<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer Types Overview</td>
<td></td>
</tr>
<tr>
<td>Types of layers</td>
<td>110</td>
</tr>
<tr>
<td>Layers overview</td>
<td>115</td>
</tr>
<tr>
<td>Types of layers</td>
<td>115</td>
</tr>
<tr>
<td>Controlling layer properties with ArtNet</td>
<td>115</td>
</tr>
<tr>
<td>Controlling layer properties with OSC</td>
<td>115</td>
</tr>
<tr>
<td>Creating layers</td>
<td>116</td>
</tr>
<tr>
<td>Editing layers</td>
<td>119</td>
</tr>
<tr>
<td>What is the Layer editor?</td>
<td>119</td>
</tr>
<tr>
<td>Opening / closing the Layer editor</td>
<td>119</td>
</tr>
<tr>
<td>Layer properties</td>
<td>119</td>
</tr>
<tr>
<td>MultiLayer Editor</td>
<td>120</td>
</tr>
<tr>
<td>Common layer properties</td>
<td>123</td>
</tr>
<tr>
<td>Preview Thumbnail</td>
<td>124</td>
</tr>
<tr>
<td>Brightness</td>
<td>125</td>
</tr>
<tr>
<td>Content</td>
<td>126</td>
</tr>
<tr>
<td>Content properties</td>
<td>127</td>
</tr>
<tr>
<td>Mapping</td>
<td>134</td>
</tr>
<tr>
<td>Blend Mode</td>
<td>136</td>
</tr>
<tr>
<td>Colour Shift</td>
<td>140</td>
</tr>
<tr>
<td>Palette</td>
<td>143</td>
</tr>
<tr>
<td>xCol yCol</td>
<td>144</td>
</tr>
<tr>
<td>Controlling layers with Art-Net</td>
<td>145</td>
</tr>
<tr>
<td>How to control layer properties with Art-Net</td>
<td>145</td>
</tr>
<tr>
<td>How to enable the layer property to work only between the range 0-1</td>
<td>147</td>
</tr>
<tr>
<td>Controlling layers with OSC</td>
<td>148</td>
</tr>
<tr>
<td>Expressions</td>
<td>151</td>
</tr>
<tr>
<td>Layer Types Overview</td>
<td>153</td>
</tr>
<tr>
<td>Types of layers</td>
<td>153</td>
</tr>
<tr>
<td>Layer Types Overview</td>
<td>158</td>
</tr>
<tr>
<td>Types of layers</td>
<td>158</td>
</tr>
<tr>
<td>Audio</td>
<td>162</td>
</tr>
<tr>
<td>Bitmap</td>
<td>165</td>
</tr>
<tr>
<td>TestPattern</td>
<td>184</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Video</td>
<td>202</td>
</tr>
<tr>
<td>Legacy video</td>
<td>222</td>
</tr>
<tr>
<td>RenderStream</td>
<td>242</td>
</tr>
<tr>
<td>Overview</td>
<td>242</td>
</tr>
<tr>
<td>VideoTrigger</td>
<td>244</td>
</tr>
<tr>
<td>Web</td>
<td>263</td>
</tr>
<tr>
<td>CameraControl</td>
<td>284</td>
</tr>
<tr>
<td>CameraControl layer properties</td>
<td>284</td>
</tr>
<tr>
<td>Creating/sequencing CameraControl commands</td>
<td>284</td>
</tr>
<tr>
<td>Editing CameraControl commands</td>
<td>285</td>
</tr>
<tr>
<td>Control</td>
<td>287</td>
</tr>
<tr>
<td>Command Syntax</td>
<td>287</td>
</tr>
<tr>
<td>Ascii examples</td>
<td>288</td>
</tr>
<tr>
<td>Hex examples</td>
<td>289</td>
</tr>
<tr>
<td>DMXLightsControl</td>
<td>290</td>
</tr>
<tr>
<td>DMXLightsControl layer properties</td>
<td>290</td>
</tr>
<tr>
<td>Creating DMXLightsControl commands</td>
<td>291</td>
</tr>
<tr>
<td>Editing DMXLightsControl commands</td>
<td>291</td>
</tr>
<tr>
<td>DMXShare</td>
<td>292</td>
</tr>
<tr>
<td>DMXShare layer properties</td>
<td>292</td>
</tr>
<tr>
<td>MatrixControl</td>
<td>293</td>
</tr>
<tr>
<td>MatrixControl layer properties</td>
<td>293</td>
</tr>
<tr>
<td>Creating &amp; sequencing MatrixControl commands</td>
<td>294</td>
</tr>
<tr>
<td>Using the MatrixControl layer</td>
<td>295</td>
</tr>
<tr>
<td>TimecodeMode</td>
<td>296</td>
</tr>
<tr>
<td>MasterBrightness</td>
<td>297</td>
</tr>
<tr>
<td>MasterBrightness layer properties</td>
<td>298</td>
</tr>
<tr>
<td>MidiNote</td>
<td>302</td>
</tr>
<tr>
<td>MidiNote layer properties</td>
<td>302</td>
</tr>
<tr>
<td>MDC</td>
<td>303</td>
</tr>
<tr>
<td>MTC</td>
<td>304</td>
</tr>
<tr>
<td>MTC Layer Properties</td>
<td>304</td>
</tr>
<tr>
<td>Open</td>
<td>306</td>
</tr>
<tr>
<td>OscControl</td>
<td>309</td>
</tr>
<tr>
<td>Command Syntax</td>
<td>310</td>
</tr>
<tr>
<td>Property</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------</td>
</tr>
<tr>
<td>PlayMode</td>
<td>311</td>
</tr>
<tr>
<td>PlayMode Properties</td>
<td>311</td>
</tr>
<tr>
<td>TrackJump</td>
<td>312</td>
</tr>
<tr>
<td>TrackJump Layer Props</td>
<td>312</td>
</tr>
<tr>
<td>TransportControl</td>
<td>313</td>
</tr>
<tr>
<td>TransportBrightnessLocal</td>
<td>314</td>
</tr>
<tr>
<td>TransportVolumeLocal</td>
<td>315</td>
</tr>
<tr>
<td>Blur</td>
<td>316</td>
</tr>
<tr>
<td>Common Layer Properties</td>
<td>317</td>
</tr>
<tr>
<td>Blur layer properties</td>
<td>325</td>
</tr>
<tr>
<td>ChannelRouter</td>
<td>326</td>
</tr>
<tr>
<td>ColourAdjust</td>
<td>338</td>
</tr>
<tr>
<td>Overview</td>
<td>338</td>
</tr>
<tr>
<td>Workflow</td>
<td>338</td>
</tr>
<tr>
<td>Common layer properties</td>
<td>339</td>
</tr>
<tr>
<td>Compose</td>
<td>349</td>
</tr>
<tr>
<td>EdgeFilter</td>
<td>364</td>
</tr>
<tr>
<td>Fade</td>
<td>376</td>
</tr>
<tr>
<td>FilmicGrain</td>
<td>379</td>
</tr>
<tr>
<td>Overview</td>
<td>379</td>
</tr>
<tr>
<td>Workflow</td>
<td>379</td>
</tr>
<tr>
<td>Common layer properties</td>
<td>380</td>
</tr>
<tr>
<td>Kaleidoscope</td>
<td>389</td>
</tr>
<tr>
<td>Lut</td>
<td>399</td>
</tr>
<tr>
<td>Overview</td>
<td>399</td>
</tr>
<tr>
<td>Workflow</td>
<td>399</td>
</tr>
<tr>
<td>Common layer properties</td>
<td>400</td>
</tr>
<tr>
<td>Lut layer properties</td>
<td>408</td>
</tr>
<tr>
<td>MotionBlur</td>
<td>411</td>
</tr>
<tr>
<td>Noise</td>
<td>423</td>
</tr>
<tr>
<td>Pixelate</td>
<td>433</td>
</tr>
<tr>
<td>PixelMap</td>
<td>449</td>
</tr>
<tr>
<td>Scroll</td>
<td>463</td>
</tr>
<tr>
<td>SpinBitmap</td>
<td>475</td>
</tr>
<tr>
<td>Strobe</td>
<td>490</td>
</tr>
</tbody>
</table>
Trigger ........................................................................................................... 503
Bugs .................................................................................................................. 516
Chevron ............................................................................................................ 529
Colour .............................................................................................................. 542
Generative Layers ............................................................................................ 554
Gradient .......................................................................................................... 555
Notch ............................................................................................................... 568
Overview ......................................................................................................... 568
Workflows ....................................................................................................... 570
Notch layer properties ..................................................................................... 575
Radar ................................................................................................................. 579
RGBColour ..................................................................................................... 592
Scan .................................................................................................................. 602
Tennis ............................................................................................................... 612
Options .............................................................................................................. 623
Control ............................................................................................................. 624
Readout ............................................................................................................. 625
AnimateCameraControl .................................................................................. 639
AnimateCameraPreset ..................................................................................... 642
Creating a camera position animation ............................................................ 642
Previsualisation Layers Overview ................................................................... 645
AnimateObjectPreset ....................................................................................... 646
Creating a AnimateObjectPreset animation .................................................... 647
TargetPreset ..................................................................................................... 651
Targeting/animating fixtures .......................................................................... 652
Opening the Fixture Targets editor ................................................................. 653
TargetObject .................................................................................................... 655
TargetControl .................................................................................................. 657
Bitmap .............................................................................................................. 660
Legacy video .................................................................................................... 679
Keyframing overview ...................................................................................... 699
What is the Keyframe editor? ......................................................................... 699
Opening a Keyframe editor ............................................................................ 699
Keyframe toggles ............................................................................................. 699
Opening multiple Keyframe editors ............................................................... 700
Sockpuppet personality editor ................................................................. 773
Sequencing with Sockpuppet ................................................................. 780
Dormant layers ....................................................................................... 780
RGB colour overview ........................................................................... 780
Video playback modes overview ........................................................... 780
Toggle Sockpuppet .................................................................................. 782

**Universal crossfade overview**

What is Universal crossfade? ................................................................. 783
Crossfade duration ................................................................................. 783
Setting up a sequenced transition ........................................................ 787
Universal crossfade overview ............................................................... 789
What is Universal crossfade? ................................................................. 789
Crossfade duration ................................................................................. 789

**Indirections**

Overview ............................................................................................... 795
Workflow .................................................................................................. 796
Examples ................................................................................................. 796

**Generating textures**

Texture types .......................................................................................... 806
Sequencing Overview

This section explains how to sequence the d3 timeline, layers & content mappings.
Sequencing Overview

This section explains how to sequence the d3 timeline, layers & content mappings.
Timeline overview

What is the Timeline?

The terms **Timeline** and **Track Player** both refer to the section at the bottom of the interface. The reason for using two different names to describe the same section is that they suggest different meanings; whilst **Timeline** conveys a sense of time, **Track Player** refers to the function of playing a track. Therefore both names are used synonymously within the User Guide depending on their context.

Objects within the Track Player

The Track Player is a specialised object editor used to play and edit **tracks**. All sequencing and Timeline information is organised into tracks. The Track Player is constructed from a series of properties explained below to make playing and editing tracks more intuitive.

Track Anatomy

- Track name (1)
- Time passed since the section started (2)
- Section name & time remaining within the section (3)
- Track collapsor, makes a mini version of the Track where only Transport controls are visible (4)
- Notes & Cues on the Timeline (5)
- Timeline zoom controls allowing users to toggle between zoom levels on the timeline (6)
- Track scroll bar which allows you to scroll the length of the Track as well as view the current
location of the playhead (7)

Machine status including name, current FPS, battery monitor and local time (8)

Please see the sub-chapter Tracks overview for information on tracks.

Closing/re-opening the Track Player

1. Left-click the x icon in the top left corner of the Track Player to close the Track Player.

2. Right-click track in the dashboard (bar at the top of the screen) to re-open the Track Player.

This is a useful feature, particularly when you want to make screenshots and the Track Player is obscuring part of your view.

Controlling the Timeline from external sources

The disguise software can be configured to control the Timeline with SMTPE, MTC, Art-Net, MIDI notes, OSC and more. Read the Transports chapter for more information.
Transport controls

There are three play modes: **Play**, **Play-to-end-of-section**, and **Loop-section**. When you are using any of these modes, switching to a different mode happens seamlessly, without disturbing the progress of the Playhead.

Move the Playhead

The Playhead selects a position on the Timeline. When entering a play mode, the track will play from the current position of the Playhead.

To move the Playhead:

1. Left-click anywhere on a coloured Track bar to move the Play cursor to that point.
2. Hit the left and right **Arrow** keys to move the Play cursor one Track bar to the left or right.

Play
Normal play can be triggered by pressing the **Enter** key. Hitting **Enter** again will stop the cursor.

Normal play mode does not respect section boundaries; when the end of a section is reached, the playhead continues into the next section, stopping only at the end of the track. The Play button will flash to indicate which mode you are using.

### Play-to-end-of-section

1. **Play-to-end-of-section** mode can be triggered by hitting the **Space bar**. Hitting the **Space bar** again will stop the playhead.

   In this mode, the playhead will only progress until it reaches the end of a section (i.e. just before the next cue point). When this happens, the disguise software enters in a holding state. To indicate this, the button will flash at twice the normal speed and the playhead will remain static, although video and other content will continue to play.

2. When this happens, you can use the Next and Previous Section jump buttons to advance to the next section (i.e. trigger the next cue).

### Loop-section

- **Loop-section** play mode can be triggered by hitting the right hand **Shift** key. Hitting the right-hand **Shift** key again will stop the cursor.

In this mode, the playhead will progress until it reaches the end of a section, i.e. just before the next cue point. At this point, the playhead will loop back immediately to the beginning of the section and will continue playing from there.
Stop

Left-clicking the **Stop** button on screen will stop the playhead regardless of which play mode you are in. If you are using the keyboard, press the shortcut key for the play mode you are in to stop the cursor (i.e. the play keys act as toggles).

Previous / Next section

Left-clicking the **Previous-section** button will make the cursor jump to the previous section if the cursor is on the first bar of the current section; otherwise the playhead will jump to the first bar of the current section.

You can also jump to the previous section using the **comma [ , ]** key on the keyboard.

Left-clicking the **Next-section** button will make the playhead advance to the start of the next section. If the playhead is in holding mode at the end of the section, the disguise software returns to Play-to-end-of-section mode. This button is therefore the equivalent of hitting the next cue.

You can also jump to the next section using the **period [ . ]** key on the keyboard.
Return-to-start-of-track

| [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] |

- Left-clicking Return-to-start-of-track brings the playhead back to the very beginning of the track. Note that the software will stop playing when you click this button.

Next / Previous track

| [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] | [ ] |

- Once you have created a set-list and selected it, left-clicking the Next-track and Previous-track buttons will make you step through the selected set-list.
Tracks overview

What is a track?

All disguise sequencing and Timeline information is organised into tracks. Tracks provide a method for organising content and control commands for the show; the entire show may be placed into one track, or can be divided into multiple tracks. Better performance can be expected from dividing long tracks into individual ones.

The Track editor is used to play and edit each tracks within a project, as well as adjust each tracks individual settings like total length, and universal crossfade duration.

The Overview of the Timeline topic explains how to manage and edit the sequencing of individual tracks. Layers of content are placed on the Timeline in each track, which is explained in depth in the Working with Layers section.

Multiple tracks can be controlled at once through the use of Multitransports, to better organise content and control layers. More information on the use and creation of Multitransports can be found here.

Using tracks to sequence content to the beat

As described above in the Overview of the Timeline sub-chapter, the Track Player is used to play and edit tracks. This chapter explains how to create, manage and edit tracks. However, what is not discussed are layers and audio files. Layers are placed on the Timeline in the currently active track for disguise sequencing. This is fully explained in the Working with Layers chapter. Audio files are inserted into a track, enabling content to be sequenced to the beat. For more information please see the Audio chapter.
The track editor

Right-click the Transport controls title bar. This will open up a Track editor that lets you edit various track properties.

