disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one

r21.1 Changelog

```
+------------------------------------+
|   r21.1 Release - 89669
+------------------------------------+
```

Release day - 18th May 2022

New Features

DSOF-18230 Shot Recording System

What is a shot recording system?

The shot recording system allows users to record and export parameters from within d3, ready for use in post-production workflows. Unlike *device recordings*, which are played back within d3, the data recorded by the shot recorder can be used by external applications.

General requirements

The shot recording system is designed as a generic system to allow users to record any fields or properties displayed in the GUI. Users can build up their own collections of parameters they want to record. In addition, the system has a quick and easy workflow for recording commonly-used sets of parameters, such as camera tracking.

Users can add any fields or properties to a parameter collection for recording where technical limitations allow. Then, as the properties are recorded, they can be viewed and monitored in the GUI. Pre-defined parameter collections are automatically created for camera tracking data, whilst timecodes can be selected and recorded alongside the parameters.

The shot recording system records to an internal format (.shot), which can then be converted to different formats, including JSON, CSV and FBX formats. These recordings can be viewed and managed from within the d3 GUI and can also be started and stopped using the shot recording API.

ParameterCollection

ParameterCollection is a Resource that allows users to create several different collections and switch between them when deciding which parameters to record. It contains an array of ParameterSources, each of which is one ‘parameter’, e.g. field/property to record. ParameterCollections can also contain a list of nested ParameterCollections, allowing users to build up collections in a modular way and add them all to one outer collection which is used for recording.

Some ParameterCollections are auto-created with sets of parameters which are commonly required. For example, each Camera auto-creates a ‘tracking’ collection containing all the parameters commonly required for recreating camera tracking data. Similarly, each SpatialMap also creates a similar tracking ParameterCollection - the main difference is that this accounts for offsets of the world zero point in the spatial map. It also references the currently active camera if using an MR set.

ParameterSources contain a list of field ParameterSources for compound types (e.g. Vec contains x, y and z, Resource can contain its fields). For compound types, the value of the ParameterSource is found by populating the values of its fields. By contrast, the values for basic types are populated using the expression system, which can return floats or strings. Any limitations of the expression system, or particular fields which cannot be accessed by expressions, will also be limitations of the shot recording system.

ShotRecorder

The main components of the ShotRecorder device are:

- The ParameterCollection containing the parameters to be recorded
- A TimecodeTransport used to assign timestamps to the recorded parameters
- Recording and export settings

Recordings can be started or stopped using the ShotRecorder editor, or the shot recording API.

Recordings are saved to an internal format, the .shot file, which is designed to be a compact way of recording the data. Values are recorded every frame, but only when they change to limit the size of the data files. The files can then be converted to various more useful formats at any time after recording. Separating the recording of the data from the conversion to file formats allows the recording to be made as efficient as possible and allows us to make updates to the exported formats or add a new format which can then be applied to old recordings.

Setting up the shot recording system

1. Create a ShotRecorder device in the current DeviceManager. This device will become the master of the shot recording system
2. Name the Slate for the current shot
3. Create a new ParameterCollection in the ‘Parameters’ field and add the desired parameters. ParameterCollections are a collection of different parameters that we are interested in recording and getting the data out of i.e. camera tracking, MR set etc.
4. Add the desired Timecode provide to the ‘Timecode’ field
5. The ‘Take Snapshot’ and ‘Take Screenshot’ options can be selected to take a project snapshot and/or screenshot at the end of each take.

To add parameters to a ParameterCollection for recording:

1. Open the ParameterCollection editor
2. Alt+drag from the Sources '+' button to any field in the UI to add it as a parameter source
3. Several ParameterCollections are automatically created for convenience, such as camera tracking collections. These, or other user-created ParameterCollections, can be added to the ParameterCollections list in the editor to allow the parameters to be organised into smaller separate collections.
4. Click the arrow on the right hand side of a parameter to expand it and view its fields. Resource parameters do not have their fields added by default, but they can be added by clicking the plus button on the right hand side of the parameter.
5. Timecode can be set up inside the shot recorder in order to reference timecode in the recordings using TimeCodeTransport

disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one

To make a recording:

1. To start a recording, click the button displaying ‘Disengaged’ to engage the recorder and start recording.
2. Click the same button again to stop the recording. The take number will automatically increment, or it can be changed manually.
3. Alternatively, use the API commands documented at http://localhost/docs/v1/index.html to start and stop recordings externally
4. Slates will be listed under the ‘Recordings’ separator. Right click on a slate to see the takes for that slate.

To export recordings:

1. Select the desired formats under the ‘Export’ tab
2. Either click ‘Export all recordings’, or export recordings individually under the ‘Recordings’ tab
3. Select ‘Auto-export’ to automatically export in the selected formats when each take is finished
4. Click ‘Export comparison table’ to export a CSV table comparing the first value of each parameter for each recording
5. You can also enable to take a snapshot or a screenshot of each recording if desired

Recordings can currently be exported in the following formats:

- JSON: A compact JSON representation which contains the changed parameter values for each timestamp, similar to the .shot file
- CSV (Compact): A compact CSV representation which contains only the changed parameter values for each timestamp
- CSV (Dense): A dense CSV representation which contains all values for each parameter at each timestamp
- FBX: An FBX file containing meshes and pose information (position, rotation, scale etc.) of all recorded objects, where available. Any additional properties not represented within the FBX specification are recorded as custom properties.

