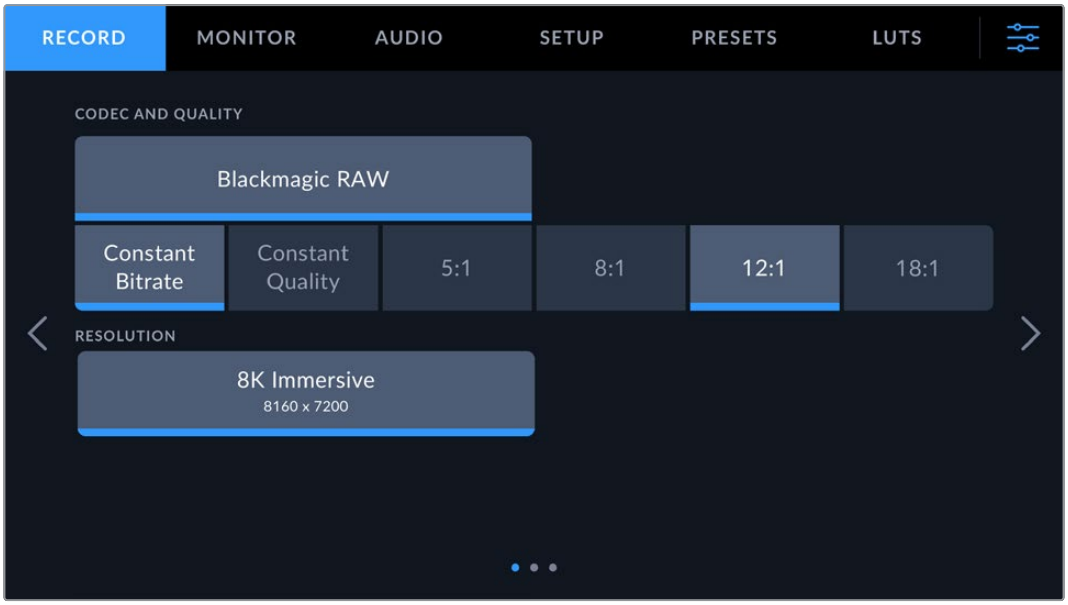
Setting the Record Settings

This section provides information on specific camera settings that are unique or important when shooting immersive content. To open your camera’s settings menu, tap the menu settings icon in the top right corner of the fold out or assist touchscreen.



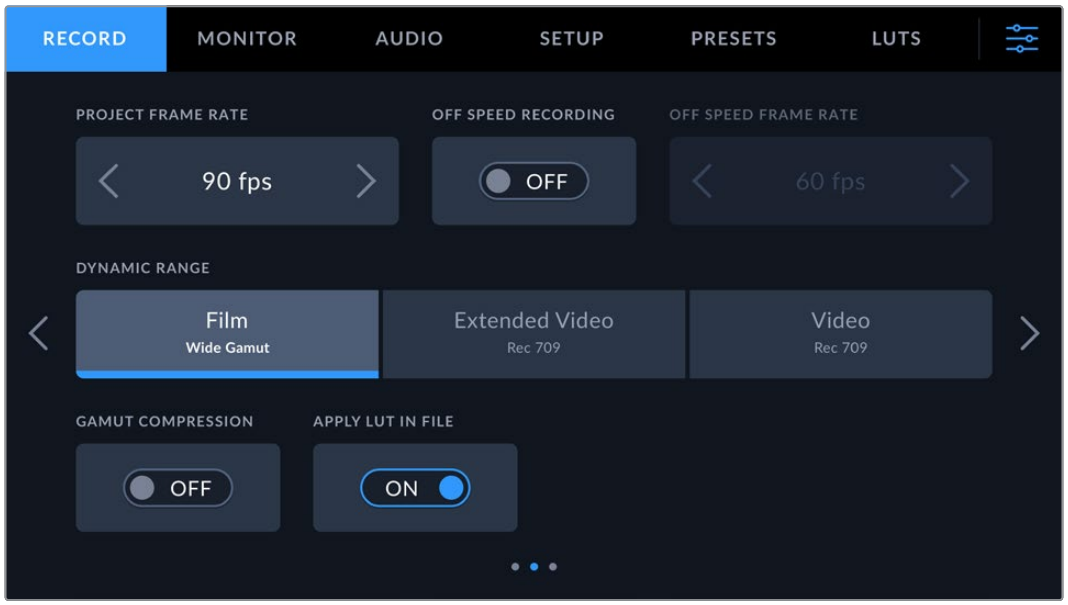
Record Settings 1

The ‘codec and quality’ options on this page lets you choose the Blackmagic RAW codec setting to suit the project you are working on. The options are divided between four ‘constant bitrate’ and ‘constant quality’ settings. We highly recommend using 12:1 or Q3 compression settings if recording 90fps content for Apple Vision Pro. These settings provide the sweet spot for maintaining manageable data rate and file sizes, with current computing hardware, while still retaining extremely high image quality.



Record Settings 2

The second page of your camera's 'record' settings includes the frame rate settings.



Your camera's project frame rate should be set to 90 frames per second to conform to Apple Vision Pro's display frame rate.

If you are shooting for other systems, lower frame rates such as 60fps or 50fps may be sufficient. Lower traditional film style capture frame rates such as 24fps or 25fps may be appropriate if you are shooting visual effects plates for 24p motion picture film content.

That's all there is to getting started with your URSA Cine Immersive! With the media module already formatted in your camera, you can start recording straight away.

Monitoring your Image

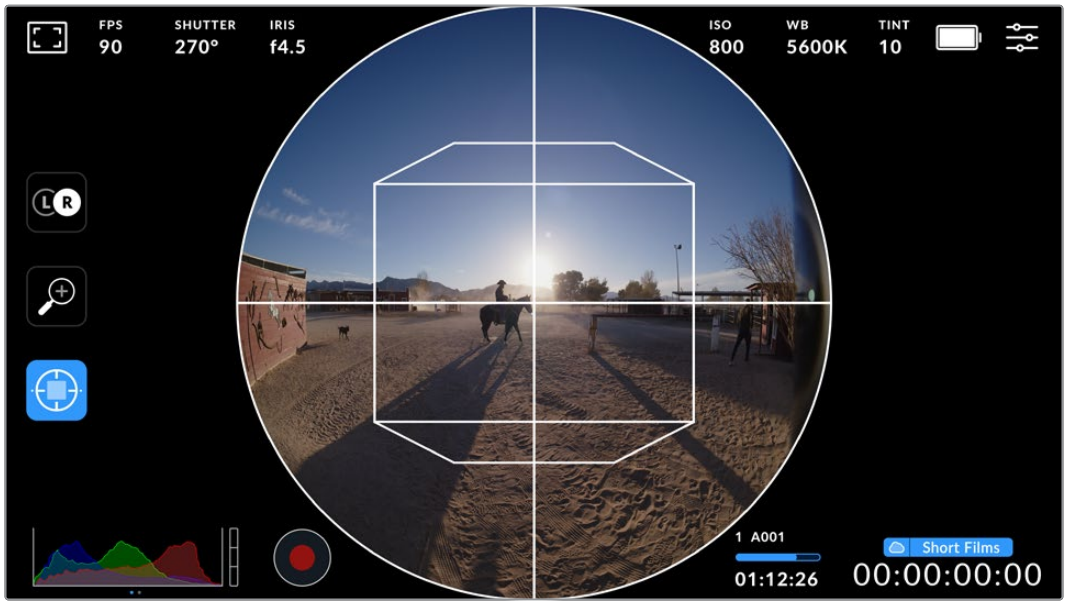
Your URSA Cine Immersive includes several options for monitoring your image while shooting.

The fold out touchscreen is intended for use by the camera operator. It can be reversed to reduce the camera's physical profile while still keeping the screen visible, or positioned forward to enable self monitoring when filming yourself.

The assist touchscreen allows another crew member to monitor the image without obstructing the operator's view. Due to the camera's extremely wide field of view, care should be taken when using the assist touchscreen.

Monitoring Options and Overlays

The camera displays lens space, with one eye viewable at a time. The fold out and assist touchscreens are masked to display a 180 degree view. The SDI outputs show the full 210 degree image captured by each sensor.



All monitoring outputs, display the same eye. You can switch between the left and right eye by tapping the 'L R' eye toggle button on either touchscreen.

An optional overlay can be enabled on the touchscreens by pressing the overlay button. This overlay includes the following:

- A 180 degree outer boundary
- A central crosshair marking the centre of the frame
- An inner guide representing the approximate field of view seen on an Apple Vision Pro when looking straight ahead.

These tools are designed to assist with accurate framing and positioning during immersive capture.

Framing your First Shot

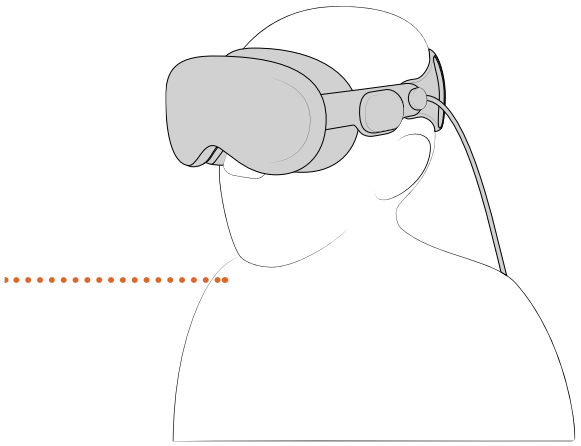
Immersive and traditional 2D content shares a lot in common but there are also distinct differences.

For example, have you ever got motion sickness in a car because you're too busy scrolling on your phone? It is caused by a sensory mismatch when your brain receives conflicting signals. Your inner ear is telling your body you are moving, but your eyes are telling it you are stationary.

It's easy to produce a similar experience with immersive. For example, if you pan the camera, your audience's eyes will tell them they are moving, but their neck and inner ear will tell them they are very much stationary. As a result, it's best to master stationary shots before learning how to make camera movement pleasant for your audience.

Camera Height

In traditional 2D cinema, you might place the camera high or low to signify the balance of power in a scene. When shooting Immersive, consider the impact of placing the camera on the ground. How might that affect the viewer? It could give the viewer the sensation of being buried up to their neck.



The perceived height of the floor can have an impact on your audience's viewing experience. A low camera will give your audience a perceived floor height that might not make sense!

When shooting immersive, low angles could make the audience feel uncomfortable

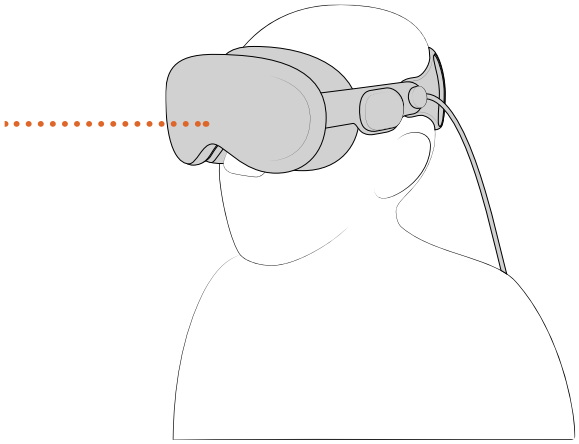
To help the viewer feel physically present, it's often best to choose a camera height and position that makes the viewer feel like they are standing or sitting in the scene. There is still room to express your creative intent. If there are people present in the scene, a slightly higher camera can imbue the viewer with power, allowing them to look down on others. Conversely, a slightly lower camera forces the viewer to look up at other people.

That's not to say that it's not possible to maintain audience comfort with more adventurous camera placement. Flying cameras, top down views, low angles and many other techniques are successfully being used in Immersive productions. When combined with the right editing, movement, and context, they can be used to entertain your audience.

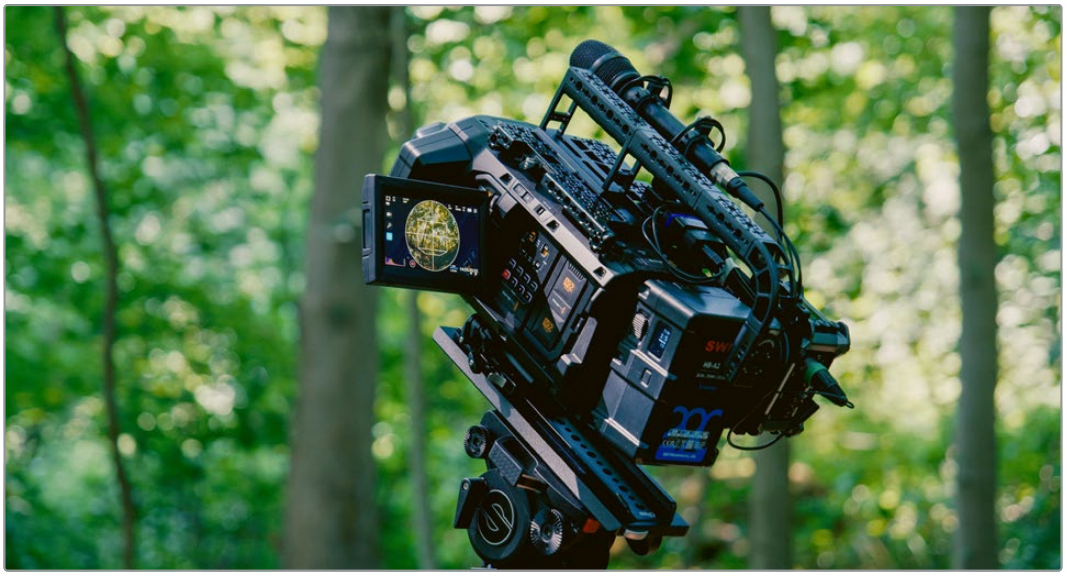
Camera Tilt

When shooting traditional 2D content and placing the camera high or low, the cinematographer will often use camera tilt to maintain composition. Consider how that would affect an Immersive image.

If you were to angle the camera downwards, for your audience to view the horizon, they would need to angle their head upwards. This would be unlikely to cause motion sickness, but it could break their sense of presence because in real life, the horizon is always directly ahead.



Immersive shots will make most sense when the horizon is at zero degrees.



If you were to tilt the URSA Cine Immersive, there will be a disconnect for the audience between where they have to look to see the horizon and where they would normally look to see the horizon.

Therefore, most Immersive shots are captured with a zero degree, or near zero degree tilt. As the camera captures a 180+ degree field of view, both horizontally and vertically, the audience is still able to look up and down. But instead of the filmmaker forcing the audience to look in a particular direction, the audience is in control.

To encourage the audience to look at different parts of your image, you can use other techniques like lighting, aperture framing, and leading lines.

Positioning

Immersive is typically viewed when seated. As it is not an augmented reality or virtual reality format, viewers do not stand or move around. Are you seated presently? As an experiment, turn your head from side to side and move it up and down. How far can you comfortably turn your head without moving your body? The immersive format's field of view exceeds most people's comfortable range of rotation. That's a good thing because it means if you rotate to the left or right, the format still provides peripheral vision. This experiment helps you start to understand where your main action should be positioned within the field of view.

Typical head movement range:

Rotation 120 to 160 degrees
Flexion and Extension: 110 to 130 degrees

When head rotation is combined with eye movement, it is possible to look over your own shoulder, however it's uncomfortable for most people to sustain.

To aid composition, enable your camera's overlay. Keeping action within the inner marks that show typical field of view will aid your content's viewability.

The movement of points of interest within the frame should also be considered. Tracking a person as they walk from extreme left to extreme right requires extreme head movement. Similarly, cutting from a shot where the point of interest is at the edge of the image to a shot where the point of interest is on the opposite edge requires extreme head movement. Careful consideration of composition can aid audience comfort while still communicating creatively.

Audience Agency

When shooting traditional 2D content, you can use the frame, choice of lens and focal settings to direct the viewer's attention. When filming immersive, the 180 degree field of view makes it harder to exclude information and direct the audience's attention. Immersive gives the audience far greater agency to explore your images. This is one of the factors that makes immersive content so engaging. It's still possible to guide your audience's attention though careful use of directional sound, lighting, blocking and composition.

Camera Movement

An immersive camera does not need to remain static, but care must be taken as some camera movement can feel unnatural or uncomfortable. The car is a good analogy for helping to understand how to make immersive camera motion comfortable. A large vehicle has mass and inertia which resists sudden changes in speed or direction. As such, it will accelerate and decelerate over time and move in a predictable and controlled manner. Once you have mastered static shots and move onto moving immersive shots, giving consideration to these factors will help you to create pleasing motion:

- **Duration:** Allow your audience time to adjust to the camera motion and to enjoy it. Brief movements are more likely to cause disorientation.
- **Intensity:** Low intensity moves are easier to integrate into an immersive edit. High intensity moves are possible, but a warm up and cool down might aid audience comfort.
- **Visual Anchors:** These are parts of the scene that don't move. A shot without visual anchors can be disorienting. Foreground elements like the inside of a car or distant objects like the horizon will help ground your moving shots.

Data Rate Tables

The tables below contain data rates for 8K x 7K immersive content at 90 frames per second and 24 frames per second.

Maximum Data Rates at 90fps with Blackmagic RAW Constant Bitrate

8K x 7K Immersive			
Codec	**Pixel Dimensions**	**Frame Rate**	**Data Rate**
Blackmagic RAW 5:1	8,160 x 7,200 Stereo	90	3,203 MB/s
Blackmagic RAW 8:1	8,160 x 7,200 Stereo	90	2,002 MB/s
Blackmagic RAW 12:1	8,160 x 7,200 Stereo	90	1,335 MB/s
Blackmagic RAW 18:1	8,160 x 7,200 Stereo	90	889 MB/s

Maximum Data Rates at 24fps with Blackmagic RAW Constant Bitrate

8K x 7K Immersive			
Codec	**Pixel Dimensions**	**Frame Rate**	**Data Rate**
Blackmagic RAW 5:1	8,160 x 7,200 Stereo	24	854 MB/s
Blackmagic RAW 8:1	8,160 x 7,200 Stereo	24	534 MB/s
Blackmagic RAW 12:1	8,160 x 7,200 Stereo	24	356 MB/s
Blackmagic RAW 18:1	8,160 x 7,200 Stereo	24	237 MB/s

Audio and Timecode

Audio is a critical part of the immersive experience. It's important to pair your URSA Cine Immersive with dedicated audio capture equipment.