Special Settings

| Song 1 | Special Settings
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Length: 00:09:20.00</td>
</tr>
<tr>
<td></td>
<td>Crossfade Override: Off</td>
</tr>
<tr>
<td></td>
<td>Default Crossfade: 0</td>
</tr>
</tbody>
</table>

Total Length

This property defines the total length of the track in Hours Minutes Seconds & Frames (HMSF).

Please note: Note that when the Timeline is Quantised, the legacy BPM options return to the dialog.
Crossfade Override

Turns on the crossfade function for this track. For further information see the Universal Crossfade section.

Crossfade Duration

When enabled, this controls the crossfade time in beats. For further information see the Universal Crossfade section.

Timecode Chase

If for any reason you need to delay various tracks differently, you can do this by right-clicking the track titlebar to open up the timecode chase tab and editing the MTC adjust property.

For more information on timecode chasing please see the sub-chapter Triggering cues with timecode.
Track-select

Track ID

To select tracks based on an external MIDI note event, type in the *track ID* property in the text field with either a note number or note name (eg. C#1). When the disguise software receives a MIDI note, it will scan all tracks in the current active set-list. If it finds a match, it will trigger the track. For more information on MIDI notes please see the sub-chapter Midi notes.

For step-by-step instructions on how to use the *track ID* property to change tracks with artnet visit Changing tracks with artnet from the sub-chapter Controlling the Timeline with Artnet.

Midi Beat-Clock
Clock Divider

This property is used when synchronising to MIDI beatclock (as opposed to using timecode). It sometimes happens that the clock received is a multiple of the track bpm. For example, the track may originally be 60 bpm, but the input clock is received at 120 bpm. Setting the clock divider property allows you to scale the input clock; in this example, setting the divider to 2 will achieve the desired result.

Count In

The count in property is also used only when synchronising to MIDI beatclock. When the clock starts for the first time, d3 will wait for the specified number of beats before triggering track play.

Scroll down to the section MIDI beat-clock track settings in the MIDI beat-clock topic for more information.

Cue Table

The disguise software allows you to export Track Cue notes to a table, allowing you to edit them en masse in a word processor or spreadsheet.

Export Cue Table

Left-clicking export cue table opens up a tab separated text file in a folder called objects/table.
Import Cue Table

Edit the cue names in the text file. Then go back to the Cue Table and left-click `import cue table` to load and apply the cue names.
Creating and managing tracks

Creating a track

1. Left-click track in the d3 State editor (bar at the top of the screen). This will open up the Track manager.

2. Type in the name of the new track in the new track text field and hit Enter. The new track will be created, added to Track manager and activated on the Timeline.

Switching from one track to another

Select the track you want to switch to from the Track manager. The currently active track in the Timeline will immediately update.

Creating a set-list

A show often uses many tracks. To make this easier to manage the tracks can be organised into Set List. A Set List can be created from either the Cue List or the transport manager.

It is important to use the Set List functionality to effectively manage what data the Cue List is showing.
1. Open the Cue List either through the Transport Manager, or by pressing CTRL G.

2. Left click the Set List tab.

3. Right click the Set List, the Set List manager opens.

4. Create a new Set List in the manager or select an already created Set List.

5. Add the desired tracks to the Set List.
Adding and removing sections

Adding a section

1. Right-click the bar at the point where you want to start a new section and select \texttt{split section} from the popup menu.

2. You can also hit \texttt{Alt+S}, which creates a section break at the current cursor position.
Removing a section

1. Right-click a bar within the section and select the **merge section** option from the popup menu. As a result, the current section will merge with the previous section.
2. You can also hit Alt+M, which merges the section containing the current cursor position with the previous section.
Working with Notes

Notes are items of text added to the timeline to aid the user in identifying parts of the track.

Adding a note to the track

Right-click the bar where you want to add a note, left-click the note text field and type in the text you want to add.
Note field complete. Click OK to add the note to the track.

Alternatively, you can hit **Ctrl+N**, which lets you type in or edit the note for the current cursor position.
Removing a note from the track

Select the already added text in the note data field and press the Backspace key followed by Enter or left click the OK button.

Alternatively, right click on the note and left click Delete in the track options editor.

Remove all Notes

It is also possible to remove all tags from a track or section of a track by selecting the section of timeline where the tags are to be deleted, then right click in the shaded area to open the Track Options window; once open, select Remove Notes.
Moving a note on the track

- Hold Alt & left click and drag the note horizontally along the track.
Cue List

The Cue List shows a list of all sections & tags on the tracks within the currently active set list, allows users to trigger specific tags with a 'go' button, view where they are in the Set List, create new Set Lists and trigger the show from one single interface.

Opening the Cue List

The Cue List can be opened in two ways.

**Option 1**

- Right click Transport in the dashboard
- Expand the Set List tab
Left click the **Open Cue List** button

---

**Option 2**

Use the keyboard shortcut - CTRL G to open the Cue List.

---

**Using the Cue List**

**Auto scroll**

The Cue List will scroll as the user plays through the timeline to allow the user to track where in the Cue List they are, without needing to manually scroll.

**Track jump**

Left clicking on a track name in Cue List will take you to the beginning of that track in timeline.

**Active selection**

The active Section will be highlighted in green, and the most recent cue will be highlighted in a lighter green.
Row selection

- The row selection follows much the same behaviour as list editor. Users can select a track for playback by hovering over cue row and selecting it with a click, users can then click the GO button to trigger that cue, or the note or tag columns to edit.

- Users cannot select more than one row

- Once a track is selected users can select other tracks by pressing up and down.

- User can press CTRL + UP to jump to top of list.

- User can press CTRL + DOWN to jump to bottom of list.

Adjustable size

Users can adjust the width as well as height of widget by hovering over the edge, then left clicking and dragging when the arrow icon is visible. The note column will truncate text (in the middle of the text) if there are large strings in the note column.

Removing tracks from the Cue List

If the Set List is set to Automatic, then all tracks in the project will display in the Cue List. However this is inefficient and can cause performance issues with large projects, therefore we recommend the use of Set Lists.

Removing a track from the Cue List can be done by specifying a Set List that does not contain that particular track.

Set List configuration

A show often uses many tracks. To make this easier to manage the tracks can be organised into Set List. A Set List can be created from either the Cue List or the transport manager.

It is important to use the Set List functionality to effectively manage what data the Cue List is showing.
1. Open the Cue List either through the Transport Manager, or by pressing CTRL G.

2. Left click the **Set List** tab.

3. Right click the Set List, the Set List manager opens.

4. Create a new Set List in the manager or select an already created Set List.

5. Add the desired tracks to the Set List.

---

**Editing items within the Cue List**

- Double click the note field to edit or add a note to the track.

- Double click the tag field to edit the tag after it has been added to the timeline. Once a tag has been added using the traditional tag workflow specified in the tags topic, users can edit that tag in the Cue List. Double clicking the tag column when before a tag is added will do nothing.
Searching the Cue List

As well as the standard text search (looking for matches in track name, notes and tags), you can filter your search by type, with the following syntax:

**track: name** will return only tracks that match the search text.

**note: name** will return only notes that match the search text.

**tag: name** will return only tags that match the search text.

**cue: name** will return notes and tags that match the search text.
Jumping track bars

Jumping units of time on the Timeline using keyboard arrows

You can control how to jump Track bars on the Timeline. This is useful if you want to, for example, find and create a cue point on the track from when the chorus is about to start later in a music video. To do this:

1. Right-click the d3 icon at the top left corner to open up the Program Settings menu.
2. Select project settings and open the timeline control tab
3. Left-click deferred jump. This will point to two options.
4. Left-click yes to activate deferred jump. With this feature activated, you can now hit the left or right Arrow keys (while playing) to jump Track bars without losing sight of the original bar you jumped from (this bar will also continue to flash).
5. Left-clicking no will deactivate this feature.

Jump to the next section using the angle bracket keys

- To jump to the next section rather than the next bar, click the greater than angle bracket (>) key.
- To jump to the previous section click the Less than angle bracket (<) key.

Jumping to cues from a cue-list

After creating sections and adding notes you can open a cue-list from which you can jump to a specific cue.
1. Hit **Ctrl+G** to open the cue-list.

2. Left-click the Go button, next to the name of the cue (the note name) to jump to the selected cue.

<table>
<thead>
<tr>
<th>Song</th>
<th>Note</th>
<th>Tag</th>
<th>Remaining</th>
</tr>
</thead>
<tbody>
<tr>
<td>Song 1</td>
<td>GO</td>
<td></td>
<td>(00:09:20.00)</td>
</tr>
<tr>
<td></td>
<td>Intro 1</td>
<td></td>
<td>(00:00:15.00)</td>
</tr>
<tr>
<td></td>
<td>Verse 2</td>
<td></td>
<td>(00:00:45.00)</td>
</tr>
<tr>
<td></td>
<td>Chorus 3</td>
<td></td>
<td>(00:01:15.00)</td>
</tr>
<tr>
<td></td>
<td>Outro 4</td>
<td></td>
<td>(00:01:45.00)</td>
</tr>
<tr>
<td>Song 2</td>
<td>GO</td>
<td></td>
<td>(00:09:20.00)</td>
</tr>
<tr>
<td></td>
<td>Intro 1</td>
<td></td>
<td>(00:00:15.00)</td>
</tr>
<tr>
<td></td>
<td>Verse 2</td>
<td></td>
<td>(00:00:45.00)</td>
</tr>
<tr>
<td></td>
<td>Chorus 3</td>
<td></td>
<td>(00:01:15.00)</td>
</tr>
<tr>
<td></td>
<td>Outro 4</td>
<td></td>
<td>(00:01:45.00)</td>
</tr>
<tr>
<td>Song 3</td>
<td>GO</td>
<td></td>
<td>(00:09:20.00)</td>
</tr>
<tr>
<td>Song 4</td>
<td>GO</td>
<td></td>
<td>(00:09:20.00)</td>
</tr>
<tr>
<td>Song 5</td>
<td>GO</td>
<td></td>
<td>(00:09:20.00)</td>
</tr>
</tbody>
</table>
Content mapping Overview

Mapping is the process of copying content from the Timeline level to the screens in the Stage level.

All visual layers contain the property mapping, which is used to direct the layer output to the screens on the stage. For information on how to use the mapping property, please see the sub-chapter creating a new mapping.

Types of mappings

Several mapping types in the disguise software: Direct mapping, Feed mapping, Parallel mapping, Perspective mapping, Cylindrical mapping, Radial mapping, Spherical mapping, CameraPlate mapping, Spatial mapping, and PreComps.

These mapping types are able to: sample content across single or multiple screens with pixel perfect accuracy; map content onto 3D objects of any form; accurately project content onto static or moving screens; and project geometrically from virtual cameras in a number of different ways. For more detailed information on the mapping types please read the following topic Mapping types overview.
All visual layers contain the property `mapping`, which is used to redirect the layer output to the screen(s) in the stage. For information on how to use the `mapping` property please see the topic Creating a new mapping.
Content mapping Overview

Mapping is the process of copying content from the Timeline level to the screens in the Stage level.

Conceptual diagram illustrating how mapping is used to control how content is copied from the Timeline to the Stage level.

All visual layers contain the property mapping, which is used to direct the layer output to the screens on the stage. For information on how to use the mapping property, please see the sub-chapter creating a new mapping.

Types of mappings

Several mapping types in the disguise software: Direct mapping, Feed mapping, Parallel mapping, Perspective mapping, Cylindrical mapping, Radial mapping, Spherical mapping, CameraPlate mapping, Spatial mapping, and PreComps.

These mapping types are able to: sample content across single or multiple screens with pixel perfect accuracy; map content onto 3D objects of any form; accurately project content onto static or moving screens; and project geometrically from virtual cameras in a number of different ways. For more detailed information on the mapping types please read the following topic Mapping types overview.
All visual layers contain the property `mapping`, which is used to redirect the layer output to the screen(s) in the stage. For information on how to use the `mapping` property please see the topic [Creating a new mapping](#).
Mapping types overview

Direct

Direct mapping is the easiest form of mapping. Simply select a piece of content and apply it directly onto some or all of your screens. If the content aspect is different from the screen aspect ratio, you can choose to crop, fit, stretch or apply the content pixel-perfect onto the screens.

Feed mapping

Feed mapping lets you specify an arbitrary number of rectangles within your content frame and map them to arbitrary rectangles on your screens. With Feed mapping, all your screens can easily be turned into one canvas, with pixel-perfect content applied. Individually move, scale, crop, chop or flip your sample rectangles to compensate for differences in pixel-density or to create interesting mapping effects.
Parallel

Parallel mapping allows you to position your content in a virtual emitter rectangle anywhere within the stage and to choose which screens and fixtures this affects. The disguise software then virtually projects the content out onto those fixtures. The content keeps the same size as it gets further away from the emitter.

Perspective

Perspective mapping is similar to Parallel mapping, except that the content originates from an emitter point and gets larger as you get further away from the emitter, similar to a real projector. You can use perspective mapping to map 3D content onto a surface that looks perfect from the specified point of view.

Cylindrical mapping

Cylindrical mapping is similar to Parallel mapping, except that the Cylindrical mapping wraps content around a cylinder and fires it outwards (or inwards) at the screens you assign to it. It does not matter if the screen is moving or expanding, the Cylindrical mapping feature re-maps the content in real-time.

Radial

Radial mapping is similar to the Cylindrical mapping, except that the emitter surface runs from the central axis to the cylinder rim, rather than being wrapped around the outside of the cylinder. The pixels inside the cylinder will therefore be mapped according to their height within the cylinder and their distance from the central axis.

Spherical

Spherical Mapping works similarly to the Cylindrical mapping, except the top and bottom are 'pinched' to create a sphere.

CameraPlate

A CameraPlate mapping is a 3D Camera based mapping that is fixed to the plate of the camera. Using a CameraPlate mapping is necessary when doing color adjusts on the set extension.

Spatial

A Spatial mapping is a 3D Camera based mapping that maps into 3D space; it has a “space” or location so it can be moved around 3D environments.
MultiChannelMap

The default mapping for the RenderStream layer, the MultiChannelMap allows you to assign multiple mappings to multiple unique RenderStream channels so you can manage all within the same mapping.
Creating a new mapping

Process used to create a new mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

Note: In the Mapping keyframe editor that will appear across the screen, move the playhead and assign multiple mappings sequentially for each layer by left clicking in the editor.

Remove mappings from your project by left clicking on the name of the mapping and dragging to the trash can icon within this menu. All mappings that are assigned to a screen or camera within the project but have been moved to the trash will appear highlighted in red.

**Please note:** it is useful to organise the different mappings in boxes, particularly if there are a very large number of mappings. For information on how to create, edit and organise objects
within boxes, please see the section 'Organising objects in boxes' in the Object libraries sub-chapter.
Common mapping properties

This section explains the properties that are shared by the six different mapping types.

**Filtering**

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

**Resolution**

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

**Screens**

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

**Mask**

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the *DxTexture* folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.
Direct mapping

Direct mapping is the easiest mapping type. It specifies a canvas size (in pixels) and a list of screens where the canvas content will be copied onto.

Direct mapping is the default mapping applied to all screens and is a 1:1 ratio of content to screen. The content will stretch and shrink to the size and resolution of the assigned screen. You can choose to crop, fit, stretch or apply the content pixel-perfect onto the one or multiple screens, all within one direct mapping.

When you create a new screen, the disguise software automatically creates a new Direct mapping with the same name as the screen and only includes that screen.

The resolution of the Direct mapping type is automatically set to the highest resolution screen of the screens included in the Direct mapping.
Creating a Direct mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

Common Mapping Properties

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neigbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the ```DxTexture``` folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.
Feed Mapping

The Feed mapping type allows you to sub-sample rectangular areas in a content area (referred to as the Mapping canvas) and copy that content onto rectangles inside one or multiple screens.