When going into the d3 projects folder, a new sub-folder will be created called ‘output’. Within there we have ‘shots’ and in there will be the slate. In the slate, there will be the .shot recording in addition to the recording in the chosen format

Limitations

- The shot recording system only writes data out of d3. There is no capability to read recordings back into d3 directly to replay the data. This is what device recordings should be used for.
- The shot recording system and the device recording systems do not interact with each other. They can still both be used at the same time.
- There is not currently an easy way to get all the recorded camera tracking data into Unreal Engine to re-render shots. Epic have outlined a workflow here to import and export FBX files which allows camera positions, rotations and FoV to be imported.

Additional information

Elements/Fields added

- Added a new VirtualCamera object, with its own editor
- Added additional fields to an AnimateCameraControl layer when a VirtualCamera is selected
- Added 'VirtualCameraPosition' object which can be selected in an AnimateCameraPreset layer

Relevant tooltips have been added to the editors of ShotRecorder and ParameterCollection. Some general improvements to the expressions system have been made. It is important to note that the shot recording system will have an additional performance load whilst recording.

We have also built a plug-in for the Unreal Switchboard, which can trigger the Shot Recorder when takes are triggered within that system.

The Unreal Switchboard plug-in is needed in order to perform a shot record when the 'take' button in the switchboard is activated. This can be downloaded here (https://download.disguise.one/#resources) and the file should be copied to:

UE install>\Engine\Plugins\VirtualProduction\Switchboard\Source\Switchboard\switchboard\devices\ and ensure a connected Disguise device can control Shot Recording.

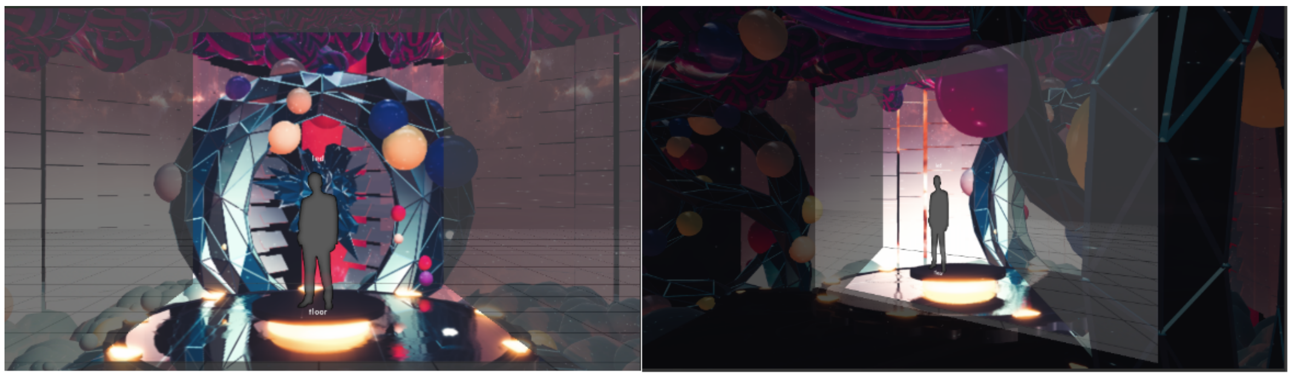
disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one

DSOF-16989 Virtual Camera workflow improvements

What is a virtual camera?

Virtual cameras can be used as part of mixed reality or green screen workflows to allow shots to be captured from positions outside of the range of the physical cameras. By reprojecting the filmed action and virtual content to a new viewpoint, wide shots can be captured for small stages with limited space.

A virtual camera can also be used to traverse a large virtual scene, such as a cityscape. It is possible to start by moving through the streets rendering the virtual scene, and end by transitioning into the view of the real camera on an xR stage, capturing the people on the stage within your virtual scene.

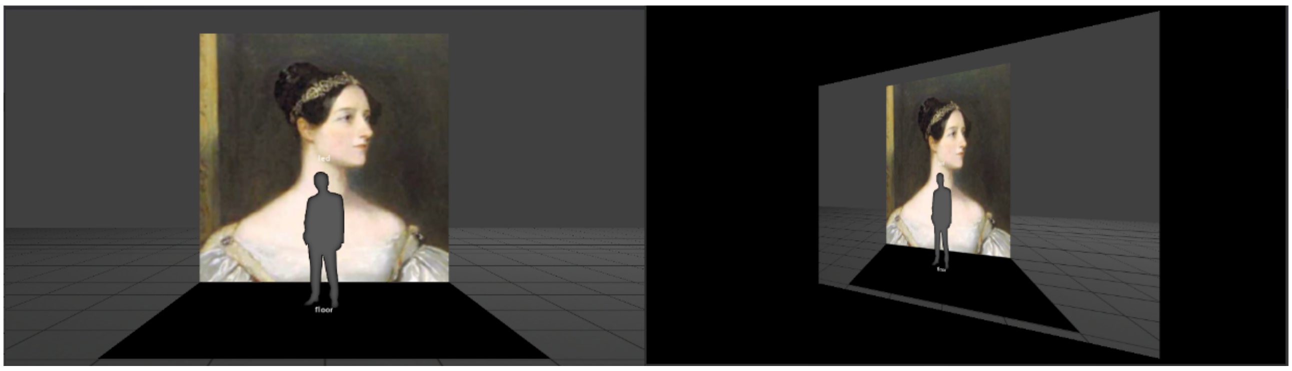


General requirements

Virtual cameras are designed to be interchangeable with real cameras as far as possible for a MR/green screen setup. They should be interacted with in similar ways to a real camera. Virtual cameras appear as an object in the stage, similar to a real camera, and can be linked to a ‘parent camera’, which is the real camera providing the filmed content from the stage.