Apple Spatial Audio Format

Apple Spatial Audio Format, or ASAF, is used to provide 3D audio when watching Apple Immersive Video. ASAF is part of the wider Apple Positional Audio Framework and can store three different types of audio recording.

- **Channel based:** You can embed a traditional surround mix in your ASAF file. This is ideal for things like head locked narration and music.
- **Object based:** Sounds can be treated as individual objects with metadata that describes their position in 3D space, independent of individual speaker channels. These are typically mono recordings. Their 3D coordinates can be animated to follow objects that move within the scene. This is ideal for diegetic audio, which refers to objects that are visible in the scene.
- **Ambisonic:** A full sphere surround format that can render audio in all directions using spherical harmonics. It is ideal for environmental sound.

The rest of this section provides an overview of the equipment that you will need to build an effective ASAF file.

Ambisonic

An Ambisonic microphone contains four or more capsules arranged in a precise geometric pattern. Each capsule captures the same sounds with different arrival times and intensity depending on the direction that a sound comes from. Those capsules are recorded to discrete channels for processing by a computer later. By comparing the differences, software can estimate the direction and nature of the recorded sounds.

A first order ambisonic microphone

Ambisonic Microphones

Ambisonic microphones are classified by the number of capsules they contain. More capsules means the microphone can reproduce spatial detail more accurately.

First order:	4 channels
Second order:	9 channels
Third order:	16 channels
Fourth order:	25 channels and above

Whether the microphone is on the camera or by its side, it's critical that ambisonic microphones are correctly aligned with the camera's shooting direction. Additionally, because ambisonic microphones can be used in different configurations, such as endfire, broadside, upward or downward, it's important to make sure your audio technician knows which configuration was used. This information can be conveyed via metadata on the audio recorder or by sharing notes.

Ambisonic recordings can be used as an environmental bed. Other channel and object based sound effects can be layered on top. If correctly aligned with the camera, ambisonic recordings can also serve as a reference for the audio engineer as they add further sound design.

Lavaliere and Miniature Microphones

When you are filming with a 180 degree camera, it's impossible to use a boom pole as it will be visible in your scene! This means you'll need to use hidden microphones and lavalieres to record diegetic audio and dialogue.

For example, you could:

- Connect those audio sources directly to the URSA Cine Immersive's two XLR inputs using wireless transmitters and receivers.
- Connect them to an external audio recorder. That could be a large multi channel device or a miniature body worn style device.

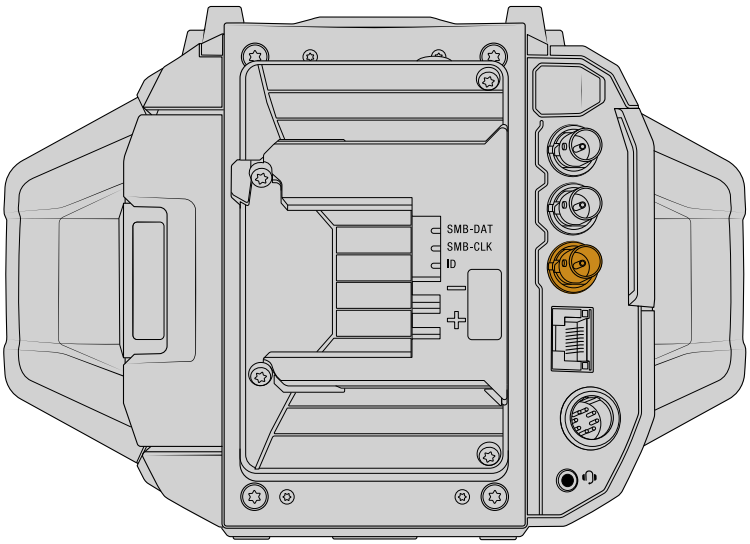
These recordings can then by synced and mixed as either channel or object based sounds.

Handheld Microphones

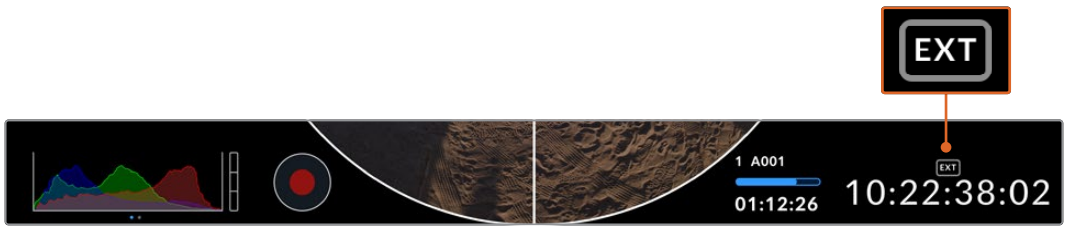
Recording additional sound effects on location after the camera has stopped rolling can not only save time in post production but may also yield a more authentic sound. Carrying a handheld hypercardioid or similar microphone on production is very beneficial, allowing actions to be re performed with the microphone positioned as closely as possible. These recordings can then be edited into the final mix as either channel or object based sounds.

Connecting Timecode

To connect external timecode to your URSA Cine Immersive, use a BNC cable to connect a timecode generator to the timecode input on your camera's rear panel. Connecting to syncronized timecode when shooting dual system ensures audio and picture can be easily synchronized during post production.



When external timecode with a matching frame rate is plugged into this connector the camera will lock to the incoming timecode automatically. The ‘EXT’ logo will appear next to the timecode counter on URSA Cine Immersive’s LCDs when external timecode is locked. If you unplug the cable, timecode will remain jammed and the ‘EXT’ logo will switch to ‘INT’ to let you know it is now running from your camera’s internal timecode clock.



The camera timecode will show ‘ext’ when connected to a timecode generator

NOTE SMPTE timecode standards exist for typical frame rates like 23.976, 24, 30, etc. There is not a SMPTE timecode standard for 90 frames per second. As such, when the URSA Cine Immersive’s Project Frame rate is set to 90 frames per second, the camera actually uses 30fps timecode. The SDI outputs also operate at 30p. Any timecode accessories or external audio recorders should be configured to 30fps.

Shooting

Now that you have a basic understanding of how to operate your Blackmagic URSA Cine Immersive camera and make the most of the Immersive format, this chapter will further explore the operation of the camera.

Exposure Requirements

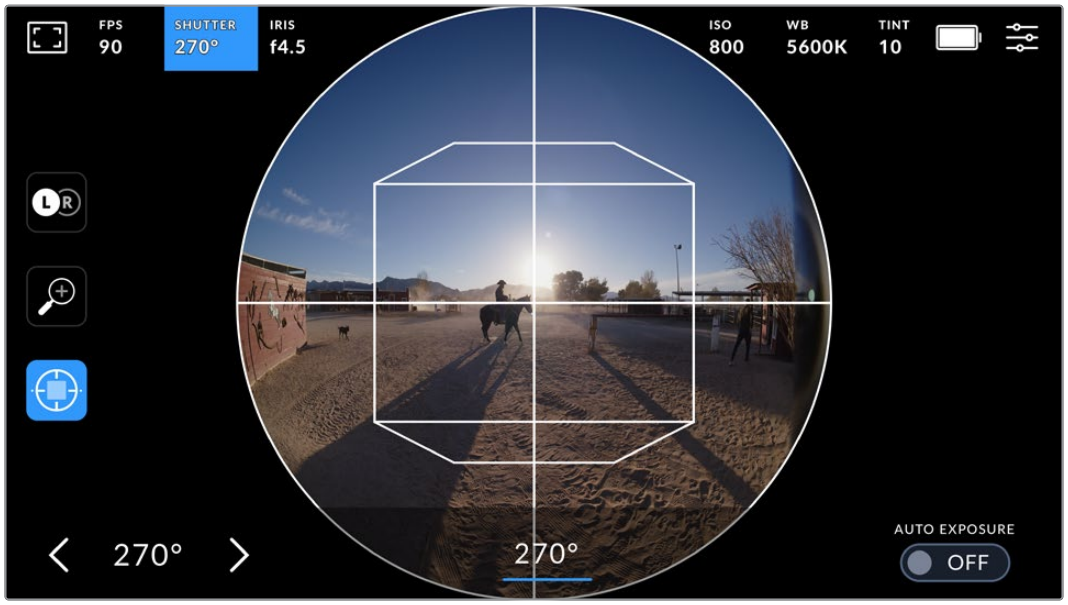
Apple Immersive Video has higher exposure requirements than traditional 24 fps capture and might require you to rethink your shooting and lighting approach. If you maintain a 180 degree shutter, 90 fps capture requires almost four times more light to expose to the same amount. In addition, the fixed iris of the lenses, which is around f4.5, is up to three stops darker than typical super speed lenses. This means you might need up to seven stops more light to achieve adequate exposure. However, if you are shooting at 90fps, using a wide shutter angle between 270° and 360° can help to compensate for high frame rate light loss while still achieving pleasing amounts of motion blur.

Shutter Speed

The ‘shutter’ indicator displays your shutter angle or shutter speed. The shutter measurement setting in the ‘setup’ menu can be used to select whether to display shutter information as ‘shutter angle’ or ‘shutter speed’.

To change the shutter speed:

1. Tap the ‘shutter’ icon in the upper toolbar.
2. Use the left or right arrows to cycle through common shutter speed presets. If you would like to choose a specific shutter value, you can do so by double tapping the current shutter indicator at the bottom left of your screen. This opens a keypad that allows you to type in any shutter value with up to two decimal places between 11.2 and 360 degrees.



Custom frame rates can be entered using the onscreen keypad.

Shutter speed has less of an impact on the appearance of high frame immersive content. While a 180 degree shutter is still favored by many for shooting at 24 or 30p, it’s more common to see shutter speed used to aid exposure when shooting for immersive.

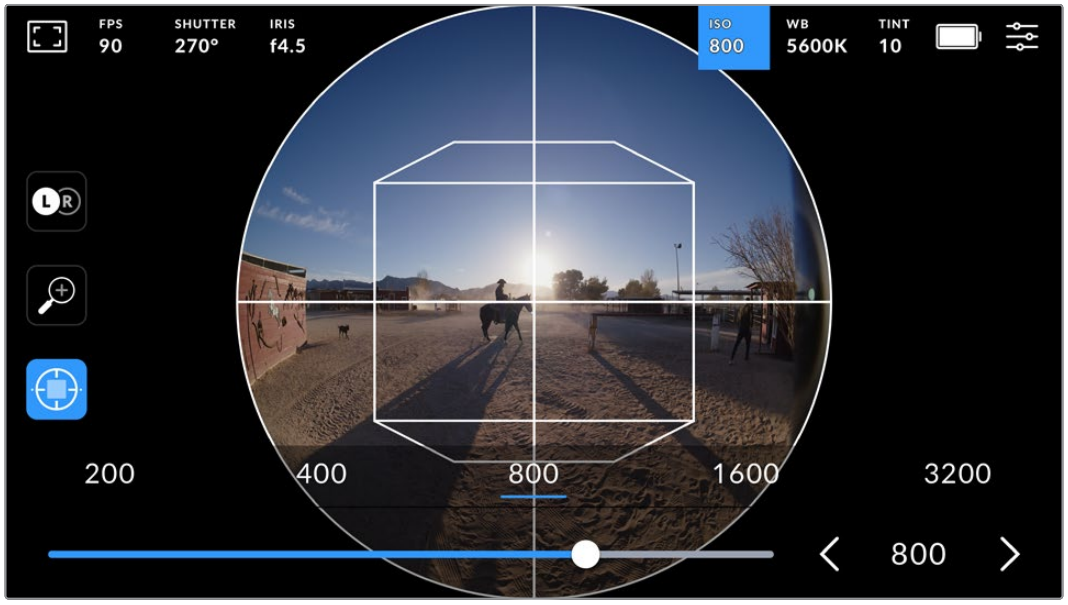
When shooting in low light, the shutter can be increased to 270 degrees or, in some exceptional circumstances, 360 degrees.

When shooting in bright conditions, the shutter speed can be lowered as far as 11.2 degrees. However, in bright conditions, it is more common to leave the shutter speed at your preferred setting for motion and use ND filters to control exposure.

It's worth noting that Apple Immersive Video's high frame rate of 90 frames per second can cause issues when shooting with flickering lights. A 270 degree shutter is suggested when working in 60hz environments and a 324 degree shutter is recommended when working in 50hz environments.

ISO and Dynamic Range

The 'ISO' indicator displays your camera's current ISO setting, or light sensitivity. Tapping this indicator lets you adjust your ISO to suit varying lighting conditions. The ISO range on URSA Cine Immersive is from ISO 200 to 3200. The optimum ISO is 800.



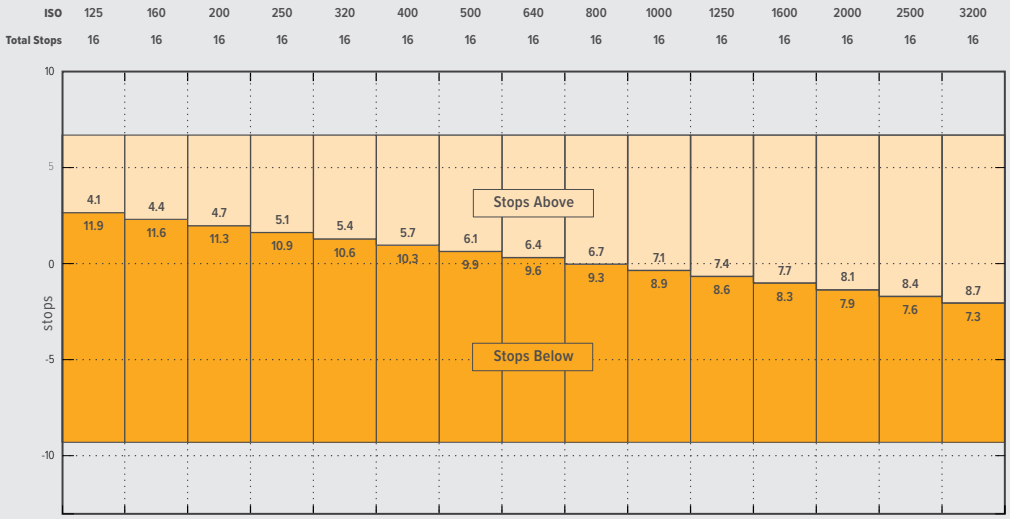
The iso can be configured by tapping the iso indicator.

Numerical presets provide the option to select an ISO setting in full stop increments by tapping on the number. The slider below the presets allows you to fine tune your ISO setting in between these presets in 1/3 stop increments.

Depending on your situation, you may choose a lower or higher ISO setting. For example, in low light conditions ISO 1600 can be suitable but may introduce some visible noise. In bright conditions ISO 200 can provide richer colors.

Blackmagic cameras are able to capture in wide dynamic range which gives you lots of flexibility when capturing images and also when color grading in DaVinci Resolve. This table shows the available dynamic range in your URSA Cine Immersive.

URSA Cine Immersive Dynamic Range



ND

The camera's ND filters are typically the most common way of controlling exposure, allowing you to reduce the amount of light reaching your camera's sensor by a preset number of exposure 'stops'.

The URSA Cine Immersive's ND are adjusted using the buttons on the left hand side above the fold out touchscreen. Together with a clear filter, the available options are '2', '4', '6' and '8' stops.

The current ND setting is displayed via the indicator in the top left of the touchscreen. When ND is engaged, this indicator remains on permanently. When ND is set to clear the indicator will disappear after 4 seconds.

NOTE You can change the ND filter terminology to reflect the conventions you're used to. Use the setup menu to select between ND number, stops and fractions.

Off Speed Recording

The project frame rate in the setup menu is URSA Cine Immersive's recording format frame rate and defines the speed at which your footage will play back. This is normally set to match your playback speed and audio sync used in your post production workflow and delivery requirements.

Tap your URSA Cine Immersive's frames per second indicator to access frame rate settings. URSA Cine Immersive has 8 project frame rate settings including 23.98, 24, 25, 29.97, 30, 50, 59.94, 60 and 90 frames per second. When shooting Immersive, the project frame rate should be set to 90 frames per second.

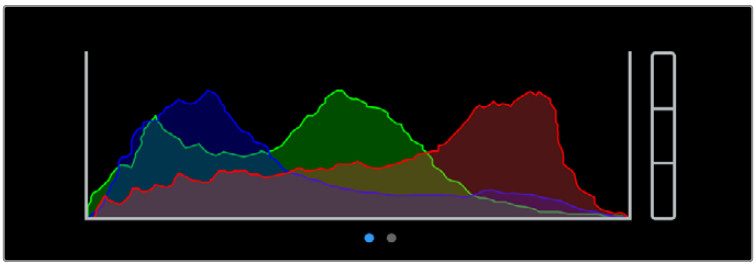
The sensor frame rate allows you to define the speed at which your sensor captures an image. By default, your URSA Cine Immersive's project and sensor frame rates are matched for a natural playback speed. However, by tapping the 'off speed frame rate' switch icon in the bottom right hand side of your camera's 'FPS' menu, you can independently set your sensor frame rate.