Feed mapping lets you specify any number of source rectangles within your content and map them individually to multiple surfaces. With Feed mapping, all your screens can easily be turned into one overall raster, and content applied to each surface with pixel-perfect accuracy. Individually move, scale, crop, chop or flip your source rectangles to compensate for differences in pixel density or to create interesting mapping effects.
Creating a Feed Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

Feed Mapping Properties

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.

Using the Feed mapping type

In addition to the property explanations provided in the sub-chapter Common mapping type properties, it is important to:

1. Set the resolution of the Feed mapping to the same resolution as the content intended to be used for that Feed mapping. If the content is of a higher resolution than the resolution of the Feed mapping mapping canvas, the disguise software will by default crop away the sides of the content. If the content is of lower resolution, the disguise software will scale up the content inside the mapping canvas.

2. Add the screens to the Feed mapping in the same order as they appear in the visualiser, from left to right. If the screens are not listed in that order yet, you can re-order the screens in the screens list by selecting and dragging a screen up or down.

Adding a Feed rectangle to the mapping canvas

To add a Feed rectangle to the mapping canvas:
1. Right-click the red screen canvas and select **Add feed rectangle**.

![View of the Feed mapping editor when adding a new Feed rectangle](image)

Note that the Feed rectangle is by default positioned in the top left corner of the mapping canvas. To change the position of the Feed rectangle, left-click it once, place the cursor in the middle and start moving it.

**Editing a Feed rectangle**

The Feed rectangle has a number of options for scaling, flipping and chopping the rectangle. These options are very similar to those used when editing a Feed rectangle in the Output Feeds level. For more information please see the sub-chapter **Quick editing controls**.

This image explains the various options associated with a Feed rectangle in the Stage level.
<table>
<thead>
<tr>
<th>Item</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="585_271.png" alt="Image" /></td>
<td>The position of the sample rectangle placed on the Mapping Canvas, in pixels</td>
</tr>
<tr>
<td><img src="174_674.png" alt="Image" /></td>
<td>The <strong>Scissor tool</strong> on the sample rectangle for vertical cuts. Left-click and hold down the mouse button to move the cut horizontally.</td>
</tr>
<tr>
<td><img src="174_623.png" alt="Image" /></td>
<td>The <strong>Scissor tool</strong> on the sample rectangle for horizontal cuts. Left-click and hold down the mouse button to move the cut vertically.</td>
</tr>
<tr>
<td><img src="174_571.png" alt="Image" /></td>
<td>The <strong>Bin</strong>. Left-click the bin icon once to delete the sample rectangle (and the Feed rectangle).</td>
</tr>
<tr>
<td><img src="172_339.png" alt="Image" /></td>
<td>The size of the sample rectangle in pixels.</td>
</tr>
<tr>
<td><img src="174_305.png" alt="Image" /></td>
<td>The <strong>Flipping tool</strong> to flip the sample rectangle horizontally.</td>
</tr>
<tr>
<td><img src="175_270.png" alt="Image" /></td>
<td>The <strong>Scissor tool</strong> on the Feed rectangle for vertical cuts. Left-click and hold down the mouse button to move the cut horizontally.</td>
</tr>
<tr>
<td><img src="175_235.png" alt="Image" /></td>
<td>The <strong>Rotate tool</strong> on the Feed rectangle for rotating the Feed rectangle 90 degree clock/anti-clockwise.</td>
</tr>
<tr>
<td><img src="175_190.png" alt="Image" /></td>
<td>The <strong>Scissor tool</strong> on the Feed rectangle for horizontal cuts. Left-click and hold down the mouse button to move the cut vertically.</td>
</tr>
<tr>
<td><img src="175_144.png" alt="Image" /></td>
<td>The <strong>Flipping tool</strong> to flip the sample rectangle vertically.</td>
</tr>
<tr>
<td><img src="175_99.png" alt="Image" /></td>
<td>The <strong>Bin</strong>. Left-click the bin icon once to delete the Feed rectangle (and the sample rectangle).</td>
</tr>
<tr>
<td><img src="175_54.png" alt="Image" /></td>
<td>The <strong>Lock button</strong> locks the size of the sample rectangle to the size of the Feed rectangle and vice versa. Left-click the Lock button once to unlock the dependency.</td>
</tr>
<tr>
<td>Item</td>
<td>Explanation</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>surface 1</td>
<td>The name of the screen.</td>
</tr>
<tr>
<td></td>
<td>The size of the Feed rectangle in pixels.</td>
</tr>
</tbody>
</table>

Using these tools gives you a lot of freedom for configuring the Feed mapping on the screens in the Stage. Below, you can see an extreme example of this (for demonstration purposes only). Note that the red color in the background on the left hand side will only show up if the Feed mapping editor is open. When closed, the background will remain black.

Feed rectangle properties

Right clicking a feed rectangle offers the following properties.

Add feed rectangle

This option adds another feed rectangle to the canvas.

Export current frame as PNG

A PNG of the feed rectangle is exported to the screenshots folder of your project folder. For more information see project structure.

Edit feed warping

This option allows you to draw a 2D warp on an individual feed rectangle. Useful for warping within the mapping layer as oppose to the feed level.

Select mask

This option allows you to add a mask to the feed rectangle. Masks are read from the DxTexture folder of your Objects folder within your project folder. For more information see project structure.

Duplicate

This allows you to create a duplicate of the feed rectangle.
Duplicate multiple

This allows you to duplicated multiple source and destination feed rectangles to create arrays of feed rectangles within the mapping.

Turn snap off

This option toggles feed rectangle snapping on and off. When on, feed rectangle edges will snap to one another, and when off they do not do this (note the rulers still show with snapping off).

Import & Export feed maps

The easiest way to import and export feed maps from projects is via a table. A feed map can be exported to a text file from the disguise software and then imported to another project.

Export

1. In your feedmap, right click and choose Export to table.
2. Enter a name
3. Hit OK
4. The feed map is now exported to a newly created table folder in the objects folder. The table that is exported will look similar to this example:

"Screen Name" refers to the name of the feed rectangle while "Head" refers to the number of the output slot.

**Import**

1. In your new project, create a feed mapping. The mapping does not need to share the same name as the one you are importing.

2. Left click the Import from table option, this will only appear if there is a table folder present in the project folder.
3. Left click your table from the available list.

4. Choose yes when prompted to overwrite existing feed rectangles.
Parallel mapping

The Parallel mapping type projects content geometrically into the scene, as if ‘virtually shooting content from an emitting rectangle. The mapped image does not increase in size the further away you go from the emitting rectangle. Instead, the image remains the same size, hence the term parallel.

Parallel mapping will emit content over multiple screens and treat the configuration as a single canvas, projecting one unified image over all assigned surfaces. There is no more need to calculate the exact distance and pixel density of the empty space between screens, the disguise software will maintain the correct content sampling even across moving surfaces.

Parallel mapping is also very useful when you want to apply content onto moving screens. As long as the disguise software receives the correct tracking signal, the system will automatically map the content onto the moving screens. See more about screen tracking in the sub-chapter Motion control systems.

Furthermore, Parallel mapping is a great tool for combining not only LED technology of various pixel-pitches, but also to combine LED, projection and DMX lighting into one canvas. As disguise considers
all screens to be made up of pixels, regardless if they are LED, projection or DMX-based screens (including moving heads), it is very easy to combine these different type of screens into one canvas.

**Please note:** the Parallel mapping may not be useful when the screens are facing awkward angles in relation to the emitting rectangle of the Parallel mapping. The more they rotate away from the emitting rectangle, the more stretched the pixels will be on the screens. Ideally, the Parallel mapping should be used for straight-facing rectangular screens. For low-res sculptural screens, this is however not a problem.
Creating a Parallel Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

When the Parallel mapping editor initially opens, and assuming that the screens have been added, the disguise software will colour the region outside the projected image red and will give you the location and orientation details of the image source.
Common Mapping Properties

This section explains the properties that are shared by the six different mapping types.

Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which
shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.

The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to **mask** to channel the layer content into the mapping mask.

---

**Parallel Mapping Properties**

**Lock**

For convenience while editing, the mapping can either maintain its aspect ratio, which changes vertical size or resolution when you change horizontal size or resolution, and vice versa. The pixel density, which changes the vertical or horizontal size when you change the vertical or horizontal resolution can also be locked.

*Please note:* if the aspect or the density of the source content is not locked the Parallel mapping may project stretched pixels onto the screens

**Anchor**

This controls the anchor point around which the mapping surface moves, scales and rotates. Select either **centre** to specify the position of the centre of the projection image or select **corner** to specify the bottom left corner of the image.

**Size**

This controls the size of the mapped image, in meters (horizontal and vertical).
**Position**

The position, in stage space (i.e. in meters) of the anchor point of the mapping source. This can be either the centre of the image or the bottom left corner of the image.

**Heading**

This controls the direction of the mapping in the horizontal space, in degrees: 0 degrees points north (i.e. in the positive z axis direction). Increasing the angles will make the direction to rotate clockwise, so 90 degrees is west (i.e. the negative x direction), 180 degrees is south (negative z direction) and 270 degrees is east (positive x direction).

**Elevation**

This controls the vertical elevation, in degrees, of the projection direction: 0 degrees is horizontal (i.e. parallel to the floor), 90 degrees is straight up and 90 degrees is straight down.

**Rotation**

This controls the rotation of the image around the mapping axis, clockwise in degrees.

**Using the Parallel mapping type**

In addition to the property explanations provided in the sub-chapter Common mapping type properties, it is important to:

1. Set the aspect property to off and set the resolution to match the resolution of the content intended to be used for this specific Parallel mapping.

2. Set the size of the emitting rectangle of the Parallel mapping to match the aspect of the resolution (and thereby the aspect of the content) and then set the aspect to locked. For example, if your content is 1920x1080, set the size to 19.2 10.8 and then set the aspect to lock. If the aspect or the density of the source content is not locked the Parallel mapping may project stretched pixels onto the screens.

3. Re-size the x or y coordinates to cover all areas of the screens (the red areas will gradually disappear as you re-size the emitting rectangle).
4. Make sure to specify content which has the same, or a slightly higher pixel density than the screen with the highest pixel density.
CameraPlate mapping

The CameraPlate mapping always targets a specific camera.

The CameraPlate mapping was added for the Augmented Reality (AR) workflow, where virtual objects can be mapped within the scene. The CameraPlate mapping will essentially “stick” the content to the perceived lens of the camera and adjust the content to always appear in proportion despite the position or movement of the camera.

Creating a CameraPlate Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

Common Mapping Properties

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.
Cylindrical mapping

The Cylindrical mapping type wraps content around a cylinder and fires it outwards (or inwards) onto the selected screens. Similar to the Parallel mapping and the Perspective mapping types, the content is being virtually projected from the surfaces of the cylinder, both inwards and outwards.

Cylindrical mapping wraps content around a cylinder and emits it inward or outward towards the screens assigned to it. It does not matter if the screen is moving or expanding, the Cylindrical mapping re-maps the content in real-time to accommodate for the movement, similar to parallel mapping.

Creating a Cylindrical mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

**Common Mapping Properties**

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.

Cylindrical mapping properties

Output
By default, this is set to colour. This implies that red, green and blue content channels are sent to the red/green/blue colour output channels of the screen (or converted to cyan, magenta and yellow if the output fixtures are CMY). However, content can also be channelled to other channels in the case of DMX-based fixtures. Currently, this is a beta functionality, so for the moment you are advised to leave this set to colour.

Anchor
As with the Parallel mapping type, you can anchor the cylinder around the centre, or the top or bottom points. Resize and rotation operations will keep the anchor point in the same place.

Centre
This controls the stage position of the centre of the cylinder (in other words, its anchor point).

Size
This controls the size of the cylinder in meters, on the x, y and z axes respectively. The y property (the second value) controls the height; the two other properties control the radii in the x and z axes. Setting x and z to the same value keeps the cylinders crosssection circular; setting them to different values makes it elliptical.
Rotation

This rotates the cylinder around the anchor point; specified in degrees of rotation around the x, y and z axes respectively.

Using the cylindrical mapping type

In addition to the property explanations provided in the sub-chapter Common mapping type properties, it is important to:

- Base the resolution of the Cylindrical mapping on the aspect between the circumference of the cylinder and the height of the screen. The circumference can be calculated by taking the \( \text{diameter} \times \pi = \text{size} \times 3.14 \). Set the size x and z to the same value to establish as perfect circular shape.
Perspective mapping

Perspective mapping is similar to Parallel mapping, except that the content originates at an emitter point and gets larger in size as you go further away from the emitter, similar to a real projector. You can use Perspective mapping to make 2D content appear as 3D (not stereoscopic but rather as a 3D effect), but only from a specific point of view which is the same point as the position of the emitter.

Perspective mapping can map 3D content onto a surface from a specified vantage point of a virtual camera. This camera can be linked to a virtual camera in a generative software such as Notch to create an immersive 3D environment. The content being displayed by the mapping will begin emitting at the vantage point and get larger as that surface moves farther from it, similar to a real projector.

Perspective mapping type being used to project content onto three Screens, notice the content becomes larger as the Screens move further away from the emitter point.
Creating a Perspective Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

**Common Mapping Properties**

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.
2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.
3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.

Perspective Mapping Properties

**Eye point**
The Eye point specifies the source point (emitting point) of the Perspective mapping.

**Rotation**
The Rotation parameter specifies the rotation of the frustum of the Perspective mapping (i.e. the aiming point of the Perspective mapping).

**Field of view**
Defines the field of view of the Perspective mapping, in degrees (i.e. the size of the mapping coverage).
PreComp

A PreComp (or pre composition) in the disguise software acts as a virtual screen for you to write content to which can then be used as an input to another effect or composition. It also provides an alternative to the arrowing system which was limited to a one to one input, whereas a PreComp can be one to many.

A PreComp (or pre composited texture) acts as a container for multiple textures, which can then be used as an input source to another effect or composition. It also expands the capabilities of the arrow/piping system, which was limited to a one input per source. A PreComp can be many inputs to many sources.

The PreComp is applied as a mapping to all the texture/content inputs of the composition.

Creating a PreComp

A PreComp is created just like other textures in the disguise software, from the location where you will be using it. You cannot create a PreComp in some texture fields (such as a palette).

A PreComp is created like other textures in the disguise software such as masks, from the Textures menu, which can be accessed from any layer that utilizes a texture input. Like any other texture, once it is created, it will be stored in the Textures menu and can be used in other locations.

1. Create a new layer and select a layer type that has a Texture source property, such as an Effect layer or Legacy Bitmap layer.
2. In the new texture field, enter a name for your PreComp. Choose PreComp Texture as the texture type.
3. Set the desired resolution for the PreComp (just as you would do if this was a real, physical screen. This should correspond to the resolution of the screen it will be mapped to.
Building the PreComp

Once you have created a PreComp, a new default Direct mapping will now be present in the Mappings Manager, allowing you to route content into the PreComp from a content layer. Content layers include other texture layers as listed above, Video layers, and Generative layers.

1. Create a content layer and left click on the content layer in the timeline to open the layer's properties.

2. Set the layer to display the desired content.

3. Under the Mapping properties field, select the mapping with the same name as the newly created PreComp.

4. Repeat these steps with as many content layers as will be used in the PreComp.
Please note: A PreComp cannot be chained, i.e. a PreComp cannot feed into another PreComp.

Using the PreComp as a source for an Effect Layer

The PreComp can be used as the Source texture input to an effect layer.