Content can be mapped to the front and back plates of a virtual camera, and can be used in mixed reality sets in the same way as a real camera. Users can also control the position of a virtual camera in the same ways that are possible for a real camera.

Virtual camera positions can be defined in Global coordinates, or relative to the parent camera. Virtual camera positions can also be keyframed using AnimateCameraControl and AnimateCameraPreset layers.



Reprojection

Note that reprojection is done slightly differently to a real camera. For a real camera, the set extension is reprojected onto the set extension mesh to match the filmed content in order to avoid any discontinuities at the boundaries.

For a virtual camera, the filmed content is reprojected using a planar reprojection to match the fixed set extension in order to ensure the set extension remains stable when the real camera moves.

This means that there may be more obvious discontinuities between filmed content and set extension when using a virtual camera and moving its parent camera than when moving a real camera.

This effect will be more obvious the closer the virtual camera's view is, so it is recommended not to move the parent camera quickly when using a virtual camera which is only offset by a small amount from its parent.

Setting up the virtual camera:

1. Create a VirtualCamera in the Cameras list in the stage.
2. Set the 'Parent camera' property to the desired real camera.
3. Add a 'Live action position marker' located at the average position of live content on the stage (e.g. people). This will ensure that the live content appears with the correct perspective relative to the virtual content when the virtual camera moves.
4. 'Max zoom in factor' can be set to restrict how much the filmed content can be zoomed into before purely virtual content is displayed. This prevents excessive pixelation when moving the virtual camera towards the live action.
5. Set the lens settings under Physical->Lens. These can be defined locally in the virtual camera, set to follow the zoom of the parent camera (with an optional scaling applied), or set to follow all the intrinsics of the parent camera (with an optional zoom scale).

To control a virtual camera:

There are various ways for controlling the position of the virtual camera. You can either:

- Use the object manipulators to move the VirtualCamera object around in the stage
- Edit the values in the VirtualCamera editor. Note that values can be defined in a Global coordinate system, or Relative to the parent camera.
- Use a tracking source: A tracking source can be used to control the pose and lens intrinsics for a virtual camera in the same way as a real camera.
- Keyframe camera animations using AnimateCameraPreset or AnimateCameraControl.

To keyframe virtual camera animations:

Virtual camera animations can be keyframed similarly to standard cameras using the AnimateCameraControl or AnimateCameraPreset layers.

- AnimateCameraControl:
  a. Add an AnimateCameraControl layer
  b. Set the camera to the VirtualCamera you wish to animate
  c. Set the coordinate system to Global or Relative as desired
  d. Keyframe virtual camera properties. In addition to the standard properties for a normal camera, zoom scale can also be keyframed
- AnimateCameraPreset:
  a. Add an AnimateCameraPreset layer
  b. Set the camera to the VirtualCamera you wish to animate
  c. Add new VirtualCameraPositions to the Position property
  d. Set the coordinate system to Global or Relative as desired
  e. In addition to the standard properties for a normal camera, the virtual camera's zoom can be set. Toggle the 'Animate zoom scale' option to choose whether to animate zoom scale or global field of view
  f. Keyframe several VirtualCameraPositions to animate the camera between these positions. Positions in Global and Relative coordinates can be mixed, and the camera will move between the coordinate systems.

Limitations:

- If the virtual camera is significantly off-axis from the real camera, the live objects (e.g. people) will become skewed. The projection can be thought of as looking at a 2D cardboard cutout from a different angle.
- All live action must be completely within the frame captured by the real camera. If not, it will be cut off during the reprojection.
- If the lens is heavily distorted, there may be distortion artefacts. A radial mask can be set in the editor to remove these effects, which are most pronounced at the edges of the image.
- If you attempt to move the virtual camera forwards towards the live action, the picture will become progressively more pixelated. This is because you are zooming on the captured image at fixed resolution. This can be mitigated by setting the 'Max zoom in factor'.
- If the virtual camera moves behind the live action, it will render virtual content only.

- If the parent camera moves quickly, there may be some noticeable discontinuity between filmed content and the set extension. This may be more noticeable if the virtual camera is near the parent camera. For fast camera moves it is recommended to use a real camera rather than a virtual camera.

Legacy workflow

Virtual camera animations could previously be keyframed using additional options in the AnimateCameraControl or AnimateCameraPreset layers. The new workflow supersedes this method, however for backwards compatibility the old workflow can be enabled by setting the option switch *enableLegacyVirtualCameraWorkflow*.

Additional information

Elements/Fields added

- Added a new VirtualCamera object, with its own editor
- Added additional fields to an AnimateCameraControl layer when a VirtualCamera is selected
- Added 'VirtualCameraPosition' object which can be selected in an AnimateCameraPreset layer

Relevant tooltips have been added to the new elements. The previous workflow is deprecated, but can be enabled by setting the option switch *enableLegacyVirtualCameraWorkflow*.

Feature Request

DSOF-14841 Use RenderStream compressed to stream low-res proxies of RenderStream sources to Director for previsualisation

We have created a low-resolution proxy of RenderStream compressed that the Director and Editor machines can subscribe to for previsualisation.

The previous workflow for this was for the Director machine to subscribe to all of the RenderStream uncompressed streams in order to provide a previsualisation of that feed(s).