When the off speed frame rate switch is enabled, you can manually set the sensor frame rate. It's worth noting that 90 frames per second is the camera's highest frame rate and will match the project frame rate of 90 fps when filming for Apple Vision Pro. However, selecting a lower project frame rate like 30 or 60 frames and then shooting off speed at 90 frames per second will let you create beautiful slow motion footage.

While the Off Speed Frame Rate switch is engaged, the frame rate indicator in the top left of the screen will show the sensor frame rate followed by the project frame rate.

Histogram

At the bottom left of your URSA Cine Immersive touchscreen, you'll see the histogram. A histogram is a graph that shows the brightness distribution in an image. The left side represents shadows, the middle shows midtones, and the right shows highlights. The height of the graph indicates how much of the image is at that brightness level.



The histogram gives you an indication of the tonal distribution in your clip at a glance. An image where the graph sits on the left hand side is likely underexposed. Similarly, an image where the graph sits on the right hand side is likely overexposed. The histogram should always be interpreted in context with the image, but in general, a well exposed image has a balanced graph that sits in the middle. Increasing exposure will move the histogram to the right, reducing exposure will move it to the left.

To the right of the histogram is an indicator that shows when individual color channels clip. Typically, you should avoid clipping any channels. There are, however, scenarios when it's acceptable to clip your image if it helps you to preserve the image's midtones and shadows.

Expose to the Right is a technique where you deliberately overexpose an image by pushing the histogram as far to the right as possible without clipping color channels. This maximizes signal and minimizes noise. As long as the exposure is lowered in post, this is an excellent way of producing clean images!

LCD Monitor Options

Tap the monitor icon at the top left of your URSA Cine Immersive's LCD touchscreen to access the LCD monitor settings. This menu contains useful options like monitor brightness. It also contains powerful exposure tools.

Zebra

The 'zebra' setting toggles the appearance of zebra on the touchscreen, as well as setting the zebra level. Zebra displays diagonal lines over areas of your image that exceed a set exposure level. The percentage setting is used to show parts of your image that are overexposed. It's common to set this to a value like 95% rather than 100% as it allows you to identify parts of the image that are curable to over exposure. The second zebra setting lets you turn mid gray zebra zones on or off, or mid gray plus 1 stop.

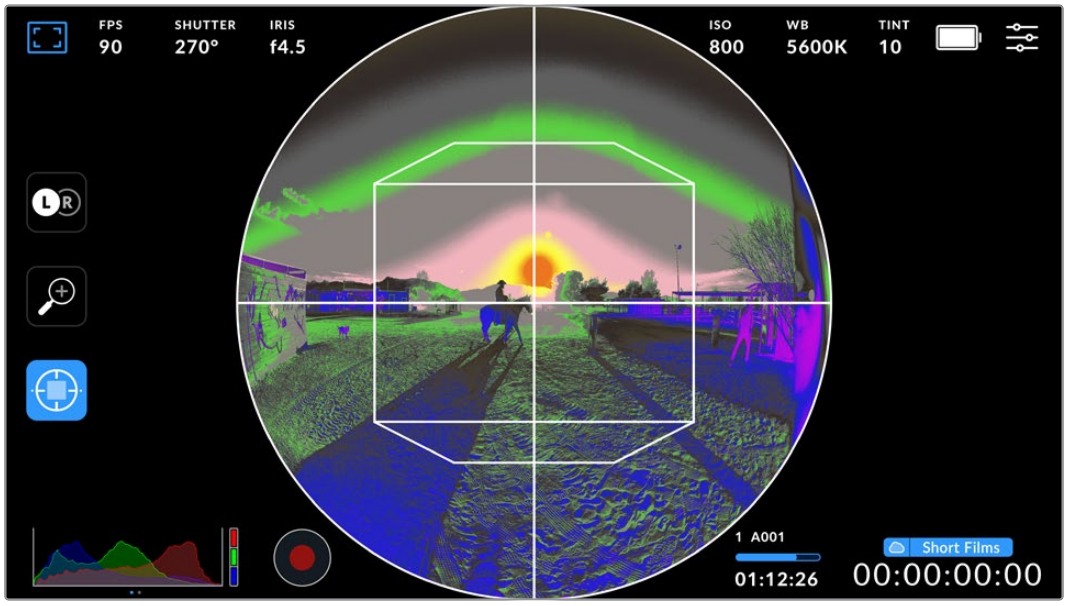
Grids

The ‘grids’ setting toggles the appearance of a rule of thirds grid, horizon meter, crosshair or center dot on the LCD touchscreen, as well as setting the overlay that will be visible on all URSA Cine Immersive’s outputs. When shooting immersive content the horizon meter is very useful. It displays a double crosshair in the centre of the image that is aligned as long as the camera is completely level, but diverges when the camera is not.

False Color

False Colour overlays different colors onto your image that represent exposure values for different elements in your image. For example, pink represents optimum exposure for lighter skin tones, while green is a good match to darker skin tones. By monitoring the pink or green false color when recording people, you can maintain consistent exposure for their skin tones.

Similarly, when elements in your image change from yellow to red, that means they are now overexposed. Similar to the histogram, this tool gives you a very quick and intuitive understanding of the image’s tonal distribution.



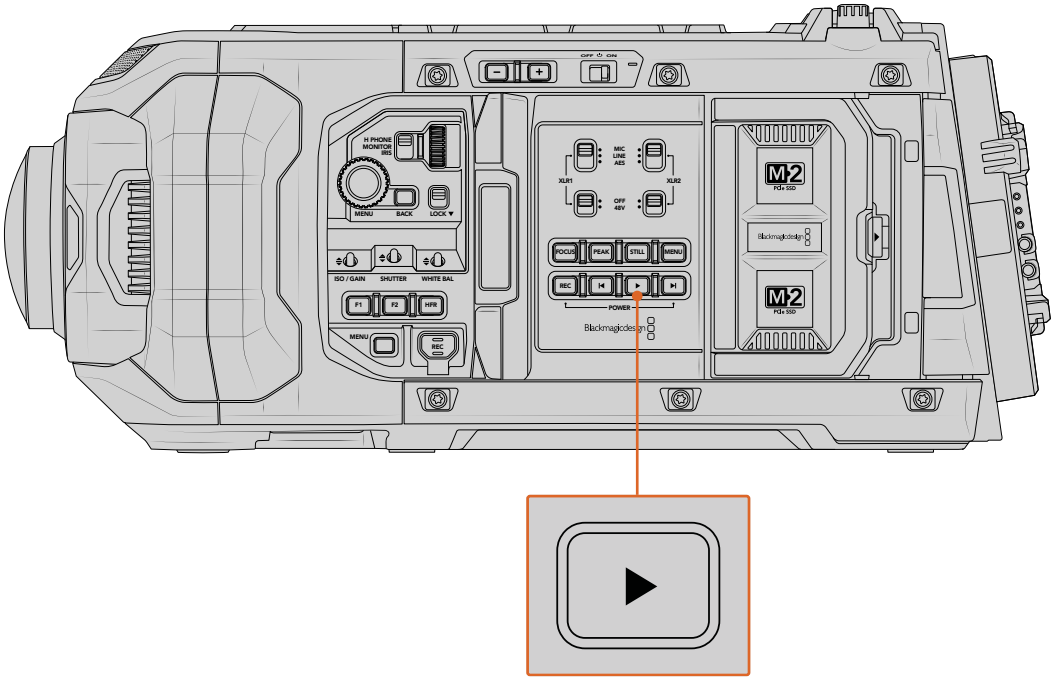
Recording

Start recording by pressing any of the red record buttons on your camera. Record buttons are located on the forward control panel, the internal control panel behind the fold out touchscreen, the assist station control panel and both LCD touchscreens. Press the record button again to stop recording.

The storage indicators show how much recording time is left. The time is displayed in hours:minutes:seconds based on your chosen frame rate. The media bar icon above the recording time will be either blue, white or red depending on its current status and will display the used space on the storage.

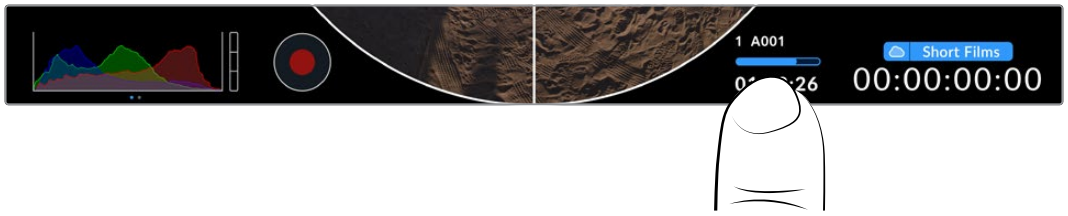
Playback

Once you have recorded your clips, you can use the transport control buttons to play them back. To play back the last clip that you recorded, push the physical play button on the left-hand side of the camera, located behind the fold out touch screen. The previous and next buttons can be used to navigate to other clips. To leave playback mode and ready the camera to continue recording, push the record button once.

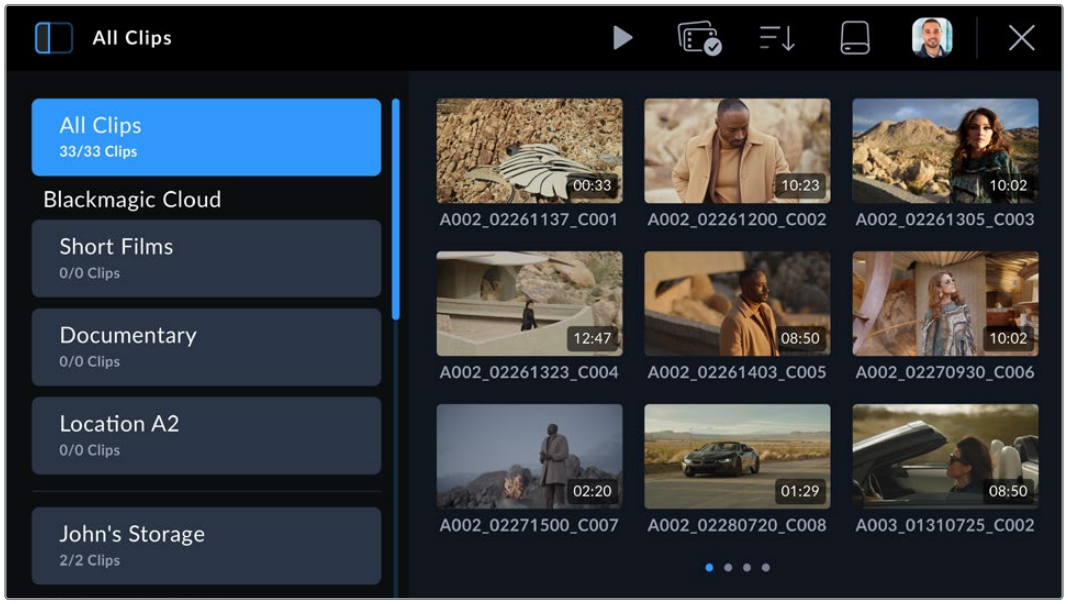


Press the forward and reverse buttons to skip to the start or end of clips. Press the ‘reverse skip’ button once to go to the start of the current clip or press twice to skip back to the start of the previous clip. Hold the ‘forward’ or ‘reverse skip’ button to play or reverse at 2x speed. Once shuttling forward or backwards, press the ‘fast forward’ or ‘reverse skip’ buttons twice for 4x, three times for x8 and four times for x16.

If a more extensive review of the footage you have shot is required, you will likely prefer to use the Media Pool. To open the media pool, tap one of the storage indicators at the bottom of your camera’s touchscreen display.



The main page browser displays thumbnails of all the recorded clips on all media connected to your camera. There are three rows of thumbnails and as more clips are added you can swipe to the next page of thumbnails using the touchscreen. Page indicators at the bottom of the display show how many pages of clips are available.



Connect the URSA Cine Immersive to a Blackmagic Cloud account and send footage directly to a DaVinci Resolve project.

If you wish to exit the media pool at any time, tap the X in the top right hand corner, or push any of the camera's record buttons once.

Transferring Clips

There are a number of ways you can transfer clips from your camera's storage. For example, you can plug URSA Cine Immersive into your computer or a network via a 10G Ethernet cable and copy clips from your camera's storage over Ethernet. You can even load the media module into a Blackmagic Media Dock connected to your network or computer, which lets you work with up to three media modules simultaneously.

When connecting to your computer over Ethernet, your computer will recognize the Blackmagic Media Module just like it would when plugging in a Blackmagic Cloud Store.

Your camera is set to use DHCP by default, so your computer or network will see the camera and assign an IP address automatically.

Enabling File Sharing

When you are setting your camera up for the first time and have connected to your computer or network via Ethernet, SMB file sharing will be disabled by default. This is because unlike a Blackmagic Cloud Store where you will likely want to share files with other users on the network straight away, you may not want to immediately allow the same level of access to the master camera files after an important day's shoot. For this reason, we decided to let you enable SMB file sharing yourself.

1. To enable file sharing, you will need to download and install Blackmagic Camera Setup on your computer. This setup utility is an administration software application that lets you update your camera and change admin settings. You can download Blackmagic Camera Setup from our website at: https://www.blackmagicdesign.com/support/family/professional-cameras
2. After downloading and installing the setup utility, connect your camera to the computer via USB and launch Blackmagic Camera Setup. Connect via the top panel USB port towards the rear of your camera.

3 To open the settings, click on the image of your URSA Cine Immersive or its settings icon.

4 Scroll down to the ‘network access’ settings and enable ‘file sharing (SMB)’. Then hit ‘save’ in the bottom right for these settings to take effect.

5 Now go to the URL setting underneath and click on the URL icon to automatically populate the address. The URL address will display your camera’s name as a local network location.

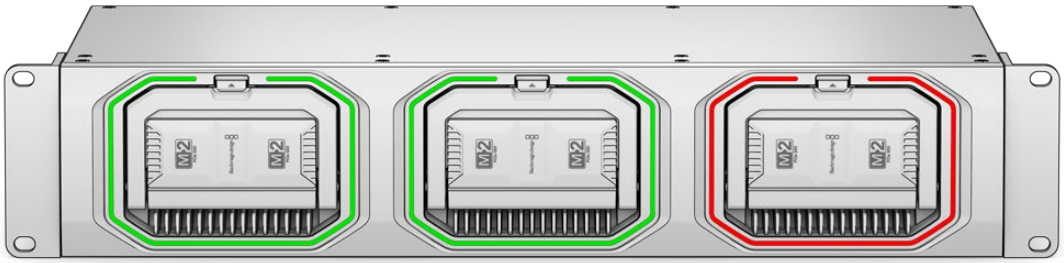
6 Click save.

To go straight to your clips, click on your camera’s URL name. A window will open asking you for a password or connect as a guest. Click on the ‘guest’ radio button and click ‘connect’.

7 Your computer will now automatically navigate to your media module in Finder on a Mac and Windows Explorer in Windows. The utility will remember your Ethernet setup so you only need to do this once when setting up for the first time and then the next time you plug in your camera via Ethernet, you will be able to access the media module straight away without touching any of these settings.

Using the Blackmagic Media Dock

The Blackmagic Media Module is designed to be quickly loaded and unloaded when you need to transfer clips to external storage or to swap to an empty module so you can keep recording. The Blackmagic Media Dock can be used to access footage on a Media Module without the camera.



Summary

Apple Immersive Video introduces a new approach to visual storytelling, and offers an alternative to traditional 2D cinema. It is not intended as a replacement but as an additional format that can be selected based on the needs of the project. As with any format, creative choices will vary depending on the story being told.

While new formats often involve complex workflows, URSA Cine Immersive is designed to simplify the process of capturing Apple Immersive Video. It provides a straightforward production path for immersive content and may also be used for visual effects and other applications.

This guide outlines the core features and workflows of the URSA Cine Immersive. It is intended as a starting point. As with any camera system, further exploration and testing are encouraged to achieve the best results for your specific production needs.

Developer Information

Camera Control REST API

If you are a software or hardware developer you can build custom applications or leverage ready to use tools such as REST client or Postman to seamlessly control and interact with your compatible Blackmagic camera using Camera Control REST API. This API enables you to perform a wide range of operations, such as starting or stopping recordings, accessing disk information and much more. Whether you're developing a custom application tailored to your specific needs or utilizing existing tools, this API empowers you to unlock the full potential of your Blackmagic camera with ease. We look forward to seeing what you come up with!

NOTE It's important to mention that controlling Blackmagic cameras via REST API relies on the web manager being enabled on each compatible Blackmagic camera. Enable the web media manager in the Blackmagic Camera Setup 'network access' settings for each camera you are controlling.

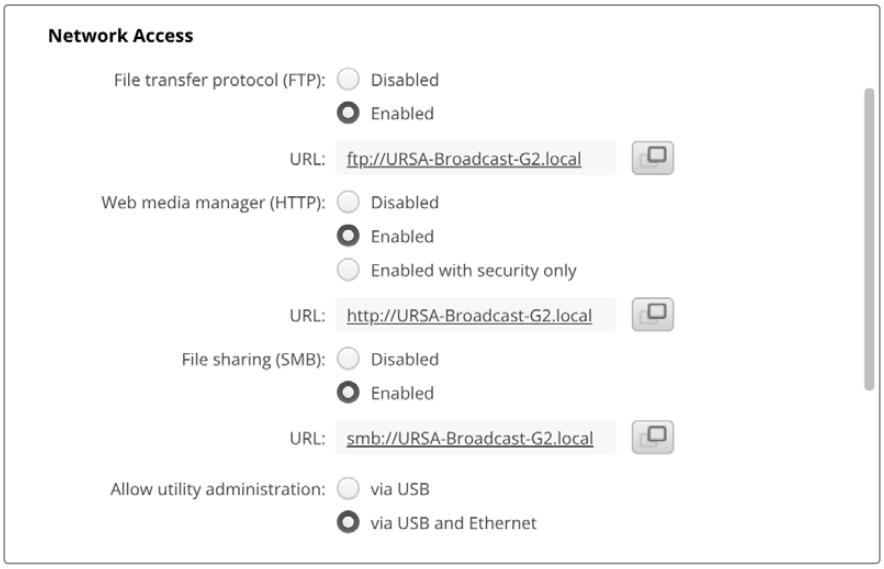
The following Blackmagic cameras are compatible with Camera Control REST API:

- Blackmagic URSA Cine 12K LF
- Blackmagic URSA Cine 17K 65
- Blackmagic URSA Cine Immersive
- Blackmagic PYXIS 6K
- Blackmagic Cinema Camera 6K
- Blackmagic URSA Broadcast G2
- Blackmagic Micro Studio Camera 4K G2
- Blackmagic Studio Camera 4K Plus
- Blackmagic Studio Camera 4K Pro
- Blackmagic Studio Camera 6K Pro
- Blackmagic Studio Camera 4K Plus G2
- Blackmagic Studio Camera 4K Pro G2

Sending API Commands

To send an API command to your camera from a third party application such as Postman, add /control/api/v1/ to the end of the camera's Web media manager URL or IP address. For example, https://ursa-broadcast-g2.local/control/api/v1/

You can find the Web media manager URL and IP address information in Blackmagic Camera Setup.