1. Create an Effect Layer and position it above the source PreComp layers.
2. Set the Effects layer mapping to the screens the content will be shown on (do not choose the PreComp mapping).
3. From the Source Input field under the Default tab, select the PreComp.
4. All of the individual content layers will now be piped into the Effect layer simultaneously.
5. Hold down shift and left click to select all of the source layers used in the PreComp.
6. Right click on the layers and name a group for the source layers.
7. Note that the source layers must be underneath the Effect layer for the PreComp to show up.

Please note: The source layers that make up the PreComp have to be positioned under the Layer that is referencing the PreComp texture. To reuse the same PreComp in multiple places, the source layers must be duplicated.
Using the PreComp in multiple tracks

In order to use the PreComp in multiple locations, all of the layers that are contained in the PreComp stack need to be copied as well. One way to do this is to create a Group for the PreComp and all of its layers.

1. Select all of the layers that are contained in the PreComp stack; once all have been selected, then right click on the group while they are selected. Type in a name for the group of layers in the Group field and click OK.

2. Next, right click on the name of the newly created PreComp group in the timeline; click OK next to the **Save As** field; this will add the group of layers to the Layer Library.
3. Switch to the desired track.

4. Right click in the timeline for the track.

5. Select Import Layer.
6. Select the name of the group of layers you wish to import from the Layer Library.

Now you can use the same PreComp many times throughout the project.
Previewing a PreComp

1. Right click on the PreComp thumbnail in the Layer properties window.

2. Use middle mouse button to pan around the expanded preview.
Radial mapping

The Radial mapping is similar to the Cylindrical mapping type, except that the content surface runs from the central axis to the cylinder rim, rather than being wrapped around the outside of the cylinder. Any fixtures or pixels inside the cylinder will therefore be mapped according to their height within the cylinder (y coordinate) and their distance from the cylinders axis (x coordinate). Fixtures outside of the cylinder are not affected.

Radial mapping is similar to the Cylindrical mapping, except that the emitter surface runs from the central axis to the cylinder rim, rather than being wrapped around the outside of the cylinder. The pixels inside the cylinder will therefore be mapped according to their height within the cylinder and their distance from the central axis.
Creating a Radial Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

Common Mapping Properties

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.

Radial Mapping Properties

**Anchor**
Base point from which the scale of the Radial mapping has its origin.

**Centre**
Position of the Radial mapping based on the centre point of the cylinder.

**Size**
The size of the Radial mapping. The x and z values represent the diameter of the cylinder.

**Rotation**
The x, y, z rotation of the cylinder.
Spherical Mapping

The Spherical Mapping works similarly to the Cylindrical mapping, except the top and bottom are 'pinched' to create a sphere.

Spherical Mapping works similarly to the Cylindrical mapping, except the top and bottom are 'pinched' to create a sphere.
Creating a Spherical Mapping

1. In your track, create a new visual layer. These layer types can include content layers, generative layers, and effect layers.

2. Left click on the layer in the track to open the Layer Editor on the left side of the GUI.

3. Under the Default tab, left click on the Mapping parameter to display a list of the mappings in the project.

4. A manager titled mappings will appear – this is a list of all mappings within your project. By default, all screens will be populated with a direct mapping sharing the same name, and all cameras will be populated with a perspective mapping sharing the same name.

5. Type directly into the New mapping: text field to create a new mapping. A list will appear prompting to you select the mapping type.
6. An editor with the user assigned name of the mapping will appear once a mapping type is selected.

7. Assign all screens that will be used for the mapping and enter the resolution of the content that will be displayed on the screens.

8. Edit the specific properties of the chosen mapping type.

9. Assign a piece of content to the layer in the track, and the content will be displayed on the assigned screens of the mapping in the stage and the feed output window.

**Common Mapping Properties**

This section explains the properties that are shared by the six different mapping types.
Filtering

**Nearest** - Nearest neighbor filtering. Use nearest-neighbour sampling, to disable blending between pixels when scaling. Can be used to create pixellated looks, or to ensure hard edges on certain types of content.

**Bilinear** - Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**2x Multi-sample** - Multi-Sample filtering can help fix issues with scaled content, but can introduce some blurriness.

Resolution

This controls the canvas size the layer renders into, in pixels. The Direct mapping type starts with a 256x256 pixel canvas and automatically sets the canvas size to that of the first screen you add.

Screens

This is a list of screens that the selected mapping type can copy content to.

1. Left-click + to open the Screens manager.

2. Left-click the Screens you want to map. This will copy the individual canvas content onto these three Screens simultaneously and will add the Screen names to the mapping object editor.

3. Left-click and drag the Screens listed in the mapping object editor to -. This will remove the canvas content from the Screens and delete the Screen names from the mapping object editor.

Mask

This points to the Texture file that defines a Mask bitmap. You can use this property to apply a Mask bitmap to the mapping canvas. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To apply a mapping mask you will need to create and import a custom still image file.
The step-by-step instructions on how to create and import a custom Population mask can be used to create a custom mapping mask. The only difference is that the mapping masks resolution should be the same as the mapping canvas. For step-by-step instructions on how to create and import a Population mask into a d3 project visit to the section Population mask in the Editing screens sub-chapter.

Alternatively, set any layers blend mode to mask to channel the layer content into the mapping mask.

Spherical Mapping Properties

**Horizontal Angle**
Defined the horizontal angle of the mapping to map what the content was created to.

**Vertical Angle**
Defined the vertical angle of the mapping to map what the content was created to.
Layer Types Overview

For a broader overview of layers please read the layers overview topic.

Types of layers

There are five main layer types in the disguise software: content layers, generative layers, pre-visualisation layers, effects layers and control layers.

Content layers

The function of content layers is to play back pre-rendered content, map them onto screens and, if needed, apply a number of effects built into the layer.

Audio
RenderStream Layer
TestPattern
Video
VideoTrigger
VirtualLineup
Web

Generative layers

Generative layers are packets of algorithms inside the disguise software that produce visuals based on a number of animatable properties. They share some of the same properties as all the other layer types.
such as the Mapping property.

Pre-visualisation layers

Pre-visualisation layers are usually used during the design and pre-production phases of the project when accurate visualisation is required, for example animating screens or creating camera fly-throughs.
Effects layers

Input a bitmap, a video or an output from another content layer into the disguise software effects layers. For example spin, pixelate, blur images, video, or other inputs to create a new look.

Blur
ChannelRouter
ColourAdjust
Compose
EdgeFilter
Fade
FilmicGrain
Kaleidoscope
Lut
MotionBlur
Noise
PixelMap
Pixelate
Scroll
SpinBitmap
Trigger
VideoTransition

Control layers

Control layers are technical layers that control other equipment in the show; for example DVI matrices, switchers or unusual DMX controlled equipment.
Legacy layers

Legacy layers are kept in the software for compatibility purposes but have been replaced with more up to date layers.

Bitmap
Legacy video
Working with layers

It is highly recommended that the previous chapter Working with layers is read before reading this chapter. This is because creating/structuring/editing layers involves many processes, for example placing key-frames on the Key-frame editor, which are all explained in the previous chapter.
Layers overview

A layer is an object which renders content in response to certain inputs.

Layers can generate colours or gradients, play video or display still bitmap images. Content is generated by creating layers, placing them on the Timeline, and animating their properties using disguise’s keyframe editor within the layer editor.

The disguise software uses mappings to control how layer content is copied from the Timeline to the screens in the Stage level. Please see the chapter Content Mapping for a full explanation of this powerful feature.

Types of layers

There are five main layer types in disguise: content layers, generative layers, pre-visualisation layers, effects layers and control layers. For more detailed information on layer types please read the sub-chapter Overview of layer types.

Controlling layer properties with Artnet

Disguise can be configured to control layer properties with Artnet. For step-by-step instructions on how to do this please read the sub-chapter Controlling layer properties with Artnet. You may also be interested in the Sockpuppet workflow.

Controlling layer properties with OSC

Disguise can be configured to control layer properties with OSC. For step-by-step instructions on how to do this please read the sub-chapter Controlling layer properties with Artnet.
Creating layers

A layer can be created using two methods.

Please note: You can create a layer without a name, but it is recommended that you name your layers so that the Timeline is easier to understand.

Option 1

1. Right-click the Track bar at the point on the Timeline where you want to create the layer. This will open the Track bars menu.

2. Type the name of the layer in the new layer text field and hit Enter.
A good format for naming your layers is **Layer-type description**, for example **Bitmap-layer front screen projection**. This will open a list of all of the layers available in the disguise software.

3. The layer menu is separated into different layer categories (content, control, effects, generative and previsualisation) with each category grouped under a collapsible separator.

4. Typing into the search box opens up separators automatically to show you matching layer types.
5. Left-click the layer you want to create. This will create the new layer at the point on the Timeline you selected. The layers editor should also open in the top-left corner of your screen. You can now edit the layer or restructure the layer.

Option 2

1. Select a Track bar at the point on the Timeline where you want to create a layer.

2. Hit Ctrl+L. A list of all of the layers available in the disguise software will open.

3. Left-click the layer you want to create. This will create the new layer at the point on the Timeline you selected. You can now edit the layer or restructure the layer.
Editing layers

Editing layers involves using the Layer editor and the Keyframe editor

What is the Layer editor?

The Layer editor contains a series of properties which are specifically set for each individual layer. All of these properties can be edited.

Opening / closing the Layer editor

To open the Layer editor:

  — Left-click the layer from the Timeline.

For information on how to create a layer please see the sub-chapter [Creating layers](#).

If the currently selected Track bar is underneath the extent of the layer (not including the extender tabs on either side), the Layer editor will open at the top left of your screen. If the currently selected Track bar is outside the extent of the layer, the disguise software will close the editor. This is to ensure that you only see editors for layers that are active.

The layer editor looks like a standard object editor, except that it has a small preview window at the top of the editor showing you what the layer output looks like.

To close the Layer editor:

  — Left-click the layer in the Timeline, or left-click the close button (x icon) in the top-left corner of the Layer editor.

  Please note: hit **Ctrl** whilst left-clicking a layer in the Timeline to view multiple Layer editors simultaneously.

Layer properties

Each layer in the disguise software has a specific set of properties. Please read the chapter [Layer types](#) to understand each layer types individual properties. In addition, please read the next sub-chapter [Common layer properties](#) which explains the properties that are shared by most layers that produce visual output.
MultiLayer Editor

In r17.3 multiple layers can be edited simultaneously making it easier to assign common values or objects across a selection of layers in one action.

Using the MultiLayerEditor

1. Hold **Shift**, left click and drag a selection over multiple layers

2. Left click your selection and a MultiLayerEditor will open that contains all common values to the layers selected

Tips for using the MultiLayerEditor

- Shift + Click to add an un-selected layer to the selection, or remove a layer from the selection.
- A MultiLayerEditor is created for a layer selection, when more than one layer is selected
- A MultiLayerEditor is also created when a single GroupLayer is selected.
- If a selection contains only GroupLayers, a MultiLayerEditor is not created.
- If a selection contains Layers and GroupLayers, a MultiLayerEditor is created for the Layers only.
- Opening a MultiLayerEditor will not close existing Layer editors for the selected layers.
- Adding a layer to the selection or removing one from the selection will create a new MultiLayerEditor with the new selection.
A MultiLayerEditor can only be opened for layers when the track cursor is within all of the selected layers.

The MultiLayerEditor will show all the common fields of all selected layers. The fields will be under the same tabs.

The fields in the MultiLayerEditor show an aggregation of all the values of the selected layers.

The tooltip of the ValueBox will contain the current values and associated layer names.

Dropping a new keyframe will create a new keyframe in all selected layers.

Modifying an existing keyframe will modify the keyframe in the layers where the keyframe exists.

Deleting a keyframe will delete the keyframe in the layers the keyframe exists.

Popup menu actions in the keywidget, like changing interpolation type apply to all selected layers.

Keyframe navigation is the same as the Layer editor

Limitations

Keyframes for each layer are all visible in the same keyframe editor, which can make it hard to know which keyframe belongs to which layer.

There is no preview in the MultiLayerEditor

Keyframes for each layer are all visible in the same keyframe editor, which can make it hard to know which keyframe belongs to which layer.

Some fields may have the same display name but different internal names and will therefore not appear. Some fields may also have the same name but different values, these should not appear. An example is: multi-selecting audio and video layers misses volume control and at-endpoint control.

The MultiLayerEditor cannot aggregate fields that are dynamically created by a custom layer editor. Therefore OpenModule is not compatible and neither are dynamic fields of a notch...
Some fields may have the same name, but represent different things and therefore appear when they shouldn’t. An example is: TimecodeReadout and Gradient Position X and Position Y parameters are incompatible with one another.
Common layer properties

Most of the disguise layers share the same properties.

These are:

- Preview Thumbnail
- Blendmode
- Brightness
- Mapping
- Palette
- xCol, yCol
- Content
- Colourshift
This property is a small preview window at the top of the editor showing you what the layer output looks like.
Brightness

This property (which appears as a light bulb icon) controls the brightness of the layer output.

If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.
Content

This points to the still image/video file (depending on the layer type) that defines the Bitmap/Video layers content. Selecting this property will open the Texture/Videoclip object library, which shows all of the still image/video files saved on your local hard-drive in the DxTexture/VideoFile folder.

To change the content inputted into the layer:

1. Left-click bitmap / video (depending on the layer type) to open the Texture/Videoclip object library.
2. Left-click the still image/video file you want to input into the layer.

**Please note:** the disguise software does not provide video clips by default. If you want to use video or content other than the standard still images provided in the disguise software, you will need to import a custom still image/video file.

See the Placing media files for a project sub-chapter to understand where to place a custom still image file/video file and how to access it in the disguise software. Also save the file to a supported file format.
Content properties

Content properties are sometimes known as Video clip properties and can be accessed by right clicking a content file thumbnail from a layer.

Content properties are unique to a content file. So, by changing the content properties on a piece of content, these properties will apply whenever that clip plays on any layer. The exception to this is if you've duplicated the clip instance within the VideoClip library.

Accessing content properties

Content properties can be accessed by right clicking an asset from a content layer such as Video, LegacyVideo, Bitmap, Audio and Web.
General properties

All properties displayed at the top of the editor are read only.

**Number of frames**
Specifies the number of frames within the file.

**Resolution**
Specifies the file resolution.

**Codec**
Specifies the Codec for the file.

**Bits per Channel**
Specifies the bits per channel of the file.

**Has Alpha**
Specifies whether the file has an Alpha channel or not.

**Has Audio**
Specifies whether the file has embedded Audio or not.

**Original FPS**
Specifies the original Frames Per Second of the file.

**Data rate**
Specifies the data rate in megabytes per second of the file.

**Edit**
The Edit properties can all be altered to suit user requirements.
**FPS**
Sets the Frames Per Second for the content. For example, doubling the FPS will cause the disguise software to display the same frame twice in a row, and half the FPS would cause the disguise software to display every other frame.

**Start**
Clip start: Specifies the start frame.

**End**
Clip end: Specifies the end frame.

**In**
Loop in: Specifies the in frame.

**Out**
Loop out: Specifies the out frame.

**Filtering**

**Bilinear filtering**
Specifies whether Bilinear filtering is on or off for that content.

Bilinear filtering is a texture filtering method used to smooth textures when displayed larger or smaller than they actually are.

**Frame blending**
Specifies whether frame blending is on, off or automatic for the content.

**Deinterlace method**
This option is only available for live video inputs and allows you to specify a deinterlacing method. For more information, see [Interlaced video capture](#).
Quantisation

Quantised
When set to No, the content will play at its defined frame rate. When set to Yes, the content will play at the specified number of beats.

Beats
The number of beats the content should play back to when the Quantised property is set to Yes.

Cropping

Clip type
Specifies the clip type the content should use. This property used to be held in the direct mapping in previous versions of the disguise software.

Clip to canvas: Clips the content to the canvas.
Fit into canvas: Fits the content to the canvas, but can result in 'black bars' of empty space along one axis of the content.
Stretch to canvas: Stretches the content so it fits the whole canvas, while ignoring aspect ratio.
Pixel-perfect: Displays the content, pixel perfect and does not stretch or scale in any way.