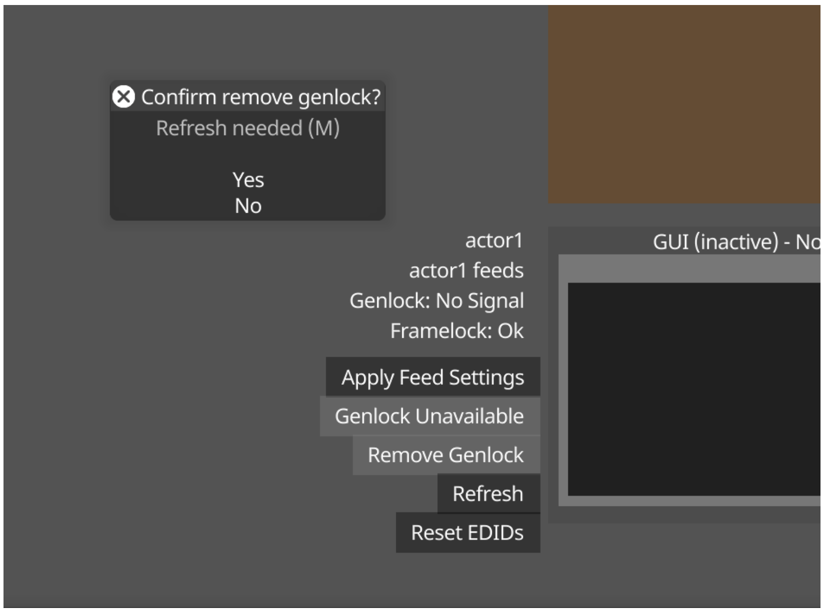
The advantage of this feature is that a preview feed can now be delivered without saturating the network bandwidth and/or the servers with RenderStream uncompressed feeds for these roles

This option can be found in the Devices dropdown / Proxy streams. A scale factor can also be set.

DSOF-11229 Ability to remove video output sync (genlock) within d3

We have added 'Remove Genlock' to the feed settings. A confirmation will pop up with the option to refresh as seen below:

disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one



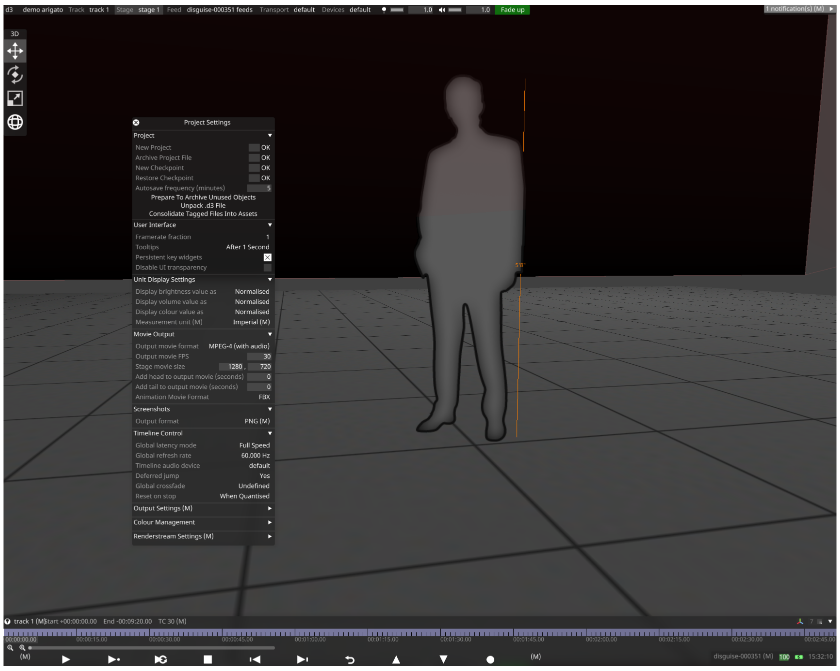
Confirm remove genlock?
Refresh needed (M)

actor1
actor1 feeds
Genlock: No Signal
Framelock: Ok

DSOF-9657 Simple Text Layer

DSOF-696 Imperial Measurements

We have enabled the option within d3 to switch between metric and imperial measurements. This can be found in Project Settings / Unit Display Settings / Measurement unit as seen below:



Project Settings

Project	
New Project	OK
Archive Project File	OK
New Checkpoint	OK
Restore Checkpoint	OK
Autosave frequency (minutes)	5
Prepare To Archive Unused Objects	
Unpack .d3 File	
Consolidate Tagged Files Into Assets	
User Interface	
Framerate fraction	1
Tooltips	After 1 Second
Persistent key widgets	☒
Disable UI transparency	☐
Unit Display Settings	
Display brightness value as	Normalised
Display volume value as	Normalised
Display colour value as	Normalised
Measurement unit (M)	Imperial (M)
Movie Output	
Output movie format	MPEG-4 (with audio)
Output movie FPS	30
Stage movie size	1280 , 720
Add head to output movie (seconds)	0
Add tail to output movie (seconds)	0
Animation Movie Format	FBX
Screenshots	
Output format	PNG (M)
Timeline Control	
Global latency mode	Full Speed
Global refresh rate	60.000 Hz
Timeline audio device	default
Deferred jump	Yes
Global crossfade	Undefined
Reset on stop	When Quantised
Output Settings (M)	
Colour Management	
Renderstream Settings (M)	

d3 demo arigato Track track 1 Stage stage 1 Feed disguise-000351 feeds Transport default Devices default
track 1 (M) start +00:00:00.00 End -00:09:20.00 TC 30 (M)
disguise-000351 (M)

DSOF-16692 Add the ability for each RenderStream Textures to be sourced from different actors

It was requested by users working with vx machines as actors alongside a cluster of rx machines to source content from different actors so that a single machine doesn’t have to stream all of the video inputs/content to the render nodes.

We have therefore created a module called TextureParameterAssigner that contains the parameter and an assigner to control the transport format. This can be extended with a Machine object to control which machine is responsible for outputting the texture.