The Web media manager URL in Blackmagic Camera Setup

Downloading API's from your Camera

You can download REST API YAML documentation from your camera by adding /control/documentation.html to the end of the camera's Web media manager URL or IP address. For example, https://ursa-broadcast-g2.local/control/documentation.html

NOTE It's worth noting that changing the camera name in Blackmagic Camera Setup will also change the camera's Web media manager URL.

Livestream Control API

API for controlling Livestreams on Blackmagic Design products.

GET /livestreams/0

Get the livestream's current status.

Response

200 - Livestream's current status.

The response is JSON.

Name	Type	Description
status (required)	string	Possible values are: Idle, Connecting, Streaming, Flushing, Interrupted.
bitrate (required)	integer	Current bitrate (bps).
effectiveVideoFormat (required)	string	Effective video format for the livestream, serialised as a string.
duration	integer	Current stream duration in seconds. Absent if livestream is idle.
cache	integer	Current stream cache usage percentage.

GET /livestreams/0/start

Determine if the livestream is active.

Response

200 - Livestream active status.

The response is JSON.

Name	Type	Description
	boolean	True when the livestream is active.

PUT /livestreams/0/start

Start the livestream.

Response

204 - Livestream started.

GET /livestreams/0/stop

Determine if the livestream is inactive.

Response

200 - Livestream inactive status.

The response is JSON.

Name	Type	Description
	boolean	True when the livestream is inactive.

PUT /livestreams/0/stop

Stop the livestream.

Response

204 - Livestream stopped.

GET /livestreams/0/activePlatform

Get the currently selected platform configuration for the livestream.

Response

200 - Livestream active platform configuration.

The response is JSON.

Livestream's current active platform configuration.

Name	Type	Description
platform (required)	string	Platform name.
server (required)	string	The platform's server name, or "Custom" when the URL is customizable.
key	string	Stream key. Assumed to be empty if missing.
passphrase	string	Passphrase. Only included for SRT streams.
quality (required)	string	Quality level name.
url	string	Livestream destination. Only included when URL is customizable.

PUT /livestreams/0/activePlatform

Set the currently selected platform configuration for the livestream.

Parameters

Livestream's current active platform configuration.

Name	Type	Description
platform (required)	string	Platform name.
server (required)	string	The platform's server name, or "Custom" when the URL is customizable.
key	string	Stream key. Assumed to be empty if missing.
passphrase	string	Passphrase. Only included for SRT streams.
quality (required)	string	Quality level name.
url	string	Livestream destination. Only included when URL is customizable.

Response

204 - Livestream active platform configuration updated.

400 - Bad Request

GET /livestreams/platforms

Get the list of available platforms.

Response

200 - List of available platforms.

The response is JSON.

Name	Type	Description
	array	List of available platforms names.
[i]	string	Platform name.

GET /livestreams/platforms/{platformName}

Get the service configuration for a platform.

Parameters

Name	Type	Description
{platformName} (required)	string	Name of the platform.

Response

200 - Service configuration for specified platform.

The response is JSON.

Livestream platform service configuration.

Name	Type	Description
platform (required)	string	Corresponding platform name.
key	string	Default stream key.
servers (required)	array	List of server configurations.
servers[i]	object	Server configuration.
servers[i].server (required)	string	Server name.
servers[i].url (required)	string	Livestream destination.
servers[i].srtExtensions	array	Miscellaneous tags used for SRT livestreams.
servers[i].srtExtensions[i]	object	Dictionary object mapping SRT tag strings to values.
servers[i].srtExtensions[i][{key}]	string	SRT tag value.
servers[i].group	string	Logical grouping of the server.
profiles (required)	array	List of profile configurations.
profiles[i]	object	Quality configuration.
profiles[i].profile (required)	string	Quality level name.
profiles[i].configs (required)	array	List of video format configurations.
profiles[i].configs[i]	object	Video format configuration for profiles.
profiles[i].configs[i].resolution (required)	string	Video format serialised as a string.
profiles[i].configs[i].fps (required)	string	Frames per second.
profiles[i].configs[i].bitrate (required)	integer	Pixel bitrate (bps).
profiles[i].configs[i].audioBitrate	integer	Audio bitrate (bps).
profiles[i].configs[i].keyFrameInterval	integer	How often a key frame is sent, in seconds.
profiles[i].configs[i].videoCodecs	array	Supported video encoding algorithm/s.

Name	Type	Description
profiles[i].configs[i].videoCodecs[i]	string	Video encoding algorithm. Possible values are: H264, H265.
profiles[i].lowLatency (required)	boolean	If true, fewer frames will be buffered in the livestream.
defaultProfile	string	Quality level name.
credentials	object	Credientials used for RTMP streams.
credentials.username (required)	string	The username part of the creditials. Only used for RTMP streams.
credentials.password (required)	string	Used for RTMP streams, also used as Passphrase for SRT streams.
customizableUrlEnabled	boolean	True when the server URL is customizable.

400 - Bad Request

GET /livestreams/customPlatforms

Get a list of custom platform files.

Response

200 - List of custom platform files.

The response is JSON.

Name	Type	Description
	array	List of custom platform file names.
[i]	string	Custom platform file name.

DELETE /livestreams/customPlatforms

Remove all custom configuration files.

Response

204 - All custom configuration files removed.

GET /livestreams/customPlatforms/{filename}

Get a custom platform file.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to get.

Response

200 - Custom platform file.

The response is XML.

Blackmagic streaming XML file format.

Name	Type	Description
	object	Blackmagic streaming XML file format.

404 - Not Found

PUT /livestreams/customPlatforms/{filename}

Update a custom platform file if it exists, if not, create a new file with the given file name.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to update/create.

Blackmagic streaming XML file format.

Name	Type	Description
	object	Blackmagic streaming XML file format.

Response

204 - Custom platform file created or updated.

400 - Bad Request

DELETE /livestreams/customPlatforms/{filename}

Remove the given custom platform file.

Parameters

Name	Type	Description
{filename} (required)	string	Name of the file to be removed.

Response

204 - Custom platform file removed.

404 - Not Found

Clips Control API

API for listing clips on disk.

GET /clips

Get the list of clips on the active disk.

Response

200 - List of clips on the active disk.

The response is JSON.

List of media clips.

Name	Type	Description
clips (required)	array	
clips[i]	object	Media clip.
clips[i].clipUniqueId (required)	integer	Unique ID used to identify this clip.
clips[i].filePath	string	Path to the file relative to the root of a mount.
clips[i].fileSize	integer	Size of file on disk in bytes.
clips[i].codecFormat	object	
clips[i].codecFormat.codec	string	Currently selected codec.

Name	Type	Description
clips[i].codecFormat.container	string	Multimedia container format.
clips[i].videoFormat	object	Video format configuration.
clips[i].videoFormat.name (required)	string	Video format serialised as a string.
clips[i].videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
clips[i].videoFormat.height	number	Height dimension of video format.
clips[i].videoFormat.width	number	Width dimension of video format.
clips[i].videoFormat.interlaced	boolean	Is the display format interlaced?
clips[i].startTimecode	string	Start timecode of the clip serialised as string.
clips[i].durationTimecode	string	Duration of the clip in timecode format serialised as string.
clips[i].frameCount	integer	Number of frames in clip; duration of the clip in frames.

404 - There is no active disk.

Media Pool Control API

API to manage media pool and handle uploads and project data.

GET /cloud/projects

List all projects within the media pool.

Response

200 - Successfully retrieved the list of all projects.

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].libraryID	string	
[i].name	string	
[i].private	boolean	
[i].shared	boolean	
[i].clips	array	List of clips associated with the project.
[i].clips[i]	string	
[i].status	object	
[i].status.numClipsRequested	integer	
[i].status.numClipsComplete	integer	
[i].status.uploadPercent	integer	Percentage of upload completion.
[i].status.numClipsPaused	integer	
[i].status.outOfSpace	boolean	
[i].status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
[i].status.currentByteRate	integer	Current byte rate of the upload process.

GET /cloud/projects/active

Retrieve data of the actively uploading project.

Response

200 - Successfully retrieved the active project's data.

The response is JSON.

Name	Type	Description
libraryID	string	
name	string	
private	boolean	
shared	boolean	
clips	array	List of clips associated with the project.
clips[i]	string	
status	object	
status.numClipsRequested	integer	
status.numClipsComplete	integer	
status.uploadPercent	integer	Percentage of upload completion.
status.numClipsPaused	integer	
status.outOfSpace	boolean	
status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
status.currentByteRate	integer	Current byte rate of the upload process.

GET /cloud/projects/{projectID}

Retrieve specific project data by project ID.

Parameters

Name	Type	Description
{projectID} (required)	integer	Unique identifier of the project.

Response

200 - Successfully retrieved the project data.

The response is JSON.

Name	Type	Description
libraryID	string	
name	string	
private	boolean	
shared	boolean	
clips	array	List of clips associated with the project.
clips[i]	string	
status	object	
status.numClipsRequested	integer	
status.numClipsComplete	integer	
status.uploadPercent	integer	Percentage of upload completion.
status.numClipsPaused	integer	

Name	Type	Description
status.outOfSpace	boolean	
status.secsRemaining	integer	Estimated seconds remaining until upload is completed.
status.currentByteRate	integer	Current byte rate of the upload process.

404 - Project not found.

GET /cloud/clips

List all clips within the media pool.

Response

200 - Successfully retrieved the list of all clips.

The response is JSON.

Name	Type	Description
	array	
[i]	string	

GET /cloud/clips/activeUploading

Retrieve data of actively uploading clips.

Response

200 - Successfully retrieved the list of actively uploading clips.

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].path	string	
[i].projectID	integer	
[i].status	object	
[i].status.projectID	integer	
[i].status.outOfSpace	boolean	
[i].status.proxyExtension	string	
[i].status.growingFile	boolean	
[i].status.originalUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
[i].status.proxyUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
[i].status.originalClipTotalSize	integer	
[i].status.proxyClipTotalSize	integer	
[i].status.originalClipCompletedSize	integer	
[i].status.proxyClipCompletedSize	integer	
[i].status.secsRemaining	integer	

GET /cloud/clips/{deviceName}/{path}

Retrieve specific clip data by device and path.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored.
{path} (required)	string	Path to the clip.

Response

200 - Successfully retrieved the clip data.

The response is JSON.

Name	Type	Description
path	string	
projectID	integer	
status	object	
status.projectID	integer	
status.outOfSpace	boolean	
status.proxyExtension	string	
status.growingFile	boolean	
status.originalUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
status.proxyUploadState	string	Possible values are: Unqueued, Paused, Queued, Uploading, Uploaded, Failed.
status.originalClipTotalSize	integer	
status.proxyClipTotalSize	integer	
status.originalClipCompletedSize	integer	
status.proxyClipCompletedSize	integer	
status.secsRemaining	integer	

404 - Clip not found.

Monitoring Control API

API for monitoring and controlling display settings in video equipment.

GET /monitoring/display

Retrieve a list of all display names.

Response

200 - Returns a list of display names.

The response is JSON.

Name	Type	Description
displays	array	List of display names available.
displays[i]	string	

GET /monitoring/{displayName}/cleanFeed

Get the clean feed enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/cleanFeed

Set the clean feed enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Clean feed enabled/disabled successfully.

400 - Invalid input.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/displayLUT

Get the display LUT enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

400 - Invalid display name.

404 - Display name not found.

PUT /monitoring/{displayName}/displayLUT

Set the display LUT enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Display LUT enabled/disabled successfully.

400 - Invalid input.

422 - Unprocessable Entity - Unable to process the contained instructions.

GET /monitoring/{displayName}/zebra

Get the zebra enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/zebra

Set the zebra enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Zebra enabled/disabled successfully.

400 - Invalid input.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/focusAssist

Get the focus assist enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display name not found.

PUT /monitoring/{displayName}/focusAssist

Set the focus assist enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

Response

204 - Focus assist settings updated successfully.

400 - Invalid input or configuration.

422 - Unable to process the contained instructions.

GET /monitoring/focusAssist

Get the focus assist settings.

Response

200 - OK

The response is JSON.

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

404 - Display name not found.

PUT /monitoring/focusAssist

Set the focus assist settings.

Parameters

Name	Type	Description
mode	string	Mode of focus assist, e.g., 'Peak' or 'ColoredLines'. Possible values are: Peak, ColoredLines.
color	string	Color of the focus assist highlight. Possible values are: Red, Green, Blue, White, Black.
intensity	integer	Intensity of the focus assist highlight (0-100).

Response

204 - Focus assist settings updated successfully.

400 - Invalid input or configuration.

422 - Unable to process the contained instructions.

GET /monitoring/{displayName}/frameGuide

Get the frame guide enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the frame guide enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/frameGuide

Set the frame guide enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Frame guide state updated successfully.

422 - Unable to update the frame guide state.

GET /monitoring/frameGuideRatio

Get the current frame guide ratio.

Response

200 - Returns the current frame guide ratio.

The response is JSON.

Name	Type	Description
ratio	string	The frame guide ratio.

PUT /monitoring/frameGuideRatio

Set the frame guide ratio.

Parameters

Name	Type	Description
ratio	string	The frame guide ratio.

Response

204 - Frame guide ratio updated successfully.

422 - Unable to update the frame guide ratio.

GET /monitoring/frameGuideRatio/presets

Get the presets for frame guide ratios.

Response

200 - Returns a list of preset frame guide ratios.

The response is JSON.

Name	Type	Description
presets	array	
presets[i]	string	A frame guide ratio.

GET /monitoring/{displayName}/frameGrids

Get the frame grids enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the frame grids enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/frameGrids

Set the frame grids enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Frame grids state updated successfully.

422 - Unable to update the frame grids state.

GET /monitoring/frameGrids

Get the global frame grids settings.

Response

200 - Returns the current frame grids settings.

The response is JSON.

Name	Type	Description
frameGrids	array	List of frame grids enabled.
frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

PUT /monitoring/frameGrids

Set the global frame grids settings.

Parameters

Name	Type	Description
frameGrids	array	List of frame grids enabled.
frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

Response

204 - Frame grids settings updated successfully.

400 - Invalid input, check the number of frame grids or values.

422 - Unable to update the frame grids settings.

GET /monitoring/{displayName}/safeArea

Get the safe area enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the safe area enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/safeArea

Set the safe area enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - Safe area state updated successfully.

422 - Unable to update the safe area state.

GET /monitoring/safeAreaPercent

Get the current safe area percentage.

Response

200 - Returns the current safe area percentage.

The response is JSON.

Name	Type	Description
percent	integer	Safe area coverage percentage.

PUT /monitoring/safeAreaPercent

Set the safe area percentage.

Parameters

Name	Type	Description
percent	integer	Safe area coverage percentage to set.

Response

204 - Safe area percentage updated successfully.

400 - Invalid percentage value.

422 - Unable to update the safe area percentage.

GET /monitoring/{displayName}/falseColor

Get the false color enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Response

200 - Returns the false color enable state.

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

404 - Display not found.

PUT /monitoring/{displayName}/falseColor

Set the false color enable state for a specific display.

Parameters

Name	Type	Description
{displayName} (required)	string	Name of the display. Obtainable from /monitoring/display which returns a list of displayNames.

Name	Type	Description
enabled	boolean	Indicates if the feature is enabled.

Response

204 - False color state updated successfully.

422 - Unable to update the false color state.

Event Control API

API For working with built-in websocket.

GET /event/list

Get the list of events that can be subscribed to using the websocket API.

Response

200 - Websocket API events list.

The response is JSON.

Name	Type	Description
events	array	List of events that can be subscribed to using the websocket API.
events[i]	string	

System Control API

API for controlling the System Modes on Blackmagic Design products.

GET /system

Get device system information.

Response

200 - System summary.

The response is JSON.

The properties will be populated only with the values that are supported/implemented by the device in use.

Name	Type	Description
codecFormat	object	Codec format configuration.
codecFormat.codec	string	Codec serialised as string.
codecFormat.container	string	Multimedia container format.
videoFormat	object	Video format configuration.
videoFormat.name (required)	string	Video format serialised as a string.
videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
videoFormat.height	number	Height dimension of video format.
videoFormat.width	number	Width dimension of video format.
videoFormat.interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

GET /system/product

Get device product information.

Response

200 - Device product information.

The response is JSON.

Product information.

Name	Type	Description
deviceName	string	Name of device as displayed in Setup.
productName	string	Device's product name.
softwareVersion	string	Software version running on device.

501 - This functionality is not implemented for the device in use.

GET /system/supportedCodecFormats

Get the list of supported codecs.