Zoom point

Crop left/right/top/bottom
Crop can be used to crop the canvas manually. Entering minus values creates additional black space around the content.
Colour shift

Defines the colour shift for the clip. Colour shifts allow you to specify brightness (shift), contrast (scale), saturation scale, Hue shift, RGB control as well as Keying threshold, Hardness and Key colour. See Colour shifts for more information.

Colour profile

This property allows you specify a colour profile for the content. Colour profiles consist of a colour space and gamma curve specification. See Colour profiles for more information.

Versions

Proxies

This properties shows the available proxy files for the content. For example Full, 1/2, 1/4 and 1/8 resolution.

VideoAsset

This defines which video file is being played. Right clicking this thumbnail will open the version control editor which allows for specifying which version of content is being used. For more information see Content versioning.

Prefetch

Prefetch options

This option allows you to change the Prefetch behaviour for a particular video clip. PrefetchLookAheads can be created that specify how many frames are prefetched ahead of time for that particular clip.
**Warning:** Use the prefetch options with caution. The prefetch settings are carefully configured to give optimal performance for disguise hardware. Altering them can have a serious impact on the playback capacity of the machine. Consult the disguise support team for further information.
Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.

Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.

**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.
Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colous. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.

Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.
Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value. Blending with white inverts the base colour values and blending with black produces no change.

Exclusion
Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

Colour Dodge
Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

Hard Mix
Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.
The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.

Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:

- Default
- Media
- Audio
- Colour Shift
  - Brightness (shift) 0
  - Contrast (scale) 1
  - Saturation scale 1
  - Hue shift 0
  - RGB controlled Together
  - Red minimum 0
  - Red maximum 255
  - Red gamma 1
  - Green minimum 0
  - Green maximum 255
  - Green gamma 1
  - Blue minimum 0
  - Blue maximum 255
  - Blue gamma 1
- Keying
- Move
- Crop
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

RGB controlled

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

Min

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.

*min* = 00 (left), *min* = 96 (right)
red.min = 0 (left), red.min = 96 (right)

Max

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.

Invert all (left), invert red (right)

Gamma

When gamma is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5
Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.

See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.
These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Controlling layers with Art-Net

This sub-chapter explains how to use expressions to control layer properties with Art-Net.

How to control layer properties with Art-Net

1. Open the layers editor. Please see the chapter Working with Layers for information on creating and editing layers.

2. Right-click the layer property you want to make Art-Net controllable. This will open a small editor.

3. Type the Art-Net expression into the **Expression** text field according to the format:

   \texttt{dmx:universe.channel}

   For example,

   \texttt{dmx:7.19}

When the expression has been accepted the layer property will turn green.
Art-Net expression (dmx:universe.channel) is used to make a layer property, in this example Brightness, Art-Net controllable.

This expression assumes each property to work within the 0-255 range. This is good for properties in the disguise software such as brightness and red min but for other properties such as saturation in a Video layer, the range should only be from 0-1 (you can set the saturation to more than 1 but it is not recommended).
How to enable the layer property to work only between the range 0-1

1. Set the min property to 0 and the max property to 1.

2. Type the following Art-Net expression into the untitled Expression text field:

\[(\text{dmx:7.19/255} \times (\text{max-min}) + \text{min})\]

This expression will automatically take the min and max values into account and will be fully controllable from the lighting desk with incoming DMX values between 0-255.

3. Set the default start value for the parameter in the lighting desk.
Controlling layers with OSC

To control a numerical OSC parameter in a layer, create an expression:

Right-click a parameter and type in the following syntax:

osc:.d3.parameterName*AValue

Tips for Setting up OSC

**Base Address:** check the base address in the disguise software matches the OSC path from the OSC sender/application.

Tip 1: The base address syntax within the disguise software omits the layer name but this should be included in the OSC path from the OSC sender/application.

Syntax: /d3/layer/[layer_name]/[layer_property], e.g. /d3/layer/show_intro/video

Companion: configure a 'GenericOscSender' and 'Send String'.

Tip 2: The base address in disguise defaults to the original layer name. After patching the layer, the layer can be renamed.

Example

For example, if you wish to control the brightness of a video layer type in:

osc:.d3.brightness*255
Frame: 0

Default
- Blend mode: Premult-Alpha
- Mapping: led1 (direct)
- Palette: hsvpalette

Colour X: 0
Colour Y: 0

Media
- Video: d3_test_4-3.png

Speed: 1
Mode: Normal
At end point: Loop
Transition time: 0

Expression: $\text{osc}::d3\cdot\text{brightness}\times255$
The reason for multiplying the incoming OSC value by 255 is because in this particular example the OSC command generates float values between 0.0-1.0.

The corresponding syntax for the brightness slider in TouchOSC is:

/d3/brightness

To enable OSC control, first make sure to set up the OSC input in the disguise software as described in the OSC Devices sub-chapter.
Expressions

Expressions for numeric animations

Expressions allow you to have a more powerful control over animation properties. They can be used with any numeric animation property.

**Warning:** the disguise software does not support expressions that uses layer names that has spaces between the words, for example "video explosions". In this case you have to rename it to "video_explosions". Expressions can only contain a-z A-Z 0-9 _

Setting up Expressions

To set up an Expression, right-click the property you want to control. This will open up the min/max/step viewer, which allows you to edit the limit values of the property and the value step used while animating. Below the min/max/step properties is a field called Expression, which initially contains a single word: `self`. This is the Expression.

The easiest Expression is `self`. It tells the system to use the value taken from the keyframe sequence. However, self can be built into more complex mathematical Expressions (formulae) to do useful things, such as:

`self * 2`

255 * self multiplies the animated value by 255.
When you type in an Expression, the property field turns green when the Expression has been correctly understood. If there is a mistake, the property field turns red; moving the mouse over the property field will open up a text message telling you what went wrong.

Expressions can refer to any property of any other object in the system. For example, if you want to read the x coordinate of the position of a screen called myscreen and multiply it by 2. The Expression to use would be:

```
screen:myscreen.pos.x * 2
```

In general, to access (property) of an object called (objectName) of type (type), we use:

```
type:objectName.property
```

---

**Expanded Functionality - r21**

In previous releases, Expressions could only access keyframed values. Module/layer expressions were hard-coded to only support field sequences.

As of r21, the functionality of Expressions has been improved, enabling them to be used in all modules in the disguise software.

Support for properties like `frameNumberDisplay` - the current frame readout at the top of the video layer- have been added for additional functionality.

This new feature allows for an expression to be created that has the ability to get information about a resource, and they can be used with all module properties which have existed previously and any new ones that are to be added.

An example of where this can now be used is in the case of fading the brightness down at the end of a clip; an expression can now be used on the brightness field that reads the frame index to do the fade.

Example expression used on Brightness field:

```
if(module:video.frameNumberDisplay >= 50, 0.5, 1.0)
```
Layer Types Overview

For a broader overview of layers please read the layers overview topic.

Types of layers

There are five main layer types in the disguise software: content layers, generative layers, pre-visualisation layers, effects layers and control layers.

Content layers

The function of content layers is to play back pre-rendered content, map them onto screens and, if needed, apply a number of effects built into the layer.

Audio
RenderStream Layer
TestPattern
Video
VideoTrigger
VirtualLineup
Web

Generative layers

Generative layers are packets of algorithms inside the disguise software that produce visuals based on a number of animatable properties. They share some of the same properties as all the other layer types
such as the Mapping property.

**Pre-visualisation layers** are usually used during the design and pre-production phases of the project when accurate visualisation is required, for example animating screens or creating camera fly-throughs.
Effects layers

Input a bitmap, a video or an output from another content layer into the disguise software effects layers. For example spin, pixelate, blur images, video, or other inputs to create a new look.

- Blur
- ChannelRouter
- ColourAdjust
- Compose
- EdgeFilter
- Fade
- FilmicGrain
- Kaleidoscope
- Lut
- MotionBlur
- Noise
- PixelMap
- Pixelate
- Scroll
- SpinBitmap
- Trigger
- VideoTransition

Control layers

Control layers are technical layers that control other equipment in the show; for example DVI matrices, switchers or unusual DMX controlled equipment.
CameraControl
Control
MatrixControl
DMXLightsControl
DMXShare
TimecodeMode
MasterBrightness
MTC
MDC
MidiNote
Open
OscControl
PlayMode
ProjectorControl
TargetObject
TrackJump
TransportControl
TransportBrightnessLocal
TransportVolumeLocal

Legacy layers

Legacy layers are kept in the software for compatibility purposes but have been replaced with more up to date layers.

Bitmap

Legacy video
Working with layers

It is highly recommended that the previous chapter Working with layers is read before reading this chapter. This is because creating/structuring/editing layers involves many processes, for example placing key-frames on the Key-frame editor, which are all explained in the previous chapter.
Layer Types Overview

For a broader overview of layers please read the layers overview topic.

Types of layers

There are five main layer types in the disguise software: content layers, generative layers, pre-visualisation layers, effects layers and control layers.

Content layers

The function of content layers is to play back pre-rendered content, map them onto screens and, if needed, apply a number of effects built into the layer.

- Audio
- RenderStream Layer
- TestPattern
- Video
- VideoTrigger
- VirtualLineup
- Web

Generative layers

Generative layers are packets of algorithms inside the disguise software that produce visuals based on a number of animatable properties. They share some of the same properties as all the other layer types such as the Mapping property.
Pre-visualisation layers

Pre-visualisation layers are usually used during the design and pre-production phases of the project when accurate visualisation is required, for example animating screens or creating camera fly-throughs.

AnimateCameraControl
AnimateCameraPreset
AnimateObjectPreset
TargetControl
TargetPreset

Effects layers

Input a bitmap, a video or an output from another content layer into the disguise software effects layers. For example spin, pixelate, blur images, video, or other inputs to create a new look.
Blur
ChannelRouter
ColourAdjust
Compose
EdgeFilter
Fade
FilmicGrain
Kaleidoscope
Lut
MotionBlur
Noise
PixelMap
Pixelate
Scroll
SpinBitmap
Trigger
VideoTransition

Control layers

*Control layers* are technical layers that control other equipment in the show; for example DVI matrices, switchers or unusual DMX controlled equipment.

CameraControl
Control
MatrixControl
DMXLightsControl
Legacy layers

Legacy layers are kept in the software for compatibility purposes but have been replaced with more up to date layers.

Working with layers

It is highly recommended that the previous chapter Working with layers is read before reading this chapter. This is because creating/structuring/editing layers involves many processes, for example placing key-frames on the Key-frame editor, which are all explained in the previous chapter.
Audio

The Audio layer allows you to play audio files from disguise, without the need for embedding those files into a video file first.

Workflow

1. Add the audio files to the audiofile folder in the disguise project folder.
2. Add an audio layer to the timeline.
3. Assign the layer an audio patch (assuming a logical audio out has already been configured).

Media
Audio
Opens the audio file list

Mode

Normal
When the play head stops, the audio will also stop and the frame number will lock to the timeline position. When the play head continues to play or holds at the end of a section, the audio will play continuously. Jumping around the timeline while playing does not affect which part of the audio is being played.

Locked
If the play cursor continues to play or stops at the end of a section, the audio locks to the timeline and thus stops playing.

At End Point

Loop
When playback reaches the end of the video clip, the clip will loop immediately back to the beginning and will start playing again from there.

Ping-pong
When playback reaches the end of the video clip, the clip will play in reverse back to the beginning.

Pause
When playback reaches the end of the video clip, the clip will pause on the last frame.

Transition Time
The transition property specifies the dissolve time, in seconds, when switching from one audio clip to the next on the timeline. By default, this is set to 0, so the player cuts immediately from one clip to the
next. Setting this value to 1 causes the player to dissolve from one clip to the next over 1 second.

Audio

Volume
You can specify any number between 0 (mute) and 255 (full) in the text field to set a constant volume, or open the properties key-frame editor to fade volume in or out.

Balance
You can balance the playback of a file to either right or left by altering the value of the parameter.

Output
This controls which audio output device the audio is played to.
Using the Bitmap layer is the easiest way to display still images. The Bitmap layer takes over certain properties from the Colour layer and the Video layer. To see where to place media files in Windows see the Placing media files for a project page. To see what video file formats the disguise software support see the Supported file formats page.

Please note: In release r13 onwards it is recommended to use a video layer for still images. For more information see the Placing media files for a project page.

The Bitmap layer supports the use of Arrows, allowing content layers to be piped into the Bitmap layer. For more information please see the sub-chapter Composing layers using Arrows.
This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is also alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

**Colour Shift properties are:**

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brightness (shift)</td>
<td>0</td>
</tr>
<tr>
<td>Contrast (scale)</td>
<td>1</td>
</tr>
<tr>
<td>Saturation scale</td>
<td>1</td>
</tr>
<tr>
<td>Hue shift</td>
<td>0</td>
</tr>
<tr>
<td>RGB controlled</td>
<td>Together</td>
</tr>
<tr>
<td>Red minimum</td>
<td>0</td>
</tr>
<tr>
<td>Red maximum</td>
<td>255</td>
</tr>
<tr>
<td>Red gamma</td>
<td>1</td>
</tr>
<tr>
<td>Green minimum</td>
<td>0</td>
</tr>
<tr>
<td>Green maximum</td>
<td>255</td>
</tr>
<tr>
<td>Green gamma</td>
<td>1</td>
</tr>
<tr>
<td>Blue minimum</td>
<td>0</td>
</tr>
<tr>
<td>Blue maximum</td>
<td>255</td>
</tr>
<tr>
<td>Blue gamma</td>
<td>1</td>
</tr>
</tbody>
</table>

[Image of Video settings showing Colour Shift properties]
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

RGB controlled

This property controls how the min/max levels and gamma corrections are applied. If set to together, then the min, max and gamma values are the same for the red, green and blue channels. If set to separate, the min,max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

Min

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing min equally across red, green and blue enforces shadows, whereas increasing min on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing red min shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): $g=0.5$, $g=1$, $g=1.5$

**Transition**

Determines the crossfade (in seconds) between two adjacent bitmaps placed in the keyframe editor of the Bitmap layer.

**Threshold**

The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.
Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key colour rgb

The value of the key colour, 0-255, causing the key colour to become transparent.

Scale mode

Provides mapping options to apply to content of a certain resolution which does not match the resolution of the targeted canvas.

Scale

This controls the size of the image. When set to 1, the image is displayed at its normal size. Increasing the scale increases the size of the image; to display at double size, set scale to 2. To halve the image size, set scale to 0.5.

Aspect

This controls the aspect ratio of the image. When set to 1, the image is displayed at a 1:1 aspect ratio. Increasing the aspect value increases the height of the image in relation to its width.

Position

This controls the horizontal (x) and vertical (y) position of the image. When both are set to 0, the centre of the image is centered in the output canvas. Setting x to 1 moves the center to the left edge of the canvas; setting it to +1 moves the center to the right edge of the canvas. Setting y to 1 moves the center
to the top edge of the canvas; setting it to +1 moves the center to the bottom edge of the canvas.

**Rotation**

This controls the rotation of the bitmap, in degrees.

**Crop**

The *crop* pulldown tab contains a set of properties (top, left, bottom, right) that allow you to remove pixels from the edges of the bitmap. Each property specifies the number of pixels you want to remove from the corresponding edge. For example, setting top to 10 will clip 10 pixels off the top edge.
TestPattern

The TestPattern layer allows users to generate basic test patterns within the disguise software.

Workflow

- Create a new TestPattern layer
- Assign the layer to a mapping

**Please note:** The test-pattern layer is unique in that it fills the entire area of the surface it’s mapped to, thus any mapping type applied to this layer will function as a direct map.
Define the circles & lines parameters to suit

TestPattern common layer properties
**Preview Thumbnail**

This property is a small preview window at the top of the editor showing you what the layer output looks like.