We have also reflected this in the proxy levels system so that only the appropriate selected machines render the texture inputs as appropriate.

Improvements

DSOF-20318 Warn when EDID timings don't match

We have added a warning if users have different EDID timings when trying to genlock. In order to genlock, these need to be the same across the output heads. In previous releases, if the EDID timings were different then genlock would fail without any indication as to why it had failed.

DSOF-20205 OscControl: rename OscCommand's address to "OSC address" that distinguishes it from the IP Address

DSOF-20153 More informative "zero-sized texture being used" notification

We have added the full file path and explicitly noted whether there is a missing file to the “zero-sized texture being used” notification.

DSOF-20098 Improve performance of NotFoundMediaSystem

We have added a “Checking media” progress bar when performing project upgrades.

DSOF-20083 Make default VFC to be HDMI rather than DVI

DSOF-19946 Improve Project / Upgrade Load Time

We have improved the load time of upgrading large projects to later releases.

DSOF-19865 [ACES] CDL functionality outside ACES mode

In r21.0 we released CDL layers and objects, which only work when a user is in ACES mode. We have added an error message when a user is creating a CDL layer and also turns it red. We have then added a tooltip that explains why the CDL layer is red when not in ACES mode.

DSOF-19785 RenderStream Compressed: increase target bits/sec when using alpha plane

We have increased the alpha channel of RenderStream Compressed from 25 Mbps at 1080p 60 to 32 Mbps.

disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one

DSOF-19645 Increase default resolution maximum of VFC cards from 5120x5120

We have increased the resolution in VFCPortConfig from 5120 pixels to 8192 pixels.

DSOF-19547 Left-click on layers in the layer stack to immediately open the layer editor

DSOF-19546 Notes: seen/unseen states

In r21.0 we added the notes feature. We have now added icons to notify users whether a note is unseen or updated.

DSOF-19535 Simplify and fix SlugFont implementation

DSOF-19318 Add support for alpha in 4:2:2 RenderStream uncompressed format

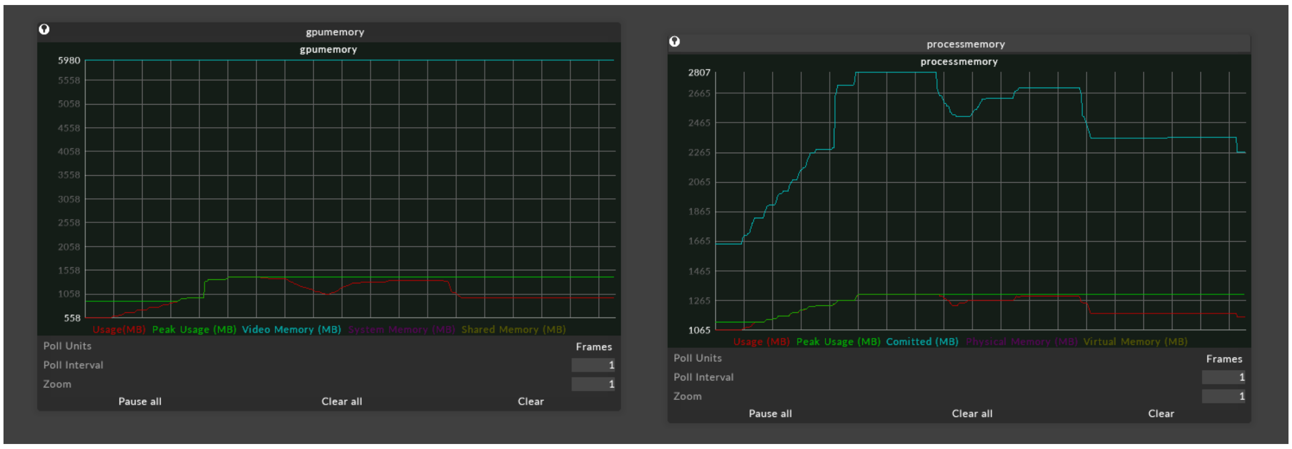
We have added support for alpha in 4:2:2 chroma subsampling mode for RenderStream uncompressed. Previously, the only format which supported RenderStream uncompressed alpha is RGBA mode, which is 4 channels per pixel.

DSOF-19065 RenderStream: Include asset schema JSON in diagnostics

DSOF-18602 Improve process and gpu memory graphs to better represent state of memory usage

The GPU graph has been improved to show total physical and virtual memory available on the machine, in addition to historic peak values to allow more careful management of resources. It also includes Notch and Notch LC memory usage as well as texture allocations.

An example of where this could be used is a long running project where it may not be possible to monitor each session. An outline can be seen below:



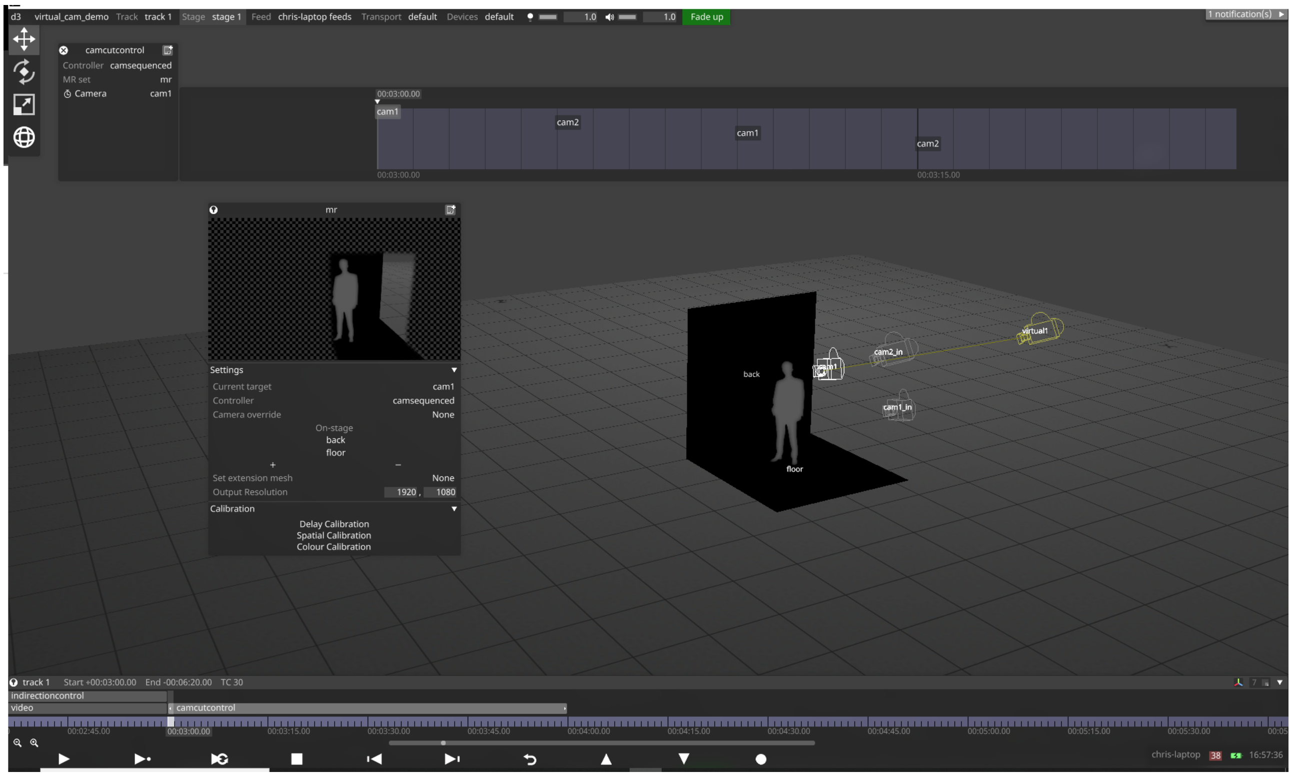
DSOF-18258 Camera switching control module

In previous releases, users experienced difficulties to time an xR camera switch exactly as there is a delay between pressing the button and the switch occurring in the output.

This led to users having to try and estimate the switching latency themselves and trigger the switch an appropriate amount of time in advance.

We have therefore created a CameraCutControl module to allow camera switches to be sequenced on the timeline for xR workflows. It takes into account the calibrated delays to ensure that the switch happens on a keyframed beat by calculating the total switching delay for any given frame, and uses that to work out when to trigger the indirection change so that it lands on the correct beat.

This module can be found by first adding a SequencedInDirectionController as the camera indirection control in the mixed reality set. From here, you can add a CameraCutControl module to the timeline. Select the controller and mixed reality set, and then add keyframes to the camera property.


d3 virtual_cam_demo Track track 1 Stage stage 1 Feed chris-laptop feeds Transport default Devices default 1.0 1.0 Fade up
1 notification(s)
00:03:00.00
cam1
cam2
cam1
cam2
00:03:15.00
back
floor
cam2_in
cam1_in
virtual1
track 1 Start +00:03:00.00 End -00:06:20.00 TC 30
indirectioncontrol
video
camcutcontrol
chris-laptop 16:57:36

camcutcontrol
Controller camsequenced
MR set mr
Camera cam1

mr
Settings
Current target cam1
Controller camsequenced
Camera override None
On-stage
back
floor
Set extension mesh None
Output Resolution 1920 , 1080
Calibration
Delay Calibration
Spatial Calibration
Colour Calibration

DSOF-18234 API control for shot recording

DSOF-17374 Expanded LUT file support (including 1D LUTs)

We have enabled 1D LUT file support in Designer to support linear colour reproduction when using industry-standard solutions such as CalMan, which requires a 1D LUT file to adjust brightness, contrast, and gamma in addition to 3D LUT files to adjust colours.

DSOF-17343 Camera bookmarks for the feed view

The camera bookmarks associated with the F keys are now shared between the Stage and Feed views.

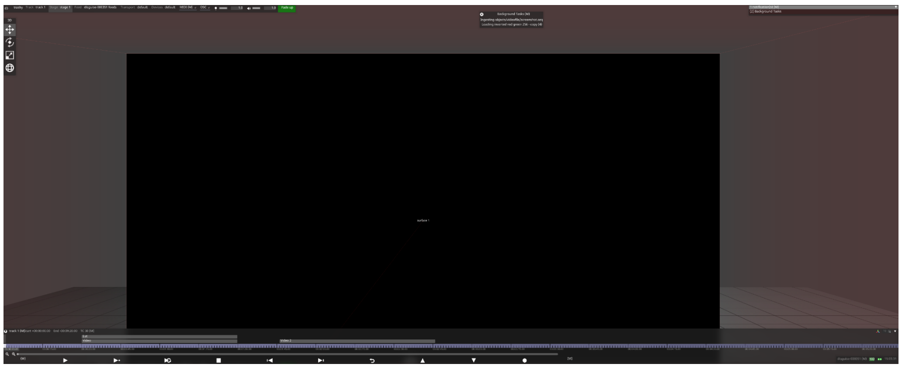
DSOF-16686 Avoid pixelation when doing virtual camera moves through the reprojection plan

In previous releases, if a user did a virtual camera movement which goes through the reprojection plan (i.e. past the live action position market), the content got very pixelated. This is due to the fact that users are zooming on the camera image that is captured at a fixed resolution. We have set a 'maximum zoom' value on a virtual camera to avoid this issue.