Response

200 - List of supported codec formats.

The response is JSON.

Name	Type	Description
codecs	array	
codecs[i]	object	Codec format configuration.
codecs[i].codec	string	Codec serialised as string.
codecs[i].container	string	Multimedia container format.

501 - This functionality is not implemented for the device in use.

GET /system/codecFormat

Get the currently selected codec.

Response

200 - Current codec format.

The response is JSON.

Codec format configuration.

Name	Type	Description
codec	string	Codec serialised as string.
container	string	Multimedia container format.

501 - This functionality is not implemented for the device in use.

PUT /system/codecFormat

Update the system codec.

Parameters

Codec format configuration.

Name	Type	Description
codec	string	Codec serialised as string.
container	string	Multimedia container format.

Response

204 - The codec updated successfully.

400 - The specified codec format is unsupported.

501 - This functionality is not implemented for the device in use.

GET /system/videoFormat

Get the currently selected video format.

Response

200 - Current system video format.

The response is JSON.

Video format configuration.

Name	Type	Description
name (required)	string	Video format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
height	number	Height dimension of video format.
width	number	Width dimension of video format.
interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

PUT /system/videoFormat

Set the system video format.

Parameters

This parameter can be one of the following types:

Name	Type	Description
name (required)	string	Video format serialised as a string.

Name	Type	Description
frameRate (required)	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
height (required)	number	Height dimension of video format.
width (required)	number	Width dimension of video format.
interlaced	boolean	Is the display format interlaced?

Response

204 - The video format updated successfully.

400 - Invalid request.

409 - Operation unsupported in the current state.

501 - This functionality is not implemented for the device in use.

GET /system/supportedVideoFormats

Get the list of supported video formats for the current system state.

Response

200 - List of supported video formats.

The response is JSON.

List of supported video formats.

Name	Type	Description
formats	array	List of video formats.
formats[i]	object	Video format configuration.
formats[i].name (required)	string	Video format serialised as a string.
formats[i].frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
formats[i].height	number	Height dimension of video format.
formats[i].width	number	Width dimension of video format.
formats[i].interlaced	boolean	Is the display format interlaced?

501 - This functionality is not implemented for the device in use.

GET /system/supportedFormats

Get supported formats.

Response

200 - List of supported formats.

The response is JSON.

Name	Type	Description
supportedFormats	array	
supportedFormats[i]	object	
supportedFormats[i].codecs	array	
supportedFormats[i].codecs[i]	string	
supportedFormats[i].frameRates	array	
supportedFormats[i].frameRates[i]	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
supportedFormats[i]. maxOffSpeedFrameRate	number	
supportedFormats[i]. minOffSpeedFrameRate	number	
supportedFormats[i]. recordResolution	object	
supportedFormats[i]. recordResolution.height	number	Height of the resolution.
supportedFormats[i]. recordResolution.width	number	Width of the resolution.
supportedFormats[i]. sensorResolution	object	
supportedFormats[i]. sensorResolution.height	number	Height of the resolution.
supportedFormats[i]. sensorResolution.width	number	Width of the resolution.

501 - This functionality is not implemented for the device in use.

GET /system/format

Get current format.

Response

200 - Current format.

The response is JSON.

Name	Type	Description
codec	string	Codec format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
maxOffSpeedFrameRate	number	
minOffSpeedFrameRate	number	
offSpeedEnabled	boolean	
offSpeedFrameRate	number	
recordResolution	object	
recordResolution.height	number	Height of the resolution.
recordResolution.width	number	Width of the resolution.
sensorResolution	object	
sensorResolution.height	number	Height of the resolution.
sensorResolution.width	number	Width of the resolution.

501 - This functionality is not implemented for the device in use.

PUT /system/format

Set the format.

Parameters

Name	Type	Description
codec	string	Codec format serialised as a string.
frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
maxOffSpeedFrameRate	number	
minOffSpeedFrameRate	number	
offSpeedEnabled	boolean	
offSpeedFrameRate	number	
recordResolution	object	
recordResolution.height	number	Height of the resolution.
recordResolution.width	number	Width of the resolution.
sensorResolution	object	
sensorResolution.height	number	Height of the resolution.
sensorResolution.width	number	Width of the resolution.

Response

204 - System format updated.

501 - This functionality is not implemented for the device in use.

Transport Control API

API for controlling Transport on Blackmagic Design products.

GET /transports/0

Get device's basic transport status.

Response

200 - Transport status.

The response is JSON.

Name	Type	Description
mode	string	Transport mode. Possible values are: InputPreview, InputRecord, Output.

PUT /transports/0

Set device's basic transport status.

Parameters

Name	Type	Description
mode	string	Transport mode. Possible values are: InputPreview, Output.

Response

204 - Transport mode was set.

400 - Failed to set transport mode.

GET /transports/0/stop

Determine if transport is stopped.

Response

200 - Transport stop response.

The response is JSON.

Name	Type	Description
	boolean	True when transport mode is InputPreview or when in Output mode and speed is 0.

PUT /transports/0/stop

Stop transport. Deprecated, use POST /transports/0/stop instead.

Response

204 - Transport stopped.

POST /transports/0/stop

Stop transport.

Response

204 - Transport stopped.

GET /transports/0/play

Determine if transport is playing.

Response

200 - Transport play response.

The response is JSON.

Name	Type	Description
	boolean	True when transport is in Output mode and speed is non-zero.

PUT /transports/0/play

Start playing on transport. Deprecated, use POST /transports/0/play instead.

Response

204 - Transport playing.

400 - Failed to set transport to play.

POST /transports/0/play

Start playing on transport.

Response

204 - Transport playing.

400 - Failed to set transport to play.

GET /transports/0/playback

Get playback state.

Response

200 - Transport playback state.

The response is JSON.

Name	Type	Description
type	string	Possible values are: Play, Jog, Shuttle, Var.
loop	boolean	When true, playback loops from the end of the timeline to the beginning of the timeline.
singleClip	boolean	When true, playback loops from the end of the current clip to the beginning of the current clip.
speed	number	Playback speed, 1.0 for normal forward playback.
position	integer	Playback position on the timeline in units of video frames, where 0 is the first frame of the timeline.

PUT /transports/0/playback

Set playback state.

Parameters

Name	Type	Description
type	string	Possible values are: Play, Jog, Shuttle, Var.
loop	boolean	When true, playback loops from the end of the timeline to the beginning of the timeline.
singleClip	boolean	When true, playback loops from the end of the current clip to the beginning of the current clip.
speed	number	Playback speed, 1.0 for normal forward playback.
position	integer	Playback position on the timeline in units of video frames, where 0 is the first frame of the timeline.

Response

204 - Updated transport playback state.

400 - Failed to set transport playback state.

GET /transports/0/record

Get record state.

Response

200 - Recording state.

The response is JSON.

Name	Type	Description
recording	boolean	If true, transport is in InputRecord mode.

PUT /transports/0/record

Set record state. Deprecated, use POST /transports/0/record instead.

Parameters

Name	Type	Description
recording (required)	boolean	If true, starts a recording, otherwise stops.
clipName	string	Optional, sets the requested clip name to record to, when “recording” attribute is set to true.

Response

204 - Recording state updated.

400 - Failed to update recording state.

POST /transports/0/record

Start recording.

Parameters

Name	Type	Description
clipName	string	Optional, provides a specific name of clip to record to.

Response

204 - Recording started.

400 - Failed to start recording.

GET /transports/0/clipIndex

Get the clip index of the currently playing clip on the timeline.

Response

200 - Clip index response.

The response is JSON.

Name	Type	Description
clipIndex	number | null	The 0-based index of the clip being played on the timeline. null when there is no timeline or an empty timeline.

GET /transports/0/timecode

Get device timecode.

Response

200 - Timecode response.

The response is JSON.

Name	Type	Description
display	string	The display timecode serialised as a string.
timeline	string	The timeline timecode serialised as a string.

GET /transports/0/timecode/source

Get timecode source selected on device.

Response

200 - Timecode source response.

The response is JSON.

Name	Type	Description
timecode	string	Possible values are: Timeline, Clip.

Timeline Control API

API for controlling playback timeline.

GET /timelines/0

Get the playback timeline.

Response

200 - Playback timeline.

The response is JSON.

Name	Type	Description
clips	array	
clips[i]	object	Timeline clip.
clips[i].clipUniqueId (required)	integer	Unique identifier used to identify this media clip. If the same media clip is added to the timeline multiple times, each timeline clip has the same clipUniqueId
clips[i].frameCount	integer	Duration of timeline clip in frames, the number of frames in this clip on the timeline.
clips[i].durationTimecode	string	Duration of the timeline clip in timecode format serialised as string. This will differ to durationTimecode reported in /clips for this clipUniqueId if clipIn or frameCount was specified when adding this clip to the timeline.
clips[i].clipIn	string	In frame offset for the clip on the timeline, where 0 is the first frame of the on-disk clip.
clips[i].inTimecode	string	Clip timecode of the first frame of this timeline clip serialised as string (clip startTimecode + clipIn frames).
clips[i].timelineIn	string	Timeline position of the first frame of this clip, where 0 is the first frame of the timeline.
clips[i].timelineInTimecode	string	Timeline timecode of the first frame of this timeline clip serialised as string.

404 - No timeline / disk available.

DELETE /timelines/0

Clear the current playback timeline. Deprecated, prefer to use POST /timelines/0/clear

Response

204 - The timeline was cleared.

501 - The operation is not supported on this device.

POST /timelines/0

Add a clip to the timeline.

Parameters

This parameter can be one of the following types:

Add multiple media clips to the timeline with optional insertion point and clip in/out points.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	array	List of clips to add to the timeline.
clips[i]	object	Clip to add to the timeline, optionally cropping the clip before adding to the timeline.
clips[i].clipUniqueId (required)	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.
clips[i].clipIn	integer	Insert this clip starting from this frame within the media clip. If omitted, starts from the beginning of the clip -- frame 0.
clips[i].frameCount	integer	Number of frames of this clip to add to the timeline. If omitted, use the whole clip, or the rest of the clip if clipIn was specified.

Add multiple media clips to the timeline with optional insertion point.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	array	List of clips to add to the timeline.
clips[i]	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.

Add a single clip to the timeline with optional insertion point and clip in/out points.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	object	Clip to add to the timeline, optionally cropping the clip before adding to the timeline.
clips.clipUniqueId (required)	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.
clips.clipIn	integer	Insert this clip starting from this frame within the media clip. If omitted, starts from the beginning of the clip -- frame 0.
clips.frameCount	integer	Number of frames of this clip to add to the timeline. If omitted, use the whole clip, or the rest of the clip if clipIn was specified.

Add a single clip to the timeline with optional insertion point.

Name	Type	Description
insertBefore	integer	Clip(s) will be inserted before the clip at this timeline clip index, where 0 inserts to the beginning of the timeline. If omitted, inserts to the end of the timeline.
clips (required)	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.

Response

204 - The clip was added to the timeline as specified.

501 - The operation is not supported on this device.

POST /timelines/0/add

Add a clip to the end of the timeline. Deprecated, use POST /timelines/0 to add clips within the timeline.

Parameters

This parameter can be one of the following types:

Add one clip to the end of the timeline.

Name	Type	Description
clips	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.

Add many clips to the end of the timeline.

Name	Type	Description
clips	array	List of clipUniqueIds of clips to add to end of timeline.
clips[i]	integer	Unique ID (clipUniqueId) of the media clip to add to the timeline.

Response

204 - The clip was added to the end of the timeline.

501 - The operation is not supported on this device.

POST /timelines/0/clear

Clear the playback timeline.

Response

204 - The timeline was cleared.

501 - The operation is not supported on this device.

DELETE /timelines/0/clips/{timelineClipIndex}

Remove the specified clip from the timeline.

Parameters

Name	Type	Description
{timelineClipIndex} (required)	integer	The (0-based) timeline clip index of the clip to remove from the timeline.

Response

204 - The specified clip was removed from the timeline.

501 - The operation is not supported on this device.

Media Control API

API for controlling media devices in Blackmagic Design products.

GET /media/workingset

Get the list of media devices currently in the working set.

Response

200 - The list of media devices in the working set.

The response is JSON.

Name	Type	Description
size	integer	The fixed size of this device's working set.
workingset	array	The device's working set.
workingset[i]	object | null	Device within the working set. null if no device is present within the given working set slot.
workingset[i].volume	string	Volume name.
workingset[i].deviceName	string	Internal device name of this media device.
workingset[i].remainingRecordTime	integer	Remaining record time using current codec and video format in seconds.
workingset[i].totalSpace	integer	Total space on media device in bytes.
workingset[i].remainingSpace	integer	Remaining space on media device in bytes.
workingset[i].clipCount	integer	Number of clips currently on the device.

GET /media/active

Get the currently active media device.

Response

200 - The current active media device.

The response is JSON.

The active media device, or null if there is no active media.

Name	Type	Description
workingsetIndex	integer	Working set index of the active media device.
deviceName	string	Device name of media device.

204 - No media is currently active.

PUT /media/active

Set the currently active media device.

Parameters

Name	Type	Description
workingsetIndex	integer	Working set index of the media to make active.

Response

204 - The active media device was set successfully.

400 - Setting the currently active media device is not possible in the current state.

GET /media/devices/doformatSupportedFilesystems

Get the list of filesystems available to format a media device.

Response

200 - The list of filesystems permitted for formatting.

The response is JSON.

Name	Type	Description
	array	List of filesystems permitted for formatting media.
[i]	string	Filesystem serialised as string.

GET /media/devices/{deviceName}

Get information about a requested device.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by “deviceName” member of GET /media/workingset or GET /media/active.

Response

200 - Information about the requested device.

The response is JSON.

Media device state.

Name	Type	Description
state	string	The current state of the media device. Possible values are: None, Scanning, Mounted, Uninitialised, Formatting, RaidComponent.

400 - Invalid device name.

404 - Device not found.

GET /media/devices/{deviceName}/doformat

Get a format key, used to format the device with a PUT request.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by “deviceName” member of GET /media/workingset or GET /media/active.

Response

200 - Format prepared.

The response is JSON.

Name	Type	Description
deviceName	string	Device name of media device to format.
key	string	The key required to format this device, provide to PUT /media/devices/{deviceName}/doformat to perform format of media device.

400 - Cannot format the device.

404 - Device not found.

PUT /media/devices/{deviceName}/doformat

Perform a format of the specified media device.

Parameters

Name	Type	Description
{deviceName} (required)	string	Device name of the media device. Retrieved by “deviceName” member of GET /media/workingset or GET /media/active.

Name	Type	Description
key	string	The key used to format this device, retrieved from prepare format media request GET /media/devices/{deviceName}/doformat. Format key provided cannot be reused after successful format.
filesystem	string	Filesystem to format to. Supported filesystems can be retrieved with GET /media/devices/doFormatSupportedFilesystems.
volume	string	Volume name to set for the disk after format.

Response

204 - Format successful.

400 - Cannot format the device, invalid filesystem or key.

404 - Device not found.

Slate Control API

API to manage digital slate data.

GET /slates/nextClip

Retrieve the digital slate for the next clip.

Response

200 - Returns the slate data for the next clip.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	

Name	Type	Description
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Slate data is not available.

PUT /slates/nextClip

Update the slate data for the next clip.

Parameters

Name	Type	Description
clip	object	
clip.reel	integer	
clip.scene	string	
clip.take	integer	
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

Response

200 - Successfully updated the slate data.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Partial update with errors.

The response is JSON.

Name	Type	Description
error	string	
details	array	
details[i]	object	
details[i].field	string	
details[i].message	string	

POST /slates/nextClip/resetProjectData

Reset the project data for the next clip's slate.

Response

200 - Project data reset successfully.

POST /slates/clips/{deviceName}/{path}/resetProjectData

Reset the project data for the next clip's slate.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Project data reset successfully.

POST /slates/nextClip/resetLensData

Reset the lens data for the next clip's slate.

Response

200 - Lens data reset successfully.

POST /slates/clips/{deviceName}/{path}/resetLensData

Reset the lens data for the next clip's slate.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Lens data reset successfully.

GET /slates/clips/{deviceName}/{path}

Retrieve slate data for a specific clip.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Response

200 - Returns the slate data for the specified clip.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

404 - Clip not found.

PUT /slates/clips/{deviceName}/{path}

Update the slate data for a specific clip.

Parameters

Name	Type	Description
{deviceName} (required)	string	Name of the device where the clip is stored. This is the same as the web browser's device name.
{path} (required)	string	Path to the clip.

Name	Type	Description
clip	object	
clip.reel	integer	
clip.scene	string	
clip.take	integer	
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

Response

200 - Successfully updated the slate data.

The response is JSON.

Name	Type	Description
clip	object	
clip.clipName	string	
clip.reel	integer	
clip.scene	string	
clip.sceneLocation	string	Possible values are: Interior, Exterior.
clip.sceneTime	string	Possible values are: Day, Night.
clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.

Name	Type	Description
clip.slateFor	string	Possible values are: Clip, Next Clip.
clip.take	integer	
clip.takeType	string	Possible values are: None, PU, VFX, SER.
clip.goodTake	boolean	
lens	object	
lens.lensType	string	
lens.iris	string	
lens.focalLength	string	
lens.distance	string	
lens.filter	string	
project	object	
project.projectName	string	
project.director	string	
project.camera	string	
project.cameraOperator	string	

409 - Partial update with errors.

The response is JSON.