**Blend Mode**

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colour. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCol** and **yCol** values (see the section for **xCol**, **yCol**).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
max = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

Max

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**TestPattern layer properties**

**Circles**

Defines whether the test pattern has circles or not.

**Circle radius**

Defines the radius of the circles.
Label
Defines whether to show the screen/projector label in the pattern.

GridMode
Defines whether the pattern uses grids.

GridSpacing
Defines the spacing between the grid squares.

SubGridSpacing
Defines the spacing between the second layer of grid squares.
Video

The Video layer works in the same way to the Legacy video layer with the only exception of the speed property.

With the Video layer it is possible to adjust the speed of the video clip whilst the clip is playing. The playback speed will vary between the keyframe points unlike in the video layer.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multipies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multipies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xColl** and **yCol** values (see the section for **xColl**, **yCol**).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCOL, yCOL**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
\[ \text{min} = 00 \text{ (left), min} = 96 \text{ (right)} \]

\[ \text{red.min} = 0 \text{ (left), red.min} = 96 \text{ (right)} \]

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

\[ \text{max} = 255 \text{ (left), max} = 260 \text{ (right)} \]

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**Media**

The Video property points to the still media file that defines the content to be used on the layer. Selecting this property will open the VideoClip object library, which shows all of the still image files saved on your local hard-drive in the VideoFile folder.

To change the current piece of content:
1. Left-click **Video** to open the VideoClip object library.

2. Left-click the media file you want to use.

See the Placing media files for a project sub-chapter to understand where to place your content and how to access it in the disguise software. Also save the file to a supported file format.

**Speed**

Selects playback speed. When this is set to 1, the video is played at its normal framerate; when set to 2, the video is played at double speed, when set to -1, the video is played at normal framerate in reverse and so on.

**Mode**

This specifies the playback mode. There are three modes; each one has a specific behaviour that is useful for a different situation.

**Normal**

When the play cursor stops, the video will also stop and the frame number will lock to the timeline position. When the cursor continues to play or holds at the end of a section, the video will play continuously. Jumping around the timeline while playing does not affect which frame is being played.

**Locked**

If the play cursor continues to play or stops at the end of a section, the video frame number locks to the timeline. When the play cursor holds at the end of a section, the video will play continuously.

**At end point**

This specifies what happens when playback reaches the end of the video clip. There are three options to choose from:
Loop
When playback reaches the end of the video clip, the clip will loop immediately back to the beginning and will start playing again from there.

Ping-pong
When playback reaches the end of the video clip, the clip will play in reverse back to the beginning.

Pause
When playback reaches the end of the video clip, the clip will pause on the last frame.

Transition Time
The transition property specifies the dissolve time, in seconds, when switching from one video clip to the next on the timeline. By default, this is set to 0, so the player cuts immediately from one clip to the next. Setting this value to 1 causes the player to dissolve from one clip to the next over 1 second.

Volume
Some video clips may contain audio. You can fill in any number between 0 (mute) and 255 (full) in the text field to set a constant volume, or open the property's key-frame editor to fade volume in or out.

Output
If the video has an audio track, this controls which audio output device the audio is played to.

Threshold
The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.
Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key colour rgb

The value of the key colour, 0-255, causing the key colour to become transparent.

Scale

Controls the size of the video.

Aspect

Controls the aspect ratio of the video, higher numbers make the frame taller.

Pos xy

Position of the video; x = horizontal position, y = vertical position. 0 = centre of canvas, -1 = top edge of canvas, +1 = bottom edge

Rotation

Rotation (in degrees) of the video.

Crop left, right, top, bottom

Crops the frame from the respective edge. 0 = no crop, 1 = entire frame crop.
Vignette

Adds a vignette effect by softening the edges of the crop.
Legacy video

The Legacy video layer is used to play pre-rendered Quicktime video files. To see what video file formats the disguise software support see the Supported file formats page.
To see where to place media files in Windows see the Placing media files for a project page.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expansions of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xColl** and **yCol** values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

RGB controlled

This property controls how the min/max levels and gamma corrections are applied. If set to together, then the min, max and gamma values are the same for the red, green and blue channels. If set to separate, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

Min

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing min equally across red, green and blue enforces shadows, whereas increasing min on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing red min shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

Max

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**Media**

The Video property points to the still media file that defines the content to be used on the layer. Selecting this property will open the VideoClip object library, which shows all of the still image files saved on your local hard-drive in the VideoFile folder.

To change the current piece of content:
1. Left-click **Video** to open the VideoClip object library.

2. Left-click the media file you want to use.

See the Placing media files for a project sub-chapter to understand where to place your content and how to access it in the disguise software. Also save the file to a supported file format.

## Speed

Selects playback speed. When this is set to 1, the video is played at its normal framerate; when set to 2, the video is played at double speed, and so on.

*Please note:* playback speed is set at the point where the video keyframe is encountered in the video keyframe editor. Once this is set, the playback speed does not change, even if it varies over time. To change the playback speed, drop a speed keyframe at the point where the video thumbnail appears in the video keyframe editor.

## Mode

This specifies the playback mode. There are three modes; each one has a specific behaviour that is useful for a different situation.

### Normal

When the play cursor stops, the video will also stop and the frame number will lock to the timeline position. When the cursor continues to play or holds at the end of a section, the video will play continuously. Jumping around the timeline while playing does not affect which frame is being played.

### Locked

If the play cursor continues to play or stops at the end of a section, the video frame number locks to the timeline. When the play cursor holds at the end of a section, the video will play continuously.
**Freerun**

If the play cursor continues to play or stops at the end of a section, the video will play continuously. Jumping around the timeline while the cursor is playing or has stopped does not affect which frame is being played.

**At end point**

This specifies what happens when playback reaches the end of the video clip. There are three options to choose from:

**Loop**

When playback reaches the end of the video clip, the clip will loop immediately back to the beginning and will start playing again from there.

**Ping-pong**

When playback reaches the end of the video clip, the clip will play in reverse back to the beginning.

**Pause**

When playback reaches the end of the video clip, the clip will pause on the last frame.

**Transition time**

The transition property specifies the dissolve time, in beats, when switching from one video clip to the next on the timeline. By default, this is set to 0, so the player cuts immediately from one clip to the next. Setting this value to 1 causes the player to dissolve from one clip to the next over 1 beat.

**Volume**

Some video clips may contain audio. You can fill in any number between 0 (mute) and 255 (full) in the text field to set a constant volume, or open the property's key-frame editor to fade volume in or out.
Output

If the video has an audio track, this controls which audio output device the audio is played to.

Threshold

The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.

Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key color RGB

The value of the key colour, 0 255, causing the key colour to become transparent.

Scale

Controls the size of the video

Aspect

Controls the aspect ratio of the video, higher numbers make the frame taller.

Pos x y

Position of the video; x = horizontal position, y = vertical position. 0 = centre of canvas, -1 = top edge of canvas, +1 = bottom edge
Rotation

Rotation (in degrees) of the video.

Crop left, right, top, bottom

Crops the frame from the respective edge. 0 = no crop, 1 = entire frame crop.

Vignette

Adds a vignette effect by softening the edges of the crop.
RenderStream

RenderStream is the proprietary disguise protocol for controlling third party render engines from the disguise software.

Overview

RenderStream allows for the sequencing and sharing of content from a third party render engine to the disguise software, much like how Notch works onboard a gx, albeit running on the rx, our proprietary external render node.

This topic is intended to outline the basic steps involved in configuring RenderStream for use with Unreal Engine, Unity, and Notch. Once the configuration steps explained here are complete, sequencing of the third party render engine can be accomplished via the RenderStream Layer within the disguise software.

Uncompressed vs. Compressed

RenderStream Uncompressed requires the use of a 25G Mellanox network interface to stream uncompressed, 10bit video data. To test RenderStream Uncompressed appropriately, access to multiple machines and the networking equipment specified is required. Testing RenderStream Uncompressed also requires both a RenderStream Send License and a RenderStream Uncompressed License.

RenderStream Compressed provides consistent content quality with advanced H.265 compression. You can now choose high frequency compression, and ensure that even the finest details of your content are shown via our superior, reliable networking.
Plugins

In order to communicate with disguise, both Unreal Engine and Unity require the installation of a plugin on the render node. Visit the disguise Github for the latest plugins.

Cluster Rendering

**Warning:** For cluster rendering it's recommended to use render nodes from the same disguise product range, e.g. all rx series machines. Mixing of machines from different product ranges is not recommended and is unsupported. It is acceptable to mix rx and rxII types however.

Here are just some of the benefits of using Cluster Rendering:

1. Cluster Rendering allows you to span your render engine content over more than one disguise server by scaling out real-time content up to an unlimited capacity.

2. Each machine will let you render a fragment of your final content frame to increase the render power and get your content onto your displays at your desired quality.

3. Use Cluster Rendering to render real-time content of the highest quality, detail and framerate without worrying about GPU power.

Cluster Rendering is configured within the disguise software using the RenderStream Layer.
VideoTrigger

The Video layer works in the same way to the Legacy video layer with the main exceptions being that this layer does not support Audio playback, and does not have a playmode, at end point and transition time property.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expantions of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCol** and **yCol** values (see the section for **xCol**, **yCol**).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xColl, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:
**Saturation scale**

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

![Saturation scale](image)

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

**Hue shift**

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**Media**
Trigger

The trigger property allows for a video file to be triggered for playback either when a value changes from 0-1 (or 255 if using a DMX lighting console) (OnReset), or changing to another value (OnChange).
OnReset will cause the video file to playback once before returning to no content being shown. If you wish to trigger the file again, the trigger value needs to be reset to 0 before another value of 1 can be sent.

OnChange will cause the video file to playback when the trigger value changes to a new value. Trigger Delay time can be added to delay the start of the video until a specific frame count.

**Video**

The Video property points to the still media file that defines the content to be used on the layer. Selecting this property will open the VideoClip object library, which shows all of the still image files saved on your local hard-drive in the VideoFile folder.

To change the current piece of content:

1. Left-click Video to open the VideoClip object library.
2. Left-click the media file you want to use.

See the Placing media files for a project sub-chapter to understand where to place your content and how to access it in the disguise software. Also save the file to a supported file format.

**Speed**

Selects playback speed. When this is set to 1, the video is played at its normal framerate; when set to 2, the video is played at double speed, when set to -1, the video is played at normal framerate in reverse and so on.

**Colour Shift**

Colour management controls for the Video trigger layer.

**Keying**

The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.
Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key colour rgb

The value of the key colour, 0–255, causing the key colour to become transparent.

Size

Controls the size of the video.

Scale x, y

Controls the aspect ratio of the video, higher numbers make the frame taller.

Pos x,y

Position of the video; x = horizontal position, y = vertical position. 0 = centre of canvas, -1 = top edge of canvas, +1 = bottom edge

Rotation

Rotation (in degrees) of the video.

Crop left, right, top, bottom

Crops the frame from the respective edge. 0 = no crop, 1 = entire frame crop.
Web

The Web layer provides the ability to render HTML5 webpages as content in the disguise software. However, disguise goes further and allows you to 'hook' and integrate your web pages to the time.

Key features of the Web layer

- Bookmarks for navigation (with thumbnails)
- Ability to send ‘Javascript Commands’ to the open web page (see field called Command) – i.e. PageTransitions.nextPage(2, 1)
- Ability to send 5 custom timeline float values into the page via JavaScript. Provides the ability to animate items in your web page in sequence with the timeline, or change font sizes.
- Ability to send various standard timeline clock information into the page via JavaScript
- Support for transparent background pages. E.g. body { background-color: rgba(0, 0, 0, 0); }, hence allowing you to composite items into disguise (like a twitter feed).
- Support for simple user interaction with the page from the disguise software3 (see the ‘Interact’ button at the top of the Web Layer).
- Locally hosted HTML5 pages

Browser feature support

The browser has the following attributes:
Utilises software renderer (as the disguise software has exclusive lock of the GPU)

Browsers are 'sand boxed' in separate processes to prevent negative impact on the disguise software

Supports HTML5

Utilises WebKit -> Based off Chromium (check "about:version" URL for exact version using this path: Interact → URL: “about:version” → go)

Adobe Flash is not supported, but the system is capable of running limited Flash Plugin functionality. Flash must be installed independently on the machine in question.

WebGL is not supported

three.js and other libraries that utilise hardware GPU rendering are not supported

**Warning:** Layer instances reset every time universal crossfade is triggered. This means that web layers will fade out/into themselves even if the same page is used across sections.

**Preview Thumbnail**

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

xColl, yColl

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. Colour X controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. Colour Y controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the Colour Y value to 255 and use 0 for the Colour X value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, Colour Y controls saturation, and Colour X controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

RGB controlled

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

Min

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**Interact**

Opens an interaction window so that you can control the web page in real time.

**Bookmark**

The bookmark for the web page you wish to browse. If you change the bookmark along the timeline, the new webpage will load in the background and then be presented.
Command

The Javascript command you wish to issue. You can fire commands along a timeline that manipulate the site, such as turning the page, etc.

Size

Two modes:

Match Mapping - this sets the size of the website canvas to match that of the Mapping it is being applied to.

Custom - allows you to manually set the size (Width/Height) of the web browser. The browser canvas will then be scaled up to match the Mapping.

Custom Parameters

Name X: The name of the JavaScript function you wish to call every frame. The function name in Javascript will be d3.customParam1(x)

Value X: The value you wish to pass into the JavaScript function.

Clear Cache

Clears the web browsers cache

Reset Browser

Tears down and resets the web browsing session
Bookmarks

Bookmarks are the way you select your web layers content.

The options in a bookmark are:

**URL**

This address can be a web address or file address.

- Web addresses are prefixed with http:// or https://
- File addresses relate to HTML5 files contained within your d3 project folder structure. All locally stored html5 files should be under project_folder/objects/Web/sub_folder. For example, a sample page is located in every new project at "file:///sample pages/slide heading.html".

**Please note:** To check the chromium version on your machine, create a web bookmark that points to chrome://version/

**Background**

The web layer is capable of rendering pages with transparent backgrounds to allow you to composite, or with white backgrounds.

For transparent backgrounds to function correctly, the web page must set its background to be transparent.

**Example CSS:**

```css
body { background-color: rgba(0, 0, 0, 0); }
```

**Zoom**

Allows you to text zoom with correctly rendered fonts. This is similar to pressing Ctrl+/- in a regular browser.
Interact

You are able to directly interact with web pages using a small browser. To open the browser, click the interact button.

While this window has focus, you are able to interact with the webpage including:

- Typing
- Clicking links
- Scrolling with the mouse wheel

You can change the size of the interaction window with the Scale or 50%/100% buttons.

JavaScript Commands

Creating JS commands is very straightforward.

1. Create a JS Command and fill in the javascript you want executed in the browser.
2. Place the command on the timeline at the time you want it executed.

When the timeline is playing and crosses over the keyframe for the command, it will be executed.

JavaScript Functions

The Web layer contains five pairs of Name/Value.

Every frame disguise will call a function in your web page with ‘Name’ entered in that string (in the ‘d3.’ namespace), passing it a single float ‘Value’.

d3.customParam1(1.0)

Essentially disguise is ‘pushing’ the data into your webpage.
You can then write the corresponding function to do with as you please. NB: functions must be in the d3 namespace.

```javascript
var d3 = {
    customParam1: function(myVal)
    {
    /* Do something with myVal */
    }
};
```

You can download an example project using JS/HTML from our Resources page in the Web Interaction section at this link.

**Standard JavaScript Functions**

These javascript functions are designed to allow the web page to respond to changes to the disguise timeline. They are called every frame on the web browser. If the HTML5 page has been coded to respond to these functions, then it will accordingly.