DSOF-12111 Make isCefPaintDirtyRectangles behaviour default and remove option switch

DSOF-9531 Listing of background tasks

We have added a 'Background Tasks' count in the notifications to enable users to see all active and queued background tasks to support diagnosis of performance issues, and also to get visibility into incomplete media.



Ingesting objects/videofile/screenshot.seq
Loading inverted red green 256 - copy (4)

DSOF-8456 Network status widget should remember its position

In previous releases, the floating network status widget didn't remember its position, which caused the view of the stage to be unnecessarily obscured. This has now been resolved.

DSOF-8390 Notify user of that joining a session hosted by a different version of d3 is available, but not recommended or supported

We have added a notification on each actor machine in a session using the notification bar warning the user that the given machine is of a different version than the director machine.

DSOF-4298 Inclusion of timestamp to notifications

In r21.1 we have added timestamps to notifications.

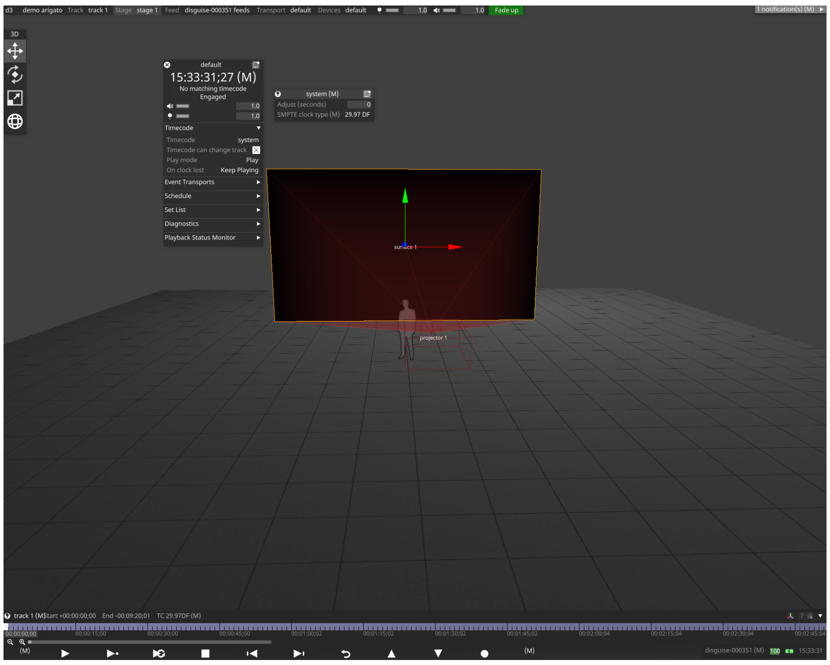
DSOF-4253 Ability to define image sequence frame rate automatically

We have added the ability to generate both metadata.txt and .xmp data from an image sequence

DSOF-3477 Selecting FPS for the system time transport control

We have added the ability to change what FPS the system time transport control runs. In previous versions, if the system time transport control was selected it forced the timeline to be in 30fps.

disguise Technologies
88-89 Blackfriars Road
London
SE1 8HA
United Kingdom
info@disguise.one
www.disguise.one



default
15:33:31;27 (M)
No matching timecode
Engaged

Timecode
Timecode system
Timecode can change track
Play mode Play
On clock lost Keep Playing
Event Transports
Schedule
Set List
Diagnostics
Playback Status Monitor

system (M)
Adjust (seconds) 0
SMPTE clock type (M) 29.97 DF

track 1 (M) Start +00:00:00;00 End -00:09:20;01 TC 29.97DF (M)

Fixes

DSOF-20666 - Fixed an issue where the virtual camera disappeared from the stage when tracking source is used

DSOF-20658 - Rendering Regression: Fixed no startup text, and flickering visualiser

DSOF-20656 - Fixed an issue where exceptions on startup could cause unclean shutdown and risk project corruption

DSOF-20654 - Fixed transport controls on separate lines from HUD status at 3840 * 2160 with 250% dpi scaling

DSOF-20621 - Fixed an issue where the recorder icon in d3 state bar had a massive border on addition

DSOF-20619 - Fixed an issue where 'Show 3D Observations' only showed text labels

DSOF-20617 - Fixed an issue where spatial calibration produced bad results

DSOF-20616 - Fixed an issue where spatial calibration didn't run for all screens in MR set

DSOF-20615 - Fixed an error when clicking 'Reset all Observations' in spatial calibration

DSOF-20608 - Fixed an issue where text seemed vertically squashed with high dpi scale

DSOF-20542 - Fixed an issue where Independent Editor could disengage transport on director

DSOF-20541 - Fixed an issue where there were device conflicts and potential crashes with two device managers

DSOF-20492 - Virtual camera can no longer be added as a parent camera

DSOF-20478 - Fixed an issue where values within Patch Assignments widget offset at random

DSOF-20447 - Fixed an Access Violation when opening DmxPersonality widget

DSOF-20433 - Fixed an Access violation when removing unassigned streams

DSOF-20388 - Fixed an error when adding fields of camera in parameter collection

DSOF-20380 - Fixed an issue where linking a Collection to a Source created an Arraybox error notification

DSOF-20365 - Fixed an issue where setting the resolution of a surface referenced by a mesh mapping to zero would cause d3 to hang

DSOF-20321 - Fixed the Missing Media widget being too narrow to read long filenames