Name	Type	Description
error	string	
details	array	
details[i]	object	
details[i].field	string	
details[i].message	string	

Preset Control API

API For controlling the presets on Blackmagic Design products

GET /presets

Get the list of the presets on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
presets	array	List of the presets on the camera (.cset files)
presets[i]	string	

POST /presets

Send a preset file to the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
presetAdded	string	Name of the preset uploaded (without .cset extension)

400 - Bad request - missing Content-Disposition header or filename

GET /presets/active

Get the currently active preset on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
preset	string	Name of the active preset (with .cset extension, or 'default')

PUT /presets/active

Set the active preset on the camera

Parameters

Name	Type	Description
preset	string	Name of the active preset (with .cset extension, or 'default')

Response

204 - No Content

404 - Preset file not found

PARAMETERS /presets/{presetName}

GET /presets/{presetName}

Download the preset file

Response

200 - OK

The response is a binary file.

404 - File does not exist

PUT /presets/{presetName}

Save current camera state as a preset

Response

204 - No Content

DELETE /presets/{presetName}

Delete a preset from the camera

Response

204 - No Content

404 - Preset file not found

Audio Control API

API For controlling audio on Blackmagic Design Cameras

GET /audio/channels

Get the total number of audio channels available

Response

200 - Returns the total number of channels

The response is JSON.

Name	Type	Description
channels	integer	Total number of audio channels available

GET /audio/supportedInputs

Get the list of supported audio inputs

Response

200 - List of all supported audio inputs

The response is JSON.

Name	Type	Description
	array	
[i]	string	A supported audio input

GET /audio/channel/{channelIndex}/input

Get the audio input (source and type) for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently selected input

The response is JSON.

Name	Type	Description
input	string	Audio input source and type

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/input

Set the audio input for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
input	string	Audio input source and type

Response

204 - No Content

400 - Invalid audio input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/input/description

Get the description of the current input of the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Description of the current input of the selected channel

The response is JSON.

Name	Type	Description
description	object	
description.gainRange	object	
description.gainRange.Min	number	The minimum gain value in dB
description.gainRange.Max	number	The maximum gain value in dB
description.capabilities	object	
description.capabilities. PhantomPower	boolean	Input supports setting of phantom power
description.capabilities.LowCutFilter	boolean	Input supports setting of low cut filter
description.capabilities.Padding	object	
description.capabilities.Padding. available	boolean	Input supports setting of padding
description.capabilities.Padding. forced	boolean	Padding is forced to be set for the input
description.capabilities.Padding. value	number	Value of the padding in dB

404 - Channel does not exist

GET /audio/channel/{channelIndex}/supportedInputs

Get the list of supported inputs and their availability to switch to for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its supported inputs are being queried. (Channels index from 0)

Response

200 - The list of supported inputs

The response is JSON.

Name	Type	Description
	array	
[i]	object	
[i].input	string	Input name
[i].available	boolean	Is the input available to be switched into from the current input for the selected channel

404 - Channel does not exist

GET /audio/channel/{channelIndex}/level

Get the audio input level for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set level for the selected channel

The response is JSON.

Name	Type	Description
gain	number	Gain value in dB
normalised	number	Normalised level value between 0.0 and 1.0

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/level

Set the audio input level for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
gain	number	Gain value in dB
normalised	number	Normalised level value between 0.0 and 1.0

Response

204 - No Content

400 - Invalid input or value out of range

404 - Channel does not exist

GET /audio/channel/{channelIndex}/phantomPower

Get the audio input phantom power status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set phantom power for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Phantom power enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/phantomPower

Set the audio phantom power for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Phantom power enabled state

Response

204 - No Content

400 - Phantom power is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/padding

Get the audio input padding status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set padding for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Padding enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/padding

Set the audio input padding for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Padding enabled state

Response

204 - No Content

400 - Padding is not supported or is forced for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/lowCutFilter

Get the audio input low cut filter status for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set low cut filter for the selected channel

The response is JSON.

Name	Type	Description
enabled	boolean	Low cut filter enabled state

404 - Channel does not exist

PUT /audio/channel/{channelIndex}/lowCutFilter

Set the audio input low cut filter for the selected channel

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Name	Type	Description
enabled	boolean	Low cut filter enabled state

Response

204 - No Content

400 - Low cut filter is not supported for this input

404 - Channel does not exist

GET /audio/channel/{channelIndex}/available

Get the audio input's current availability for the selected channel. If unavailable, the source will be muted

Parameters

Name	Type	Description
{channelIndex} (required)	integer	The index of the channel that its input is being controlled. (Channels index from 0)

Response

200 - Currently set availability for the selected channel

The response is JSON.

Name	Type	Description
available	boolean	Whether the input is currently available

404 - Channel does not exist

Lens Control API

API For controlling the lens on Blackmagic Design products

GET /lens/iris

Get lens' aperture

Response

200 - OK

The response is JSON.

Name	Type	Description
continuousApertureAutoExposure	boolean	Is Aperture controlled by auto exposure
apertureStop	number	Aperture stop value
normalised	number	Normalised value
apertureNumber	integer	Aperture number

PUT /lens/iris

Set lens' aperture

Parameters

Name	Type	Description
apertureStop	number	Aperture stop value
normalised	number	Normalised value
apertureNumber	integer	Aperture number
adjustmentStep	integer	Signed value for relative aperture adjustment

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens iris is not controllable or is controlled by auto exposure

GET /lens/zoom

Get lens' zoom

Response

200 - OK

The response is JSON.

Name	Type	Description
focalLength	integer	Focal length in mm
normalised	number	Normalised value

PUT /lens/zoom

Set lens' zoom

Parameters

Name	Type	Description
focalLength	integer	Focal length in mm
normalised	number	Normalised value
adjustmentFocalLength	integer	Signed value for relative focal length adjustment
adjustmentNormalised	number	Signed normalized value for relative zoom adjustment

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens zoom is not controllable

GET /lens/focus

Get lens' focus

Response

200 - OK

The response is JSON.

Name	Type	Description
normalised	number	Normalised value

PUT /lens/focus

Set lens' focus

Parameters

Name	Type	Description
normalised	number	Normalised value
focusDistance	integer	Focus distance value

Response

204 - No Content

400 - Bad Request if out of range value is provided

PUT /lens/focus/doAutoFocus

Perform auto focus

Parameters

Name	Type	Description
position	object	
position.x	number	Normalized x coordinate for autofocus ROI
position.y	number	Normalized y coordinate for autofocus ROI

Response

204 - No Content

400 - Bad Request if out of range value is provided

403 - Forbidden if lens focus is not controllable

GET /lens/opticalImageStabilization

Get optical image stabilization status

Response

200 - OK

The response is JSON.

Name	Type	Description
enabled	boolean	Whether optical image stabilization is enabled

501 - Not Implemented if optical image stabilization is not supported on this product

PUT /lens/opticalImageStabilization

Enable or disable optical image stabilization

Parameters

Name	Type	Description
enabled	boolean	Enable or disable optical image stabilization

Response

204 - No Content

501 - Not Implemented if optical image stabilization is not supported on this product

GET /lens/iris/description

Get detailed description of lens' iris capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the iris can be controlled
apertureStop	object	
apertureStop.min	number	Minimum aperture stop
apertureStop.max	number	Maximum aperture stop

GET /lens/zoom/description

Get detailed description of lens' zoom capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the zoom can be controlled
focalLength	object	
focalLength.adjustable	boolean	If focal length is adjustable
focalLength.min	integer	Minimum focal length
focalLength.max	integer	Maximum focal length

GET /lens/focus/description

Get detailed description of lens' focus capabilities

Response

200 - OK

The response is JSON.

Name	Type	Description
controllable	boolean	If the focus can be controlled
focusDistance	object	
focusDistance.adjustable	boolean	If focus distance is adjustable
focusDistance.min	number	Minimum focus distance
focusDistance.max	number	Maximum focus distance

Video Control API

API For controlling the video on Blackmagic Design products

GET /video/iso

Get current ISO

Response

200 - OK

The response is JSON.

Name	Type	Description
iso	integer	Current ISO value

PUT /video/iso

Set current ISO

Parameters

Name	Type	Description
iso	integer	ISO value to set

Response

204 - No Content

403 - ISO cannot be changed in the current state

GET /video/supportedISOs

Get the list of supported ISO settings

Response

200 - List of supported ISO values

The response is JSON.

Name	Type	Description
supportedISOs	array	Array of supported ISO values
supportedISOs[i]	integer	

GET /video/gain

Get current gain value in decibels

Response

200 - OK

The response is JSON.

Name	Type	Description
gain	integer	Current gain value in decibels

PUT /video/gain

Set current gain value

Parameters

Name	Type	Description
gain	integer	Gain value in decibels to set

Response

204 - No Content

403 - Gain cannot be changed in the current state

GET /video/supportedGains

Get the list of supported gain settings in decibels

Response

200 - List of supported gain values in decibels

The response is JSON.

Name	Type	Description
supportedGains	array	Array of supported gain values in decibels
supportedGains[i]	integer	

GET /video/whiteBalance

Get current white balance

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalance	integer	Current white balance

PUT /video/whiteBalance

Set current white balance

Parameters

Name	Type	Description
whiteBalance	integer	White balance to set

Response

204 - No Content

400 - Invalid white balance temperature

GET /video/whiteBalance/description

Get white balance range

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalance	object	
whiteBalance.min	integer	Minimum color temperature
whiteBalance.max	integer	Maximum color temperature

PUT /video/whiteBalance/doAuto

Set current white balance automatically

Response

204 - No Content

GET /video/whiteBalanceTint

Get white balance tint

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalanceTint	integer	Current white balance tint

PUT /video/whiteBalanceTint

Set white balance tint

Parameters

Name	Type	Description
whiteBalanceTint	integer	White balance tint to set

Response

204 - No Content

400 - Invalid white balance tint

GET /video/whiteBalanceTint/description

Get white balance tint range

Response

200 - OK

The response is JSON.

Name	Type	Description
whiteBalanceTint	object	
whiteBalanceTint.min	integer	Minimum white balance tint
whiteBalanceTint.max	integer	Maximum white balance tint

GET /video/ndFilter

Get ND filter stop

Response

200 - OK

The response is JSON.

Name	Type	Description
stop	number	Current filter power (fStop)

501 - Not implemented for this device

PUT /video/ndFilter

Set ND filter stop

Parameters

Name	Type	Description
stop	number	Filter power (fStop) to set

Response

204 - No Content

400 - Invalid ND filter stop

501 - Not implemented for this device

GET /video/supportedNDFilters

Get the list of available ND filter stops

Response

200 - List of available ND filter stops

The response is JSON.

Name	Type	Description
supportedStops	array	Array of available ND filter stops
supportedStops[i]	number	

501 - Not implemented for this device

GET /video/supportedNDFilterDisplayModes

Get the list of supported ND filter display modes

Response

200 - List of supported display modes

The response is JSON.

Name	Type	Description
supportedDisplayModes	array	Array of supported display modes
supportedDisplayModes[i]	string	Possible values are: Stop, Number, Fraction.

501 - Not implemented for this device

GET /video/ndFilter/displayMode

Get ND filter display mode on the camera

Response

200 - OK

The response is JSON.

Name	Type	Description
displayMode	string	Possible values are: Stop, Number, Fraction.

501 - Not implemented for this device

PUT /video/ndFilter/displayMode

Set ND filter display mode on the camera

Parameters

Name	Type	Description
displayMode	string	Possible values are: Stop, Number, Fraction.

Response

204 - No Content

400 - Invalid display mode for ND filter

501 - Not implemented for this device

GET /video/ndFilterSelectable

Check if ND filter adjustments are selectable via a slider

Response

200 - Indicates if ND filter is selectable

The response is JSON.

Name	Type	Description
selectable	boolean	True if ND filter adjustments are selectable via a slider

501 - Not implemented for this device

GET /video/shutter

Get current shutter. Will return either shutter speed or shutter angle depending on shutter measurement in device settings

Response

200 - OK

The response is JSON.

Name	Type	Description
continuousShutterAutoExposure	boolean	Is shutter controlled by auto exposure
shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
shutterAngle	number	Shutter angle

PUT /video/shutter

Set current shutter

Parameters

Name	Type	Description
shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
shutterAngle	number	Shutter angle

Response

204 - No Content

GET /video/shutter/measurement

Get the current shutter measurement mode

Response

200 - OK

The response is JSON.

Name	Type	Description
measurement	string	Possible values are: ShutterAngle, ShutterSpeed.

PUT /video/shutter/measurement

Set the shutter measurement mode

Parameters

Name	Type	Description
measurement	string	Possible values are: ShutterAngle, ShutterSpeed.

Response

204 - No Content

400 - Invalid shutter measurement

GET /video/supportedShutters

Get supported shutter settings based on current camera configuration

Response

200 - OK

The response is JSON.

Name	Type	Description
shutterAngles	array	Array of supported shutter angles
shutterAngles[i]	number	Shutter angle
shutterSpeeds	array	Array of supported shutter speeds
shutterSpeeds[i]	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)

GET /video/flickerFreeShutters

Get flicker-free shutter settings based on current camera configuration

Response

200 - OK

The response is JSON.

Name	Type	Description
shutterAngles	array	Array of flicker-free shutter angles
shutterAngles[i]	number	Shutter angle
shutterSpeeds	array	Array of flicker-free shutter speeds
shutterSpeeds[i]	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)

GET /video/autoExposure

Get current auto exposure mode

Response

200 - OK

The response is JSON.

Name	Type	Description
mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
type	string	Comma-separated list of device types in the auto exposure stack

PUT /video/autoExposure

Set auto exposure

Parameters

Name	Type	Description
mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
type	string	Comma-separated list of device types in the auto exposure stack

Response

204 - No Content

400 - Failed to set auto exposure mode

GET /video/detailSharpening

Get the current state of detail sharpening

Response

200 - Current detail sharpening state

The response is JSON.

Name	Type	Description
enabled	boolean	Whether detail sharpening is enabled

501 - Not implemented for this device

PUT /video/detailSharpening

Enable or disable detail sharpening

Parameters

Name	Type	Description
enabled	boolean	Enable or disable detail sharpening

Response

204 - Detail sharpening state updated

501 - Not implemented for this device

GET /video/detailSharpeningLevel

Get the current detail sharpening level

Response

200 - Current detail sharpening level

The response is JSON.

Name	Type	Description
level	string	Current detail sharpening level Possible values are: Low, Medium, High.

501 - Not implemented for this device

PUT /video/detailSharpeningLevel

Set the detail sharpening level

Parameters

Name	Type	Description
level	string	Desired level of detail sharpening Possible values are: Low, Medium, High.

Response

204 - Detail sharpening level updated

400 - Invalid detail sharpening level

501 - Not implemented for this device

Camera Control API

API For controlling the Camera specific features on Blackmagic Design products

GET /camera/colorBars

Get the status of color bars display

Response

200 - Returns the current status of color bars

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the color bars are currently enabled

PUT /camera/colorBars

Set the status of color bars display

Parameters

Name	Type	Description
enabled	boolean	Enable or disable the color bars

Response

204 - Color bars status updated successfully

GET /camera/programFeedDisplay

Get the status of program feed display

Response

200 - Returns the current status of program feed display

The response is JSON.

Name	Type	Description
enabled	boolean	Indicates if the program feed display is currently enabled

PUT /camera/programFeedDisplay

Set the status of program feed display

Parameters

Name	Type	Description
enabled	boolean	Enable or disable the program feed display

Response

204 - Program feed display status updated successfully

GET /camera/tallyStatus

Get the tally status of the camera

Response

200 - Returns the current tally status of the camera

The response is JSON.

Name	Type	Description
status	string	Current tally status of the camera Possible values are: None, Preview, Program.

GET /camera/power

Get the power status of the camera

Response

200 - Returns the current power status

The response is JSON.

Name	Type	Description
source	string	Current power source of the camera Possible values are: Battery, AC, Fiber, USB, POE.
milliVolt	integer	Current voltage level in millivolts (rounded to nearest 100mV)
batteries	array	List of batteries currently connected to the camera
batteries[i]	object	
batteries[i].milliVolt	integer	Battery voltage in millivolts (rounded to nearest 100mV)
batteries[i].chargeRemainingPercent	integer	Remaining battery charge percentage
batteries[i].statusFlags	array	List of battery status flags
batteries[i].statusFlags[i]	string	Possible values are: Unknown Battery Status, Battery Is Present, Battery Is Charging, Battery Percentage Is Low, Battery Voltage Is Low, Battery Is Critically Low, Charge Remaining Percentage Is Estimated, Battery Communications Is Active, Battery Is Connected.

GET /camera/power/displayMode

Get the power display mode of the camera

Response

200 - Returns the current power display mode

The response is JSON.

Name	Type	Description
mode	string	Current power display mode Possible values are: Percentage, Voltage.

PUT /camera/power/displayMode

Set the power display mode of the camera

Parameters

Name	Type	Description
mode	string	Power display mode to set Possible values are: Percentage, Voltage.

Response

204 - Power display mode updated successfully

400 - Invalid power display mode

GET /camera/timingReferenceLock

Get the timing reference lock status

Response

200 - Returns the timing reference lock status

The response is JSON.

Name	Type	Description
locked	boolean	Indicates if timing reference is locked

Immersive Control API

API for controlling immersive camera settings on Blackmagic Design cameras

GET /immersive/display/{displayName}/eye

Get the current eye view for a specific display

Parameters

Name	Type	Description
{displayName} (required)	string	The display name to query (from /monitoring/display endpoint)

Response

200 - OK

The response is JSON.

Name	Type	Description
eye (required)	string	The eye view to display Possible values are: Left, Right.

400 - Invalid display name format

404 - Display not found

422 - Failed to get eye view

PUT /immersive/display/{displayName}/eye

Set the eye view for a specific display

Parameters

Name	Type	Description
{displayName} (required)	string	The display name to control (from /monitoring/display endpoint)

Name	Type	Description
eye (required)	string	The eye view to display Possible values are: Left, Right.