- `d3.lockedTime(float)` -- Time in beats within the current layer.
- `d3.normalTime(float)` -- Time in beats within the current layer, but carries on counting up. (See video player behaviour to get the idea)
- `d3.globalTime(float)` -- Time in beats within the whole track
- `d3.bpm(float)` -- Current BPM of the track
- `d3.status(string)` -- Provides the play head status. Either: "playing", "holding", "stopped"
- `d3.dmxUniverse(array[int], array[array[int]])` -- Provides a dump of the universes selected in the layer editor

Proprietary video codecs (like h264) are not supported currently.
CameraControl

The CameraControl layer controls Sony Visca Cameras configuration.

CameraControl layer properties

**Camera**

The camera object to use.

**Command**

The camera command to send.

Creating/sequencing CameraControl commands

The process used to create/sequence CameraControl commands is done by placing key-frames onto the Command key-frame editor, and is therefore the same as any other layer type that uses key-frame editors, for example the AnimateCameraPreset layer. For an example on how to create and animate key-frames using the key-frame editor please follow the step-by-step instructions in the AnimateCameraPreset sub-chapter.
There are three types of CameraControl commands:

Editing CameraControl commands

Each CameraControl command type opens a specific camera command editor.

**CameraConfig editor**

- **Zoom**: sets the cameras zoom, 0 100%
- **Focus**: sets the cameras focus, 0 100%
- **Exposure mode**: sets the exposure mode to auto, manual, iris or shutter.
- **Shutter**: sets the shutter value, 0 100%. Only valid in shutter mode.
- **Iris**: sets iris value, 0 100%. Only valid in iris mode.
- **Gain**: sets the cameras gain, 0 100%.
- **Pan**: sets the cameras pan, in degrees.
- **Tilt**: sets the cameras tilt, in degrees.
Camera Focus
sets the cameras focus, 0 100%

CameraOnOff editor
Switch : turns the camera on or off.

CameraRecallPreset editor
Preset : sets the preset number assigned to the Sony Visca Camera. A Sony Visca Camera can have up to four presets.
Control

The Control layer is a tool for executing custom commands on 3rd party products via the following protocols - Serial, Telnet or UDP.

The Control layer allows for the definition of custom commands that can be sequenced on the timeline to perform functions at certain times. The custom commands are command strings which can be written in normal character format (ASCII) or in Hex.

Command

The selected command to be executed at this point on the timeline.

The variable to be passed into the command, if the command syntax contains the use of variables. The values can be sequenced on the timeline. Use Auto Resend to control when values are sent.

Auto resend

When set to ‘On’ the command will be executed every time the variable changes, as well as on normal command issuing.

Control Device

The protocol device through which the command should be issued, either a Serial, Telnet or UDP device.

Please note: Ensure you that your protocol device is in the Device manager. Often it is better to create the devices from the manager and then select them in your layers, to avoid confusion.

Command Syntax
Commands are made up of a string. The string can be ascii or hex. The \ escape character is used to send special data.

The command can contain a variable set in the Control Layer, which is a timelined event.

<table>
<thead>
<tr>
<th>Special Characters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>Variable value as string</td>
</tr>
<tr>
<td>%</td>
<td>Variable value as binary byte</td>
</tr>
<tr>
<td>\13</td>
<td>Carriage return</td>
</tr>
<tr>
<td>\10</td>
<td>New Line</td>
</tr>
<tr>
<td>\xYY</td>
<td>Send YY as hex value, example, \xF5</td>
</tr>
<tr>
<td>\</td>
<td>Send special ASCII character (0-255). Used to send characters outside the normal range, e.g 1-31, 96 and up. Example, \37 will send '%’ character.</td>
</tr>
</tbody>
</table>

A useful ASCII/Hex lookup table can be found here, [http://www.ascii-code.com/](http://www.ascii-code.com/).

Ascii examples

- To send the command "START 568" followed by the enter key we would use the command string: START 568\13

- To send the command "CUE XXX" where XXX is the variable value, we would use the command string: CUE \$
Hex examples

- To send the byte values of 42, 62, 93 in sequence, you would use the string: \x2A\x3E\xD

- If you wished for the last value to be the variable value in a byte/uchar fashion you would use: \x2A\x3E\%

Real World Example

To control a Lightware matrix you would use the following examples:

- Load preset (using a variable) via telnet: {\$\%}

- To route a particular input (using a variable) to output 1: {1\$}
DMXLightsControl

The DMXLightsControl layer can add DMX commands and animate their values using a key-frame editor.

⚠️ Warning: the DMXLightsControl layer can only output DMX when a DMX device is set up. See DMX devices for information on setting up a device.

DMXLightsControl layer properties

Dmx Command 1-8

The DMX command to send.

The image above shows that the DMXLightsControl layer has eight \textit{dmx command} properties. By left-clicking the property a key-frame editor will open. It is important to note that only one DMX command can be added per key-frame editor, so a maximum of eight different DMX commands can be added in one DMXLightsControl layer.

However, multiple DMXLightsControl layers can each contain a set of specific DMX commands. For example, one DMXLightsControl layer can include DMX commands that control the beam effects of a particular fixture, whilst another DMXLightsControl layer can control the strobe options, etc.
Value 1-8

The animated value to substitute into the command.

The image below shows that the gobo DMX command is set to send a value of 128 on channel 8 to the mac2k lighting fixtures. However, by using the Value 1 key-frame editor, as shown in the image above, the value of the gobo DMX command is able to change over time.

Creating DMXLightsControl commands

The process used to create DMX commands is done by placing key-frames onto the Dmx Command key-frame editor, and is therefore the same as any other layer type that uses key-frame editors, for example the AnimateCameraPreset layer. For an example on how to create and place key-frames using the key-frame editor please follow the step-by-step instructions in the AnimateCameraPreset topic.

Editing DMXLightsControl commands

Right-click a DMX command from the DmxCommands manager to open the DmxCommands editor.

Lamp or fixture commands require you to send a particular value on a particular channel to, for example, control the focus, iris or gobo. These commands are created using the DmxCommands manager which displays a list of all of your DMX commands, and configured using the DmxCommands editor inside the driver editor of the fixture. Therefore for information on the properties of the DmxCommands editor please scroll down to the section DmxCommand in the GenericLampDrivers section.
DMXShare

The DMXShare layer enables you to specify whether you want to:

- control all DMX channels from the disguise software
- control the colour of the DMX fixture only
- control the position and orientation of the DMX fixture only

The advantage of the DMXShare layer is that in some shows you may want to control lights using both a conventional lighting desk and the disguise software. For example, disguise is an extremely useful tool for creating pixel mapping content for the lights using the advanced content mapping system, but also for sequencing lights exactly to the beat by quantising an audio track. The disguise software is also very good for controlling the movements of groups of lights. However, lighting desks give you more control over the movement of individual lights and are specialised to control lights very quickly. Based on these facts, there is a requirement to share the control of the lights between the disguise software and a lighting desk.

DMXShare layer properties

Output

Controls what properties of a DMX fixture the disguise software should control and output. Options are all control all properties of the fixture; colour only control the colour of the fixture only; and position only control the position and orientation of the fixture only (applies to moving head fixtures).

Enable blackout

Controls whether the DMX fixture/screen is muted (blacked out) from within the disguise software.

For advice on what hardware to use for merging DMX, please contact the Support team.
MatrixControl

The MatrixControl layer allows you to animate matrix presets on the Timeline. Currently, the disguise software has built in matrix devices for a number of matrices.

MatrixControl layer properties

**Matrix**

The matrix device you want to control.

**Command**

The preset you want to trigger.
Creating & sequencing MatrixControl commands

The process used to create/sequence DVI matrix presets is done by placing keyframes onto the Command keyframe editor, and is therefore the same as any other layer type that uses keyframe editors, for example the AnimateCameraPreset layer.

There are three types of matrix presets:

Select type of matrix command
Search:
MatrixCommandAutoTrans
MatrixCommandCut
MatrixCommandPreset

Editing MatrixControl commands

Each MatrixControl command type opens a specific command editor.

MatrixCommandAutoTrans
Transition with a specified transition time. Only applicable for the Barco Encore.

MatrixCommandCut
Transitions with a hard cut. Only applicable for the Barco Encore.
**MatrixCommandPreset**

Changes the *preset* of the matrix. This will only work if you already configured a DVI matrix.

---

**Using the MatrixControl layer**

1. First, create a new matrix device in the Device manager, for example Lightware matrix or BarcoMatrixPro. Make sure the ip-address of the matrix device is in the same network as your Director/Actor network.

2. Create a new MatrixControl layer.

3. Choose the matrix device you just created.

4. Create a new command. For a matrix preset, choose the **MatrixCommandPreset**

5. Type in the number of the preset.

6. To animate presets, drag and drop them onto the keyframe editor.
**TimecodeMode**

The TimecodeMode layer allows you to control whether the disguise software is chasing timecode or ignoring timecode. If it is set to default then the TimecodeMode layer will have no effect.

The layer allows you to change between the following timecode modes:

- **Default** - The layer will have no effect.
- **Ignore** - The layer will ignore incoming timecode for the duration of where the layer covers.
- **Chase** - The layer will chase timecode. The chase mode is effectively saying "The timecode tags on this area of the timeline will be triggered".
MasterBrightness

The MasterBrightness layer allows you to control the output (master) brightness of a number of layers without having to individually change their brightness levels. You can also use it to control the output brightness of any Screen or collection of Screens.

Warning: once the master brightness of a screen is set, it will retain that value until another MasterBrightness layer resets it. A common mistake is to set the master brightness of a screen to 0, then jump to another part of the timeline without a MasterBrightness layer and wonder why there is no output on the screen. For this reason, the use of the MasterBrightness layer is discouraged except in emergencies.

As you can see below, the output brightness is only visualised in the Output Feeds level.
MasterBrightness layer properties

Brightness

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.
MidiNote

The MidiNote layer sends a midi note.

MidiNote layer properties

Device
The midi port to use.

Channel (1-base)
The channel to send the note.

Note
The note to send, in decimal format.

Velocity
The velocity value of the note.
MDC

The MDC layer allows you to control a system of multiple displays over a serial connection, TelNet or UDP from within the disguise software.

MDC, the free multi-display control software created by Samsung, is used to control and exchange information between a variety of different sources including Samsung displays and the networked computers used to control the displays.

Three types of devices can be configured from the MDC layer:

- Serial
- TelNet
- UDP
MTC

The MTC layer allows you to output MIDI timecode in a controlled way so that external devices such as lighting desks can be synchronised precisely with events on the timeline.

The MTC layer lets you set a timecode start value from the time position on the timeline, a frames-per-second value and an output device. When the play cursor first enters the extent of the layer, the disguise software starts a clock at the value specified and then outputs timecode as long as the play cursor remains under the layer extent.

If moving the play cursor away from under the extent of the layer will stop the output clock. When the play cursor re-enters the layer extent, the clock will restart.

MTC Layer Properties

iDevice

This specifies the index of the MIDI device used to output timecode. In the standard disguise machine, this should always be set to AIO Midi. The other option is Microsoft GS Wavetable Synth.

Type

Choose from the following options: 24, 25, or 30 frames per second.
Start Hour, Min, Sec, Frame

This specifies the start time of the output clock. Whenever the play cursor enters the layer extent, the output clock is reset to this value.

Play mode

There are two options:

**Locked**: sets the timecode value to the position of the play cursor on the Timeline.

**Normal**: allows the time to ticks upwards regardless of cursor position on the Timeline. For example, if the play mode is set to Play-to-end-of-section, where the Play cursor will only progress until it reaches the end of a section (i.e. just before the next cue point) and disguise enters in a holding state, the time code will still tick upwards.
Open

The Open layer is the Swiss army knife of layers. It allows you to control any property of any object in the disguise software, even if that object was not previously animatable using a Key-frame editor. This ability to use a Key-frame editor for properties which were not previously animatable is the key advantage of the Open layer.

Warning: use the Open layer carefully. A common mistake is to have a short Open layer on the Timeline that changes the value of that property and then move the play cursor somewhere else on the Timeline. You might, for example, change the output master brightness to 0, then move the cursor to another location and wonder why there is no output.

When you first create an Open layer, all you will see is a title bar, with no properties beneath it.

To control a property of another object:

1. Right-click an object to open its object editor, in this example Screen 1.

2. If you want to open multiple editors click Ctrl on the objects of interest.

3. Hold down Alt, left-click and drag from the Open layer title bar, you will see a white arrow appear. For more information on Arrows please see the Composing layers with Arrows topic.
4. Drag the arrow-head over the property you want to control, in this example `scale`.

5. Release the mouse button and Alt; the Open layer now has a new property named after the property you dragged the arrow to. By left-clicking the property a key-frame editor will open. This can be used to animate the property.
Scale property has now become a property of the Open layer
OscControl

The OscControl layer is a tool for sending custom commands to 3rd party products via Open Sound Control (OSC).

The OscControl layer allows for the definition of custom commands that can be sequenced on the timeline to perform functions at certain times. The custom commands are OSC messages defined by a fixed OSC address and any amount of OSC arguments (Integer, Float, String, Blob).

Command

The selected command to be executed at this point on the timeline.

Variable 1-4

The variable to be passed into the command, if the command syntax contains the use of variables. The values can be sequenced on the timeline. Use Auto Resend to control when values are sent.

Auto resend

When set to 'On' the command will be executed everytime the variable changes, as well as on normal command issuing.

OSC Device

The protocol device through which the command should be issued, either a Serial, Telnet or UDP device.

Please note: Ensure you that your protocol device is in the Device manager. Often it is better to create the devices from the manager and then select them in your layers, to avoid confusion.
Command Syntax

Commands are made up of an OSC address and any number of OSC arguments. The arguments have a type (Integer, Float, String or Blob) and a string value.

The string value of Integer and Float arguments can contain a variable set in the Control Layer, which is a timelined event, or a fixed numerical value. $1 through $4 define which variable to use.

The string value of a Blob argument should be in hex.
PlayMode

The PlayMode layer overrides the current play mode set in the Timeline.

PlayMode Properties

Play Mode

The play mode to be in. The options are:

Any: this mode has no effect; the current play mode specified in the Timeline will not be overridden.

Play: this mode does not respect section boundaries; when the end of a section is reached, the cursor continues into the next section, stopping only at the end of the track.

Play to end of section: in this mode the Play cursor will only progress until it reaches the end of a section (i.e. just before the next cue point).

Loop section: in this mode, the Play cursor will progress until it reaches the end of a section, i.e. just before the next cue point. At this point, the cursor will loop back immediately to the beginning of the section and will continue playing from there.

For more information on these play modes see the sub-chapter Navigating the Timeline.
TrackJump

The TrackJump layer allows you to automatically jump between tracks. This can be useful in situations where you have a permanent or semi-permanent fixed installation that needs to continue running all day, switching from one track to the next or jumping between sections within the same track.

This layer works as follows: whenever the play cursor enters the extent of the layer and the cursor continues to move forward (and thus does not stop), the disguise software will jump to the track and section specified in the TrackJump layer and perform the specified action.

TrackJump Layer Properties

Track

This specifies which track the disguise software should jump to. If set to None or left empty, no track jump is issued and the disguise software stays in the same track.

Section

This specifies which section number the disguise software should jump to; 0 means jump to the first section.

Play mode

This specifies the transport control mode that the disguise software should enter when jumping to its destination. The options to choose from are stop, play, play-to-end-of-section, and loop section.
TransportControl

The TransportControl layer is tool for triggering secondary transports from another transport.

The TransportControl layer is used for triggering other transports from the timeline. One important factor to be aware of with this layer, is that it cannot control it's own transport. Use of this layer requires multiple transport managers, assigned to a different Set List with different sets of tracks within each Set List.

TransportControl properties

Transport - defines the transport manager that should be triggered (cannot be the transport manager that contains the track that this layer is on).