DSOF-20306 - Fixed an Access Violation when running Colour Calibration

DSOF-20305 - Fixed an issue where mapping plane / wireframe were not drawn for most geometric mappings

DSOF-20282 - RenderStream: Fixed an issue where the stream crashes after a while when there is a proxy stream in the assigner

DSOF-20275 - Fixed an issue where d3 locks up when ingesting large quantities of media

DSOF-20274 - - Fixed a Python error "Cannot convert from 'int' to 'Colour'" when creating QuickAlign point

DSOF-20256 - Now ensure clean shutdown of s3 main thread using RAII

DSOF-20251 - Fixed an issue where it was not possible to open list editor for Projectors and Cameras

DSOF-20248 - Fixed an exception when attempting to open Patch Assignments widget

DSOF-20182 - Fixed a crash when attempting to jump to timecode

DSOF-20154 - AnimateCameraPreset: Copy values button is no longer under the Virtual Camera separator

DSOF-20148 - Fixed an issue where resizing a module from the very start of the track pushes the module to the right, past the end of the track

DSOF-20147 - Fixed an issue when adding a new string into the Slate's input the Take number did not revert back to one

DSOF-20136 - Proxy streams are now included in bandwidth estimates

DSOF-20125 - 3D Perspective and parallel projections now use content viewpoint returned from Renderstream

DSOF-20121 - Fixed an issue where mapping was no longer directed at surface 1 and displayed a "Bad Projection Used" error

DSOF-20102 - Authorisation: reduced CPU spikes by moving time checks outside of CriticalSection

DSOF-20077 - Fixed an issue where the console window did not scroll to the end of the log

DSOF-20069 - VideoIn: Fixed unresponsive UI after being mapped to a stage camera video in

DSOF-20066 - Fixed an issue where multiple warp editors can be opened for the same feed rectangle

DSOF-20065 - Fixed an issue where feed mapping buttons have lost their backgrounds

DSOF-20063 - Fixed an issue where the feed mapping edit multiple table contained parameters that did not apply to feed mappings

DSOF-20058 - Fixed a crash on startup when director was set to local machine

DSOF-20053 - Fixed an issue where dynamic blend did not work when projector is using MSAA

DSOF-20030 - Fixed an issue where ObjectView's that were filtered via search did not display items in relevance order

DSOF-20020 RenderStream: Fixed Receive Health widget not clearing each time a workload was started

DSOF-20017 RenderStream: Fixed an issue where tests showed a symbol error about d3renderstreamsend.dll

DSOF-20007 - Choose Action button is now correctly labelled for RenderStream workloads

DSOF-19990 - Fixed an issue where Trash Icons overlapped text within devices

DSOF-19976 - Fixed an issue where "Remove Genlock" did not update the indicator of genlock status on Nvidia Machines

DSOF-19966 - Bulk edit for videoclips is now laid out better and contains appropriate fields

DSOF-19957 - Fixed an Access Violation when editing VideoIn Input Transform after defocusing VideoIn

DSOF-19842 - Fixed an issue where a project could hang at "Scanning Package" upon start

DSOF-19835 - Improved Renderstream active latency rounding up

DSOF-19829 - Fixed an issue where 'Power' in a CDL widget brought up the wrong widget of values

DSOF-19807 Video: Fixed CDL being active when ACES is disabled

DSOF-19804 - Fixed an issue where the video input patch editor's widget was minimized when stopping a preview

DSOF-19787 - Fixed an issue where greyed out resource fields could be selected

DSOF-19697 - Fixed an issue where StageRender layer only works with 3D mappings

DSOF-19690 - Fixed Timeline Control section containing non timeline related settings

DSOF-19687 - Fixed RGB Colour module having Input Transform option in Gamma mode

DSOF-19638 ACES: Fixed Input Transform field appearing for videoins regardless of Colour Management value

DSOF-19554 - Fixed thumbnail resolution for a certain types of resources

DSOF-19488 Mesh: Fixed PosIndex comparison operators > and < being incorrect

DSOF-19280 - Fixed a delay and main thread hang when switching video modes with capture cards

DSOF-19174 - Fixed Expressions referencing fields of other fields passed by value return garbage

DSOF-19146 - Fixed an issue when setting Venue to None or Changing Venue removes parent child relationships from props within it

DSOF-19025 - Fixed an issue where Video In Placeholder Clips could not be NotchLC files

DSOF-18996 - Fixed an Access violation when taking shot with no timecode selected

DSOF-18969 - Fixed an issue where the network widget will always be fully expanded when d3 is started

DSOF-18669 - Fixed vertical alignment of text not being properly centered

DSOF-18508 - Fixed Parameter collection UI not showing enum values as a string

DSOF-18409 - Fixed an Access violation when shot recorder links to 'None' resource

DSOF-18377 - Fixed multiple Access violations when adding arrowing new parameters into collection

DSOF-18289 XR: Fixed an issue where it was not possible to remove alignment points when changing mesh of Alignment Object

DSOF-18271 - Fixed an issue where importing .dpx files in d3 was giving the wrong colour shift

DSOF-18241- Fixed n Access Violation if new content is deleted from the Feed view if old content is viewed in the Edit Warp widget

DSOF-18225 - Fixed an issue where the Mapping field was not controllable via DMX if set to "None"

DSOF-17605 - Fixed an issue with 10bit feed settings requiring a restart of d3

DSOF-12071 - Fixed an issue with Video clip editor showing incorrect value for bits per channel for NotchLC clips

DSOF-9815 - Fixed an issue where fade up/down/hold with an editor in independent mode affected the master