Response

204 - No Content

400 - Invalid input or display name format

404 - Display not found

422 - Failed to set eye view

Color Correction Control API

API For controlling the color correction on Blackmagic Design products based on DaVinci Resolve Color Corrector

GET /colorCorrection/lift

Get color correction lift

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red lift component. If omitted, value remains unchanged.
green	number	Green lift component. If omitted, value remains unchanged.
blue	number	Blue lift component. If omitted, value remains unchanged.
luma	number	Luma lift component. If omitted, value remains unchanged.

PUT /colorCorrection/lift

Set color correction lift

Parameters

Name	Type	Description
red	number	Red lift component. If omitted, value remains unchanged.
green	number	Green lift component. If omitted, value remains unchanged.
blue	number	Blue lift component. If omitted, value remains unchanged.
luma	number	Luma lift component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/gamma

Get color correction gamma

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red gamma component. If omitted, value remains unchanged.
green	number	Green gamma component. If omitted, value remains unchanged.
blue	number	Blue gamma component. If omitted, value remains unchanged.
luma	number	Luma gamma component. If omitted, value remains unchanged.

PUT /colorCorrection/gamma

Set color correction gamma

Parameters

Name	Type	Description
red	number	Red gamma component. If omitted, value remains unchanged.
green	number	Green gamma component. If omitted, value remains unchanged.
blue	number	Blue gamma component. If omitted, value remains unchanged.
luma	number	Luma gamma component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/gain

Get color correction gain

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red gain component. If omitted, value remains unchanged.
green	number	Green gain component. If omitted, value remains unchanged.
blue	number	Blue gain component. If omitted, value remains unchanged.
luma	number	Luma gain component. If omitted, value remains unchanged.

PUT /colorCorrection/gain

Set color correction gain

Parameters

Name	Type	Description
red	number	Red gain component. If omitted, value remains unchanged.
green	number	Green gain component. If omitted, value remains unchanged.
blue	number	Blue gain component. If omitted, value remains unchanged.
luma	number	Luma gain component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/offset

Get color correction offset

Response

200 - OK

The response is JSON.

Name	Type	Description
red	number	Red offset component. If omitted, value remains unchanged.
green	number	Green offset component. If omitted, value remains unchanged.
blue	number	Blue offset component. If omitted, value remains unchanged.
luma	number	Luma offset component. If omitted, value remains unchanged.

PUT /colorCorrection/offset

Set color correction offset

Parameters

Name	Type	Description
red	number	Red offset component. If omitted, value remains unchanged.
green	number	Green offset component. If omitted, value remains unchanged.
blue	number	Blue offset component. If omitted, value remains unchanged.
luma	number	Luma offset component. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/contrast

Get color correction contrast

Response

200 - OK

The response is JSON.

Name	Type	Description
pivot	number	Contrast pivot point. If omitted, value remains unchanged.
adjust	number	Contrast adjustment. If omitted, value remains unchanged.

PUT /colorCorrection/contrast

Set color correction contrast

Parameters

Name	Type	Description
pivot	number	Contrast pivot point. If omitted, value remains unchanged.
adjust	number	Contrast adjustment. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/color

Get color correction color properties

Response

200 - OK

The response is JSON.

Name	Type	Description
hue	number	Color hue adjustment. If omitted, value remains unchanged.
saturation	number	Color saturation adjustment. If omitted, value remains unchanged.

PUT /colorCorrection/color

Set color correction color properties

Parameters

Name	Type	Description
hue	number	Color hue adjustment. If omitted, value remains unchanged.
saturation	number	Color saturation adjustment. If omitted, value remains unchanged.

Response

204 - No Content

GET /colorCorrection/lumaContribution

Get color correction luma contribution

Response

200 - OK

The response is JSON.

Name	Type	Description
lumaContribution	number	Luma contribution value. If omitted, value remains unchanged.

PUT /colorCorrection/lumaContribution

Set color correction luma contribution

Parameters

Name	Type	Description
lumaContribution	number	Luma contribution value. If omitted, value remains unchanged.

Response

204 - No Content

Notification websocket - 1.0.0

Service that notifies subscribers of device state changes.

messages

Subscribe (The messages from the server/device)

Websocket Opened Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: websocketOpened.
.type	string	Possible values are: event.

Response Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: subscribe, unsubscribe, listSubscriptions, listProperties, websocketOpened.
.data.properties	array	
.data.properties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*

Name	Type	Description
.data.values	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.data.success	boolean	
.data.deviceProperties	array	
.data.deviceProperties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*
.type	string	Possible values are: response.
.id	number	Optional parameter that repeats the id in the output for tracking messages.

Event Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: propertyValueChanged.

Name	Type	Description
.data.property	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*
.data.value	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.type	string	Possible values are: event.

Publish (The messages that user can send to the server/device)

Response Message (JSON)

Name	Type	Description
.data	object	
.data.action	string	Possible values are: subscribe, unsubscribe, listSubscriptions, listProperties, websocketOpened.
.data.properties	array	

Name	Type	Description
.data.properties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*
.data.values	object	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.
.data.success	boolean	
.data.deviceProperties	array	

Name	Type	Description
.data.deviceProperties[i]	string	device property that the user can subscribe to. The user can either choose a value from the predefined enum, or provide a wildcard string. Possible values are: /media/workingset, /media/active, /system, /system/codecFormat, /system/videoFormat, /system/format, /system/supportedFormats, /timelines/0, /transports/0, /transports/0/stop, /transports/0/play, /transports/0/playback, /transports/0/record, /transports/0/timecode, /transports/0/timecode/source, /transports/0/clipIndex, /slates/nextClip, /monitoring/{displayName}/cleanFeed, /monitoring/{displayName}/displayLUT, /monitoring/{displayName}/zebra, /monitoring/{displayName}/focusAssist, /monitoring/{displayName}/frameGuide, /monitoring/{displayName}/frameGrids, /monitoring/{displayName}/safeArea, /monitoring/{displayName}/falseColor, /monitoring/focusAssist, /monitoring/frameGuideRatio, /monitoring/frameGrids, /monitoring/safeAreaPercent, /audio/channel/{channelIndex}/input, /audio/channel/{channelIndex}/supportedInputs, /audio/channel/{channelIndex}/level, /audio/channel/{channelIndex}/phantomPower, /audio/channel/{channelIndex}/padding, /audio/channel/{channelIndex}/lowCutFilter, /audio/channel/{channelIndex}/available, /audio/channel/{channelIndex}/input/description, /colorCorrection/lift, /colorCorrection/gamma, /colorCorrection/gain, /colorCorrection/offset, /colorCorrection/contrast, /colorCorrection/color, /colorCorrection/lumaContribution, /lens/iris, /lens/iris/description, /lens/focus, /lens/focus/description, /lens/zoom, /lens/zoom/description, /presets, /presets/active, /camera/colorBars, /camera/programFeedDisplay, /camera/tallyStatus, /camera/power, /camera/power/displayMode, /camera/timingReferenceLock, /video/iso, /video/supportedISOs, /video/gain, /video/supportedGains, /video/whiteBalance, /video/whiteBalance/description, /video/whiteBalanceTint, /video/whiteBalanceTint/description, /video/ndFilter, /video/supportedNDFilters, /video/ndFilter/displayMode, /video/supportedNDFilterDisplayModes, /video/ndFilterSelectable, /video/shutter, /video/shutter/measurement, /video/supportedShutters, /video/flickerFreeShutters, /video/autoExposure, /video/detailSharpening, /video/detailSharpeningLevel. Must match the pattern: .*
.type	string	Possible values are: response.
.id	number	Optional parameter that repeats the id in the output for tracking messages.

Device Properties

/media/workingset

The value JSON returned via the eventResponse when the `/media/workingset` property changes on the device:

Name	Type	Description
.size	integer	The fixed size of this device's working set.
.workingset	array	Array of devices within the working set. null if no device is present within the given working set slot.
.workingset[i]		

/media/active

The value JSON returned via the eventResponse when the `/media/active` property changes on the device:

Name	Type	Description
.workingsetIndex	integer	Working set index of the active media device.
.deviceName	string	Internal device name of this media device.

/system

The value JSON returned via the eventResponse when the `/system` property changes on the device:

Name	Type	Description
.codecFormat	object	Codec format configuration.
.codecFormat.codec	string	Codec format serialised as a string.
.codecFormat.container	string	Multimedia container format.
.videoFormat	object	Currently selected video format.
.videoFormat.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.videoFormat.height	number	Height dimension of video format.
.videoFormat.width	number	Width dimension of video format.
.videoFormat.interlaced	boolean	Is the display format interlaced?.
.videoFormat.name	string	Video format serialised as a string.

/system/codecFormat

Codec format configuration.

The value JSON returned via the eventResponse when the `/system/codecFormat` property changes on the device:

Name	Type	Description
.codec	string	Codec format serialised as a string.
.container	string	Multimedia container format.

/system/videoFormat

Currently selected video format.

The value JSON returned via the eventResponse when the `/system/videoFormat` property changes on the device:

Name	Type	Description
.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.height	number	Height dimension of video format.
.width	number	Width dimension of video format.
.interlaced	boolean	Is the display format interlaced?.
.name	string	Video format serialised as a string.

/system/format

The value JSON returned via the eventResponse when the `/system/format` property changes on the device:

Name	Type	Description
.codec	string	Codec format serialised as a string.
.frameRate	string	Frame rate. Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.maxOffSpeedFrameRate	number	
.minOffSpeedFrameRate	number	
.offSpeedEnabled	boolean	
.offSpeedFrameRate	number	
.recordResolution	object	
.recordResolution.height	number	Height of the resolution.
.recordResolution.width	number	Width of the resolution.
.sensorResolution	object	
.sensorResolution.height	number	Height of the resolution.
.sensorResolution.width	number	Width of the resolution.

/system/supportedFormats

The value JSON returned via the eventResponse when the `/system/supportedFormats` property changes on the device:

Name	Type	Description
.supportedFormats	array	
.supportedFormats[i]	object	
.supportedFormats[i].codecs	array	
.supportedFormats[i].codecs[i]	string	
.supportedFormats[i].frameRates	array	
.supportedFormats[i].frameRates[i]	string	Possible values are: 23.98, 24.00, 24, 25.00, 25, 29.97, 30.00, 30, 47.95, 48.00, 48, 50.00, 50, 59.94, 60.00, 60, 119.88, 120.00, 120.
.supportedFormats[i].maxOffSpeedFrameRate	number	
.supportedFormats[i].minOffSpeedFrameRate	number	
.supportedFormats[i].recordResolution	object	
.supportedFormats[i].recordResolution.height	number	Height of the resolution.
.supportedFormats[i].recordResolution.width	number	Width of the resolution.
.supportedFormats[i].sensorResolution	object	
.supportedFormats[i].sensorResolution.height	number	Height of the resolution.
.supportedFormats[i].sensorResolution.width	number	Width of the resolution.

/timelines/0

The value JSON returned via the eventResponse when the `/timelines/0` property changes on the device:

Name	Type	Description
.clips	array	
.clips[i]	object	Timeline clip.
.clips[i].clipUniqueId	integer	Unique identifier used to identify this media clip. If the same media clip is added to the timeline multiple times, each timeline clip has the same clipUniqueId
.clips[i].frameCount	integer	Duration of timeline clip in frames, the number of frames in this clip on the timeline.
.clips[i].durationTimecode	string	Duration of the timeline clip in timecode format serialised as string. This will differ to durationTimecode reported in /clips for this clipUniqueId if clipIn or frameCount was specified when adding this clip to the timeline.
.clips[i].clipIn	string	In frame offset for the clip on the timeline, where 0 is the first frame of the on-disk clip.
.clips[i].inTimecode	string	Clip timecode of the first frame of this timeline clip serialised as string (clip startTimecode + clipIn frames).

Name	Type	Description
.clips[i].timelineIn	string	Timeline position of the first frame of this clip, where 0 is the first frame of the timeline.
.clips[i].timelineInTimecode	string	Timeline timecode of the first frame of this timeline clip serialised as string.

/transports/0

The value JSON returned via the eventResponse when the `/transports/0` property changes on the device:

Name	Type	Description
.mode	string	Transport mode. Possible values are: InputPreview, InputRecord, Output.

/transports/0/stop

true when transport mode is InputPreview or when in Output mode and speed is 0.

The value JSON returned via the eventResponse when the `/transports/0/stop` property changes on the device:

Name	Type	Description
	boolean	true when transport mode is InputPreview or when in Output mode and speed is 0.

/transports/0/play

True when transport is in Output mode and speed is non-zero.

The value JSON returned via the eventResponse when the /transports/0/play property changes on the device:

Name	Type	Description
	boolean	True when transport is in Output mode and speed is non-zero.

/transports/0/playback

The value JSON returned via the eventResponse when the `/transports/0/playback` property changes on the device:

Name	Type	Description
.type	string	Possible values are: Play, Jog, Shuttle, Var.
.loop	boolean	When true playback loops from the end of the timeline to the beginning of the timeline.
.singleClip	boolean	When true playback loops from the end of the current clip to the beginning of the current clip.
.speed	number	Playback speed, 1.0 for normal forward playback
.position	integer	Playback position on the timeline in units of video frames.

/transports/0/record

The value JSON returned via the eventResponse when the `/transports/0/record` property changes on the device:

Name	Type	Description
.recording	boolean	Is transport in Input Record mode.

/transports/0/timecode

The value JSON returned via the eventResponse when the `/transports/0/timecode` property changes on the device:

Name	Type	Description
.display	string	The display timecode serialised as a string.
.timeline	string	The timeline timecode serialised as a string.

/transports/0/timecode/source

The value JSON returned via the eventResponse when the `/transports/0/timecode/source` property changes on the device:

Name	Type	Description
.timecode	string	Possible values are: Timeline, Clip.

/transports/0/clipIndex

The value JSON returned via the eventResponse when the `/transports/0/clipIndex` property changes on the device:

Name	Type	Description
.clipIndex	number | null	The 0-based index of the clip being played on the timeline. null when there is no timeline or an empty timeline.

/slates/nextClip

The value JSON returned via the eventResponse when the `/slates/nextClip` property changes on the device:

Name	Type	Description
.clip	object	
.clip.clipName	string	
.clip.reel	integer	
.clip.scene	string	
.clip.sceneLocation	string	Possible values are: Interior, Exterior.
.clip.sceneTime	string	Possible values are: Day, Night.
.clip.shotType	string	Possible values are: None, WS, MS, MCU, CU, BCU, ECU.
.clip.take	integer	
.clip.takeType	string	Possible values are: None, PU, VFX, SER.
.clip.goodTake	boolean	
.lens	object	
.lens.lensType	string	
.lens.iris	string	

Name	Type	Description
.lens.focalLength	string	
.lens.distance	string	
.lens.filter	string	
.project	object	
.project.projectName	string	
.project.director	string	
.project.camera	string	
.project.cameraOperator	string	

/monitoring/{displayName}/cleanFeed

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/cleanFeed` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/displayLUT

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/displayLUT` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/zebra

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/zebra` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/focusAssist

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/focusAssist` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/frameGuide

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/frameGuide` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/frameGrids

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/frameGrids` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/safeArea

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/safeArea` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/{displayName}/falseColor

The value JSON returned via the eventResponse when the `/monitoring/{displayName}/falseColor` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the feature is enabled.

/monitoring/focusAssist

The value JSON returned via the eventResponse when the `/monitoring/focusAssist` property changes on the device:

Name	Type	Description
.mode	string	Possible values are: Peak, ColoredLines.
.color	string	Possible values are: Red, Green, Blue, White, Black.
.intensity	integer	

/monitoring/frameGuideRatio

The value JSON returned via the eventResponse when the `/monitoring/frameGuideRatio` property changes on the device:

Name	Type	Description
.ratio	string	

/monitoring/frameGrids

The value JSON returned via the eventResponse when the `/monitoring/frameGrids` property changes on the device:

Name	Type	Description
.frameGrids	array	
.frameGrids[i]	string	Possible values are: Thirds, Crosshair, Dot, Horizon.

/monitoring/safeAreaPercent

The value JSON returned via the eventResponse when the `/monitoring/safeAreaPercent` property changes on the device:

Name	Type	Description
.percent	integer	Safe area coverage percentage.

/audio/channel/{channelIndex}/input

Get the audio input (source and type) for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/input` property changes on the device:

Name	Type	Description
.input	string	Audio input source and type

/audio/channel/{channelIndex}/supportedInputs

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/supportedInputs` property changes on the device:

Name	Type	Description
	array	
[i]	object	
[i].input	string	Input name
[i].available	boolean	Is the input available to be switched into from the current input for the selected channel

/audio/channel/{channelIndex}/level

Get the audio input level for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/level` property changes on the device:

Name	Type	Description
.gain	number	Gain value in dB
.normalised	number	Normalised level value between 0.0 and 1.0

/audio/channel/{channelIndex}/phantomPower

Get the audio input phantom power status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/phantomPower` property changes on the device:

Name	Type	Description
.enabled	boolean	Phantom power enabled state

/audio/channel/{channelIndex}/padding

Get the audio input padding status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/padding` property changes on the device:

Name	Type	Description
.enabled	boolean	Padding enabled state

/audio/channel/{channelIndex}/lowCutFilter

Get the audio input low cut filter status for the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/lowCutFilter` property changes on the device:

Name	Type	Description
.enabled	boolean	Low cut filter enabled state

/audio/channel/{channelIndex}/available

Get the audio input's current availability for the selected channel. If unavailable, the source will be muted

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/available` property changes on the device:

Name	Type	Description
.available	boolean	Whether the input is currently available

/audio/channel/{channelIndex}/input/description

Description of the current input of the selected channel

The value JSON returned via the eventResponse when the `/audio/channel/{channelIndex}/input/description` property changes on the device:

Name	Type	Description
.description	object	
.description.gainRange	object	
.description.gainRange.Min	number	The minimum gain value in dB
.description.gainRange.Max	number	The maximum gain value in dB
.description.capabilities	object	
.description.capabilities.PhantomPower	boolean	Input supports setting of phantom power
.description.capabilities.LowCutFilter	boolean	Input supports setting of low cut filter

Name	Type	Description
.description.capabilities.Padding	object	
.description.capabilities.Padding.available	boolean	Input supports setting of padding
.description.capabilities.Padding.forced	boolean	Padding is forced to be set for the input
.description.capabilities.Padding.value	number	An object with property names as the key and a property value as json. Check the next section for the device properties and their return values.

/colorCorrection/lift

Get color correction lift

The value JSON returned via the eventResponse when the `/colorCorrection/lift` property changes on the device:

Name	Type	Description
.red	number	Red lift component
.green	number	Green lift component
.blue	number	Blue lift component
.luma	number	Luma lift component

/colorCorrection/gamma

Get color correction gamma

The value JSON returned via the eventResponse when the `/colorCorrection/gamma` property changes on the device:

Name	Type	Description
.red	number	Red gamma component
.green	number	Green gamma component
.blue	number	Blue gamma component
.luma	number	Luma gamma component

/colorCorrection/gain

Get color correction gain

The value JSON returned via the eventResponse when the `/colorCorrection/gain` property changes on the device:

Name	Type	Description
.red	number	Red gain component
.green	number	Green gain component
.blue	number	Blue gain component
.luma	number	Luma gain component

/colorCorrection/offset

Get color correction offset

The value JSON returned via the eventResponse when the `/colorCorrection/offset` property changes on the device:

Name	Type	Description
.red	number	Red offset component
.green	number	Green offset component
.blue	number	Blue offset component
.luma	number	Luma offset component

/colorCorrection/contrast

Get color correction contrast

The value JSON returned via the eventResponse when the `/colorCorrection/contrast` property changes on the device:

Name	Type	Description
.pivot	number	Contrast pivot point
.adjust	number	Contrast adjustment

/colorCorrection/color

Get color correction color properties

The value JSON returned via the eventResponse when the `/colorCorrection/color` property changes on the device:

Name	Type	Description
.hue	number	Color hue adjustment
.saturation	number	Color saturation adjustment

/colorCorrection/lumaContribution

Get color correction luma contribution

The value JSON returned via the eventResponse when the `/colorCorrection/lumaContribution` property changes on the device:

Name	Type	Description
.lumaContribution	number	Luma contribution value

/lens/iris

Get lens' aperture

The value JSON returned via the eventResponse when the `/lens/iris` property changes on the device:

Name	Type	Description
.continuousApertureAutoExposure	boolean	Is Aperture controlled by auto exposure
.apertureStop	number	Aperture stop value
.normalised	number	Normalised value
.apertureNumber	integer	Aperture number

/lens/iris/description

Get detailed description of lens' iris capabilities

The value JSON returned via the eventResponse when the `/lens/iris/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the iris can be controlled
.apertureStop	object	
.apertureStop.min	number	Minimum aperture stop
.apertureStop.max	number	Maximum aperture stop

/lens/focus

Get lens' focus

The value JSON returned via the eventResponse when the `/lens/focus` property changes on the device:

Name	Type	Description
.normalised	number	Normalised value

/lens/focus/description

Get detailed description of lens' focus capabilities

The value JSON returned via the eventResponse when the `/lens/focus/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the focus can be controlled
.focusDistance	object	
.focusDistance.adjustable	boolean	If focus distance is adjustable
.focusDistance.min	number	Minimum focus distance
.focusDistance.max	number	Maximum focus distance

/lens/zoom

Get lens' zoom

The value JSON returned via the eventResponse when the `/lens/zoom` property changes on the device:

Name	Type	Description
.focalLength	integer	Focal length in mm
.normalised	number	Normalised value

/lens/zoom/description

Get detailed description of lens' zoom capabilities

The value JSON returned via the eventResponse when the `/lens/zoom/description` property changes on the device:

Name	Type	Description
.controllable	boolean	If the zoom can be controlled
.focalLength	object	
.focalLength.adjustable	boolean	If focal length is adjustable
.focalLength.min	integer	Minimum focal length
.focalLength.max	integer	Maximum focal length

/presets

Get the list of the presets on the camera

The value JSON returned via the eventResponse when the `/presets` property changes on the device:

Name	Type	Description
.presets	array	List of the presets on the camera (.cset files)
.presets[i]	string	

/presets/active

Get the currently active preset on the camera

The value JSON returned via the eventResponse when the `/presets/active` property changes on the device:

Name	Type	Description
.preset	string	Name of the active preset (with .cset extension, or 'default')

/camera/colorBars

Get the status of color bars display

The value JSON returned via the eventResponse when the `/camera/colorBars` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the color bars are currently enabled

/camera/programFeedDisplay

Get the status of program feed display

The value JSON returned via the eventResponse when the `/camera/programFeedDisplay` property changes on the device:

Name	Type	Description
.enabled	boolean	Indicates if the program feed display is currently enabled

/camera/tallyStatus

Get the tally status of the camera

The value JSON returned via the eventResponse when the `/camera/tallyStatus` property changes on the device:

Name	Type	Description
.status	string	Current tally status of the camera Possible values are: None, Preview, Program.

/camera/power

Get the power status of the camera

The value JSON returned via the eventResponse when the `/camera/power` property changes on the device:

Name	Type	Description
.source	string	Current power source of the camera Possible values are: Battery, AC, Fiber, USB, POE.
.milliVolt	integer	Current voltage level in millivolts (rounded to nearest 100mV)
.batteries	array	
.batteries[i]	object	
.batteries[i].milliVolt	integer	Battery voltage in millivolts (rounded to nearest 100mV)
.batteries[i]. chargeRemainingPercent	integer	Remaining battery charge percentage
.batteries[i].statusFlags	array	List of battery status flags
.batteries[i].statusFlags[i]	string	Possible values are: Unknown Battery Status, Battery Is Present, Battery Is Charging, Battery Percentage Is Low, Battery Voltage Is Low, Battery Is Critically Low, Charge Remaining Percentage Is Estimated, Battery Communications Is Active, Battery Is Connected.

/camera/power/displayMode

Get the power display mode of the camera

The value JSON returned via the eventResponse when the `/camera/power/displayMode` property changes on the device:

Name	Type	Description
.mode	string	Current power display mode Possible values are: Percentage, Voltage.

/camera/timingReferenceLock

Get the timing reference lock status

The value JSON returned via the eventResponse when the `/camera/timingReferenceLock` property changes on the device:

Name	Type	Description
.locked	boolean	Indicates if timing reference is locked

/video/iso

Get current ISO

The value JSON returned via the eventResponse when the `/video/iso` property changes on the device:

Name	Type	Description
.iso	integer	Current ISO value

/video/supportedISOs

Get the list of supported ISO settings

The value JSON returned via the eventResponse when the `/video/supportedISOs` property changes on the device:

Name	Type	Description
.supportedISOs	array	Array of supported ISO values
.supportedISOs[i]	integer	

/video/gain

Get current gain value in decibels

The value JSON returned via the eventResponse when the `/video/gain` property changes on the device:

Name	Type	Description
.gain	integer	Current gain value in decibels

/video/supportedGains

Get the list of supported gain settings in decibels

The value JSON returned via the eventResponse when the `/video/supportedGains` property changes on the device:

Name	Type	Description
.supportedGains	array	Array of supported gain values in decibels
.supportedGains[i]	integer	

/video/whiteBalance

Get current white balance

The value JSON returned via the eventResponse when the `/video/whiteBalance` property changes on the device:

Name	Type	Description
.whiteBalance	integer	Current white balance

/video/whiteBalance/description

Get white balance range

The value JSON returned via the eventResponse when the `/video/whiteBalance/description` property changes on the device:

Name	Type	Description
.whiteBalance	object	
.whiteBalance.min	integer	Minimum color temperature
.whiteBalance.max	integer	Maximum color temperature

/video/whiteBalanceTint

Get white balance tint

The value JSON returned via the eventResponse when the `/video/whiteBalanceTint` property changes on the device:

Name	Type	Description
.whiteBalanceTint	integer	Current white balance tint

/video/whiteBalanceTint/description

Get white balance tint range

The value JSON returned via the eventResponse when the `/video/whiteBalanceTint/description` property changes on the device:

Name	Type	Description
.whiteBalanceTint	object	
.whiteBalanceTint.min	integer	Minimum white balance tint
.whiteBalanceTint.max	integer	Maximum white balance tint

/video/ndFilter

Get ND filter stop

The value JSON returned via the eventResponse when the `/video/ndFilter` property changes on the device:

Name	Type	Description
.stop	number	Current filter power (fStop)

/video/supportedNDFilters

Get the list of available ND filter stops

The value JSON returned via the eventResponse when the `/video/supportedNDFilters` property changes on the device:

Name	Type	Description
.supportedStops	array	Array of available ND filter stops
.supportedStops[i]	number	

/video/ndFilter/displayMode

Get ND filter display mode on the camera

The value JSON returned via the eventResponse when the `/video/ndFilter/displayMode` property changes on the device:

Name	Type	Description
.displayMode	string	ND filter display mode Possible values are: Stop, Number, Fraction.

/video/supportedNDFilterDisplayModes

Get the list of supported ND filter display modes

The value JSON returned via the eventResponse when the `/video/supportedNDFilterDisplayModes` property changes on the device:

Name	Type	Description
.supportedDisplayModes	array	Array of supported display modes
.supportedDisplayModes[i]	string	Possible values are: Stop, Number, Fraction.

/video/ndFilterSelectable

Check if ND filter adjustments are selectable via a slider

The value JSON returned via the eventResponse when the `/video/ndFilterSelectable` property changes on the device:

Name	Type	Description
.selectable	boolean	True if ND filter adjustments are selectable via a slider

/video/shutter

Get current shutter. Will return either shutter speed or shutter angle depending on shutter measurement in device settings

The value JSON returned via the eventResponse when the `/video/shutter` property changes on the device:

Name	Type	Description
.continuousShutterAutoExposure	boolean	Is shutter controlled by auto exposure
.shutterSpeed	integer	Shutter speed value in fractions of a second (minimum is sensor frame rate)
.shutterAngle	number	Shutter angle

/video/shutter/measurement

Get the current shutter measurement mode

The value JSON returned via the eventResponse when the `/video/shutter/measurement` property changes on the device:

Name	Type	Description
.measurement	string	Current shutter measurement mode Possible values are: ShutterAngle, ShutterSpeed.

/video/supportedShutters

Get supported shutter settings based on current camera configuration

The value JSON returned via the eventResponse when the `/video/supportedShutters` property changes on the device:

Name	Type	Description
.shutterAngles	array	Array of supported shutter angles
.shutterAngles[i]	number	
.shutterSpeeds	array	Array of flicker-free shutter speeds
.shutterSpeeds[i]	integer	

/video/flickerFreeShutters

Get flicker-free shutter settings based on current camera configuration

The value JSON returned via the eventResponse when the `/video/flickerFreeShutters` property changes on the device:

Name	Type	Description
.shutterAngles	array	Array of flicker-free shutter angles
.shutterAngles[i]	number	
.shutterSpeeds	array	Array
.shutterSpeeds[i]	integer	

/video/autoExposure

Get current auto exposure mode

The value JSON returned via the eventResponse when the `/video/autoExposure` property changes on the device:

Name	Type	Description
.mode	string	Auto exposure mode Possible values are: Off, Continuous, OneShot.
.type	string	Comma-separated list of device types in the auto exposure stack

/video/detailSharpening

Get the current state of detail sharpening

The value JSON returned via the eventResponse when the `/video/detailSharpening` property changes on the device:

Name	Type	Description
.enabled	boolean	Whether detail sharpening is enabled

/video/detailSharpeningLevel

Get the current detail sharpening level

The value JSON returned via the eventResponse when the `/video/detailSharpeningLevel` property changes on the device:

Name	Type	Description
.level	string	Current detail sharpening level of supported shutter speeds Possible values are: Low, Medium, High.

Help

The fastest way to obtain help is to go to the Blackmagic Design online support pages and check the latest support material available for your camera.

Blackmagic Design Online Support Pages

The latest manual, software and support notes can be found at the Blackmagic Design support center at www.blackmagicdesign.com/support.

Contacting Blackmagic Design Support

If you can't find the help you need in our support material, please use the 'Send us an email' button on the support page to email a support request. Alternatively, click on the 'Find your local support team' button on the support page and call your nearest Blackmagic Design support office.

Checking the Software Version Currently Installed

To check which version of Blackmagic Camera Setup utility software is installed on your computer, open the About Blackmagic Camera Setup utility window.

- On Mac, open Blackmagic Camera Setup utility from the Applications folder. Select About Blackmagic Camera Setup utility from the application menu to reveal the version number.
- On Windows, open Blackmagic Camera Setup utility from your Start menu or Start Screen. Click on the Help menu and select About Blackmagic Camera Setup utility to reveal the version number.

How to Get the Latest Software Updates

After checking the version of Blackmagic Camera Utility software installed on your computer, please visit the Blackmagic Design support center at www.blackmagicdesign.com/support to check for the latest updates. While it is usually a good idea to run the latest updates, it is wise to avoid updating any software if you are in the middle of an important project.

Regulatory Notices

Disposal of Waste of Electrical and Electronic Equipment Within the European Union.

The symbol on the product indicates that this equipment must not be disposed of with other waste materials. In order to dispose of your waste equipment, it must be handed over to a designated collection point for recycling. The separate collection and recycling of your waste equipment at the time of disposal will help conserve natural resources and ensure that it is recycled in a manner that protects human health and the environment. For more information about where you can drop off your waste equipment for recycling, please contact your local city recycling office or the dealer from whom you purchased the product.

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. Operation of this product in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at personal expense.

The operation of this equipment is subject to the following two conditions:

1. This equipment or device may not cause harmful interference.
2. This equipment or device must accept any interference received, including interference that may cause undesired operation.

R-R-BMD-20250213001

ISED Canada Statement

This device complies with Canadian standards for Class A digital apparatus.

Any modifications or use of this product outside its intended use could void compliance to these standards.

Connection to HDMI interfaces must be made with high quality shielded HDMI cables.

This equipment has been tested for compliance with the intended use in a commercial environment. If the equipment is used in a domestic environment, it may cause radio interference.

Safety Information

The supplied AC to 24V DC power supply for Blackmagic URSA Cine must be connected to a mains socket outlet with a protective earth connection.

No operator serviceable parts inside Blackmagic URSA Cine. Refer servicing to your local Blackmagic Design service center.

When connecting the input power connector to an external DC battery source, the supply wiring must include current limiting or fusing.

The wiring used should be marked VW-1 or comply with the relevant parts of IEC 60332 or IEC 60695".

We recommend that the DC source for this camera is 24V. If a 12V DC source is used, the wiring used should be sufficient for a current of 14A.

A B-Mount battery plate is provided with this product. We recommend the use of a 24V (28.8V) battery or dual voltage battery.

For a 12V (14.4V) battery, refer to your battery manual or markings to determine the Maximum Continuous Discharge Current. We recommend that this has a minimum rating of 12A.

State of California statement

This product can expose you to chemicals such as trace amounts of polybrominated biphenyls within plastic parts, which is known to the state of California to cause cancer and birth defects or other reproductive harm.

For more information go to www.P65Warnings.ca.gov.

European Office

Blackmagic Design Europe B.V.
Rijnlanderweg 766, Unit D
2132 NM Hoofddorp
NL

Blackmagic URSA Cine

During sunny conditions, consider shading of the camera to prevent exposure of the camera or Lithium battery to extended periods of sunlight. Keep Lithium batteries away from all sources of heat.

The 24V 'RS' output connector is suitable to provide power to the third party accessories. Please ensure the power consumption is less than 48W.

Warranty

Limited Warranty

Blackmagic Design warrants that this product will be free from defects in materials and workmanship for a period of 12 months from the date of purchase. If a product proves to be defective during this warranty period, Blackmagic Design, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, you the Customer, must notify Blackmagic Design of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. The Customer shall be responsible for packaging and shipping the defective product to a designated service center nominated by Blackmagic Design, with shipping charges pre paid. Customer shall be responsible for paying all shipping charges, insurance, duties, taxes, and any other charges for products returned to us for any reason.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Blackmagic Design shall not be obliged under this warranty: a) to repair damage resulting from attempts by personnel other than Blackmagic Design representatives to install, repair or service the product, b) to repair damage resulting from improper use or connection to incompatible equipment, c) to repair any damage or malfunction caused by the use of non Blackmagic Design parts or supplies, or d) to service a product that has been modified or integrated with other products when the effect of such a modification or integration increases the time or difficulty of servicing the product.

Exposing URSA Viewfinder to direct sunlight could damage the viewfinder display as the viewfinder optics act as a magnifying glass. Image retention or burn-in could happen on OLED panels when static or high contrast images, such as frame guides, are displayed on the panels for extended periods. To avoid this, ensure the IR sensor for face detection is not covered deliberately and disconnect the viewfinder when not in use for prolonged periods. Image retention is not covered by this product warranty.

THIS WARRANTY IS GIVEN BY BLACKMAGIC DESIGN IN LIEU OF ANY OTHER WARRANTIES, EXPRESS OR IMPLIED. BLACKMAGIC DESIGN AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. BLACKMAGIC DESIGN'S RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS DURING THE WARRANTY PERIOD IS THE WHOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER. BLACKMAGIC DESIGN WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER BLACKMAGIC DESIGN OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES. BLACKMAGIC DESIGN IS NOT LIABLE FOR ANY ILLEGAL USE OF EQUIPMENT BY CUSTOMER. BLACKMAGIC IS NOT LIABLE FOR ANY DAMAGES RESULTING FROM USE OF THIS PRODUCT. USER OPERATES THIS PRODUCT AT OWN RISK.