Command - Allows for sequencing of TransportJumpCommand keyframe objects. This object specifies the Command (Play, Play to end of section, Loop section and Stop), the track to target, the location within the track (by specifying a cue number or timecode tag) and a crossfade duration.

Trigger - Defines whether the command is triggered Always or When playing.
TransportBrightnessLocal

The TransportBrightnessLocal layer replaces the need to use the Open layer to control the master brightness via keyframes changes on a track. The layer will work whenever it is on a track which is being played by the currently active Transport.
TransportVolumeLocal

The TransportVolumeLocal layer replaces the need to use the open layer to control the master volume control via keyframes changes on a track. The layer will work whenever it is on a track which is being played by the currently active Transport.
Blur

The Blur layer blurs the content coming from another layers output.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
The Blur layer supports the use of Arrows, allowing content layers to be piped into the Blur layer. For more information please see the sub-chapter Composing layers using Arrows.

Common Layer Properties

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.

There are two ways of expressing alpha in an image:
Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.

**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.
**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.

**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.
**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value. Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.
**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Blur layer properties**

**Radius**

The amplitude of the blur effect. Default value is 6. The min/max values are 0 and 50. Decreasing the value from 6 to 0 will sharpen the content, whilst increasing the value from 6 to 50 will gradually amplify the blurriness of the content.

**BlurMask**

Applies blur based on a gray-scale still image. White areas will generate maximum blur; black areas will ignore blur.
ChannelRouter

The ChannelRouter layer creates an RGBA image made up from any combination of the RGBA components from up to two input sources. This can be used to correct problems with a source image, to verify the components of the input sources or for many artistic purposes.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Please note: The order and direction of arrowing between layers is important. Drag a pipe from the content source layer into the ChannelRouter layer to achieve the result shown in the image.

Potential uses for the ChannelRouter Layer

Alpha channel substitution

- Take the alpha from one input and apply it to the other. This can be used for artistic masking effects. (Settings: R=Source 1 Red, G=Source 1 Green, B=Source 1 Blue, A=Source 2 Alpha)

Alpha channel removal

- Remove the alpha channel from a source image by setting the layer’s alpha component to a constant value of 1. Settings: R=Source 1 Red, G=Source 1 Green, B=Source 1 Blue, A=Constant 1

Synthesising an alpha channel

- Input sources that you require to have an alpha channel but do not can often have satisfactory alpha channels synthesised from their luma channel. Settings: R=Source 1 Red, G=Source 1 Green, B=Source 1 Blue, A=Luma 1

Channel correction

- An image supplied in BGR format can be fixed by rearranging it to RGB. Settings: R=Source 1 Blue, G=Source 1 Green, B=Source 1 Red, A=Alpha 1

Image debugging

- Separately view each component from an input source to verify that they are all appear as expected. Settings: R=Source 1 Red, G=Source 1 Red, B=Source 1 Red, A=Constant 1
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiply out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

### Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

R Component
Determines which input component appears in the red channel of the layer's output.

G Component
Determines which input component appears in the green channel of the layer's output.

B Component
Determines which input component appears in the blue channel of the layer's output.

A Component
Determines which input component appears in the alpha channel of the layer's output.

The value for each of the four component properties above can be selected from the following list:

- Source 1 Red - Red component of source 1
- Source 1 Green - Green component of source 1
- Source 1 Blue - Blue component of source 1
- Source 1 Alpha - Alpha component of source 1
- Source 1 Luma - Luma (effectively similar to greyscale) value of source 1
- Source 1 Max RGB - Strongest of source 1’s red, green and blue components
Source 1 Min RGB - Weakest of source 1's red, green and blue components

Source 2 Red - Red component of source 2

Source 2 Green - Green component of source 2

Source 2 Blue - Blue component of source 2

Source 2 Alpha - Alpha component of source 2

Source 2 Luma - Luma (effectively similar to greyscale) value of source 2

Source 2 Max RGB - Strongest of source 2's red, green and blue components

Source 2 Min RGB - Weakest of source 2's red, green and blue components

Constant 0 - A constant value of 0

Constant 1 - A constant value of 1
ColourAdjust

The ColourAdjust layer is best suited to changing colour properties in a familiar set values.

Overview

The ColourAdjust layer is designed to control white balance and tint either through presets or discrete amounts.

Workflow

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
1. **Add a ColourAdjust layer to your track.** For more information about adding layers see [creating layers](#).

2. **Arrow from your media layer to the ColourAdjust layer** so the compositing order looks something like the following image.

![Video layer piped into a ColourAdjust layer on the timeline.](image)

For more information on arrowing, see [compositing layers](#).

3. **Adjust the ColourAdjust layer properties** to suit your desired sequencing.

### Common layer properties

#### Blend mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

Exclusion

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

Colour Dodge

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

Hard Mix

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

Brightness

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**White balance preset**

This property defines which White balance preset is used.

The following options are available.

- None
- Custom
- Daylight
- Shade
- Cloudy
- Tungsten
- Florescent
- Flash

Once you have selected a preset, you cannot keyframe Kelvin & Tint.

**Kelvin**

This property controls the Kelvin in degrees kelvin.
Tint

This property controls tint and allows you to fine-tune the green/magenta balance. The scale on the slider represents the actual Kelvin value, which is subject to slight variations from camera to camera.

Contrast

This property controls contrast.

Brightness

This property controls brightness.

Saturation

This property controls saturation.

Levels

Levels is a tool in the ColourAdjust layer which can move and stretch the brightness levels of an image histogram. It has the power to adjust brightness, contrast, and tonal range by specifying the location of complete black, complete white, and midtones in a histogram. Since every piece of content’s histogram is unique, there is no single way to adjust the levels for all your content. A proper understanding of how to adjust the levels of an image histogram will help you better represent tones in the final image.
Compose

The Compose layer takes multiple inputs, for example live camera input, and composes the inputs next to each other on a screen. The Compose layer can also blur the edges between the inputs enabling seamless edges.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
The Compose layer supports the use of Arrows, allowing content layers to be piped into the Compose layer. For more information please see the sub-chapter Composing layers using Arrows.

Using the Compose layer

1. Create the input layers. These can be any content layer (Bitmap layer, Video layer, Video Transition layer, or a combination). For the example here we are using Bitmap layers.
Content layers, in this example four Bitmap layers, will be used as input into a Compose layer

2. Edit the content layers properties as you wish. The content layers used in this example have had their Colour Shift properties edited. For more information on editing Colour Shift properties scroll down to the section Colour Shift in the Common layer properties chapter.

3. Create a Direct mapping with the same resolution as the source content. For example, if the image in the Bitmap layer is 256x256 pixels, create a Direct mapping type with a 256x256 resolution. As the mapping type only acts as a temporary container which is piped into the Compose layer, there is no need to add a screen.

4. Create a Compose layer. Make sure this layer is positioned at the top of your layer order.

The Compose layer should be positioned at the top of the input layers

5. Drag an Arrow from each input layer into the Compose layer. For information on Arrows please see the sub-chapter Composing layers using arrows. In the Compose layers preview window you can now see the inputs composed next to each other.
Input layers piped into the Compose layer using Arrows and mapped onto Screen1 using a Direct mapping

6. Edit the Compose layer properties as you wish. It is also possible to blur the edges between the inputs enabling seamless edges, as shown in the first image of this sub-chapter.

**Blend Mode**

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expansions of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

 Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colous. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCol** and **yCol** values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCOL yCOL**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Horizontal shift

Moves the composed content left or right.

Rotation speed

Adjusts the horizontal rotation speed of the content.
Vertical shift

Moves the content up or down.

Blend border

Adjusts the blending level between the input sources.

Source 1,2,3,4

Adjust what inputs to show, hide or partially show. When you set the value to 1, it shows the input, 0 hides the input. As inputs are being hidden, note that the disguise software maintains the aspect and therefore scales the inputs on the screen.
EdgeFilter

The EdgeFilter layer filter can be used to detect the edges within a video/bitmap.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the [compositing layers topic](#).

To draw an arrow between two layers, hold down **ALT** and left click & drag between the source and destination layer.

By adjusting the strength parameter you can change how the layer affects the image. Setting the strength to 0 will have no effect. Setting the strength to 1 will highlight the edges and exclude the rest of the image. Setting the strength to -1 will exclude the edges and highlight the rest of the image.
EdgeFilter layer set to 0.
EdgeFilter layer set to 1.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Strength**

Strength controls how strong the edge filter is applied.
Fade

The Fade layer allows control of the brightness of all screens within a mapping.

Overview

The Fade layer allows control of the brightness of all screens within a mapping.

Fade Layer Properties
Group 0..7
The mapping to use as the group.

Fade 0..7
The brightness of all layers outputting to the mapping. 0 is black, 1 is maximum brightness. The example above shows the Fade layer being used to fade the brightness of two screens.

Colour x, colour y, and fade are applied to all screens within all groups.

Common Layer Properties

Brightness
This property (which appears as a light bulb icon) controls the brightness of the layer output.

Palette
This points to the still image file that defines the bitmap from which the tint colour is taken.

The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the Colour X and Colour Y values.

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

Colour X, Colour Y
These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using
the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors.

Colour X controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge.

Colour Y controls the vertical position, where 0 is the top edge and 255 is the bottom edge. For example, to saturate the video clip red, change the Colour Y value to 255 and use 0 for the Colour X value.

These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content. When you are using the default palette HSVPAL, Colour Y controls saturation, and Colour X controls hue.
FilmicGrain

Overview

The FilmicGrain layer is designed to add animated, film grain style noise to a piece of content.

Workflow

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being ‘piped in’ to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.

1. Add a FilmicGrain layer to your track. For more information about adding layers see creating layers.
2. Arrow from your media layer to the FilmicGrain layer so the compositing order looks something like the following image.

![Video layer piped into a FilmicGrain layer on the timeline.](image)

For more information on arrowing, see compositing layers.

3. Adjust the FilmicGrain layer properties to suit your desired sequencing.

Common layer properties

### Blend mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity. The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colous. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Amount

This property controls the amount grain added to the original content.

Luma blend

This property controls the Luma blend. The luma controls whether the grain is applied to the mid tones only and not applied in bright highlights. In other words, if it is white, there will be no grain whereas if it is very dark content then it will have lots of grain.
Kaleidoscope

The Kaleidoscope effect layer can be used to create Kaleidoscope style patterns when it is composed with another content layer.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is al alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
Video

Default

Blend mode: Alpha

Mapping: direct map 1

Palette

Colour X: 0
Colour Y: 0

Media

Video: d3_test_16-9.png

Speed: 1
Mode: Normal
At end point: Loop
Transition time: 0

Audio

Colour Shift

Keying

Move

Crop

Mappings

New mapping: OK Search:

all New Mappings

direct map 1
feed map 2
feed mapping
mapping 1
mapping 2
parallel
spherical
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Repeat count**

The number of times the content is repeated in the pattern

**Zoom**

Increase value to zoom in on the content

**Source rotation offset**

Increase value to rotate the content offset

**Source rotation speed**

The speed of the rotation of the content

**Rotation offset**

The rotation offset for the kaleidoscope effect

**Rotation speed**

The rotation speed of the kaleidoscope effect
Lut

The Lut layer allow you to take .cube files and use them to alter the colour grade of content.

Overview

The Lut layer is designed to create movie style looks & for technical Luts to support conversion workflows. Lut files are commonly created and exported from image editing programs and have a .cube file extension.

Workflow

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
1. Add your Lut files to the LutFile folder inside your project folder. The LutFile folder is contained within the Objects folder. For more information see placing media files.

2. Add a Lut layer to your track. For more information about adding layers see creating layers.

3. Arrow from your media layer to the Lut layer so the compositing order looks something like the following image.

![Video layer piped into a Lut layer on the timeline.](image)

For more information on arrowing, see compositing layers.

4. Adjust the Lut layer properties to suit your desired sequencing.

Common layer properties

Blend mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effect is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

 Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

 Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

 Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

 This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Lut layer properties

Colour Lut

This property allows you to select a lut file from the project folder. Lut files live in a specified folder inside the Objects folder. For more information see Project Structure.
Chroma intensity

This property controls the chroma intensity. If the Chroma intensity is at 0, then the content is using the original chroma, whereas if it is at 1 its the chroma specified in the lut file. You can alter the max value by right clicking the property which will simply multiply the original Chroma value by the new max value.

Luma intensity

This property controls the Luma intensity. If the Luma intensity is at 0 then the luminosity will be the same as the original content, where as if it is at 1 then it will be the Luma specified in the lut file. You can
alter the max value by right clicking the property which will simply multiple the original Luma value by the new max value.
MotionBlur

The MotionBlur layer causes moving objects in the source content to leave trails behind them.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DyTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCOL yCOL**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL , **Colour Y** controls saturation, and **Colour X** controls hue.
Blur amount

The amplitude of the blur effect. Default value is 0.5. The min/max values are 0 and 1. Decreasing the value from 0.5 to 0 will sharpen the content, whilst increasing the value from 0.5 to 1 will gradually amplify the blurriness of the content.
Noise

The Noise layer creates video noise onto content when it is composed with another content layer.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Mode

- Relative
- Absolute - Noise value is added to the source image

Colour

- Colour - Noise is composed of random colour values
- Greyscale - Noise is composed of random greyscale values

Strength

Adjust the strength of the noise
Pixelate

The Pixelate layer pixelates whatever content is pumped into it.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
The Pixelate layer supports the use of Arrows, allowing content layers to be piped into the Pixelate layer. For more information please see the sub-chapter Composing layers using Arrows.

**Preview Thumbnail**

This property is a small preview window at the top of the editor showing you what the layer output looks like.

**Blend Mode**

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colous. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

xCOL, yCOL

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

Colour Shift properties are:

- **Brightness (shift)**: 0
- **Contrast (scale)**: 1
- **Saturation scale**: 1
- **Hue shift**: 0
- **RGB controlled**: Together
- **Red minimum**: 0
- **Red maximum**: 255
- **Red gamma**: 1
- **Green minimum**: 0
- **Green maximum**: 255
- **Green gamma**: 1
- **Blue minimum**: 0
- **Blue maximum**: 255
- **Blue gamma**: 1
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min, max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

Max
This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)
If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When *gamma* is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): g=0.5, g=1, g=1.5

**Size**

The size of each pixel.

**Mask**

Optional bitmap to mask the pixelate effect.
PixelMap

The PixelMap layer allows for a pixelmap input to be used to create a pixelating effect. The pixel map can be created from an external program, for example Photoshop, allowing you to create and customise your own pixel maps.

The Pixelmap Effect layer is used to create a pixelating effect that is based on the input of a pixelmap. The pixelmap determines the tile atlas for the max value of each pixelated pixel which is fixed to a 16 x 16 grid.
All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the [compositing layers topic](#).

To draw an arrow between two layers, hold down **ALT** and left click & drag between the source and destination layer.

The PixelMap layer supports the use of Arrows, allowing content layers to be piped into the PixelMap layer. For more information please see the sub-chapter [Composing layers using Arrows](#).
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCOL** and **yCol** values (see the section for **xCOL**, **yCol**).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

xColl yCol

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. Colour X controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. Colour Y controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the Colour Y value to 255 and use 0 for the Colour X value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, Colour Y controls saturation, and Colour X controls hue.
Effect

Default value is 1. The min/max values are 0 (no effect) and 1 (maximum effect). Decreasing the value from 1 to 0 will reduce the effect of the Pixel map on the content.
Scale

Default value is 1. The min/max values are 1 and 16. Increasing the value from 1 to 16 will increase the size of the Pixel map by a specific scale factor. For example, a value of two means the PixelMaps width and height will double, a value of three means the Pixel maps width and height will triple etc.

Mode

Normal
Blocky

Pixel map

This points to the still image file that defines the PixelMap. Selecting this property will open the Texture object library, which shows all of the Texture still image files saved on your local hard-drive in the DxTexture folder. However, before placing a Pixel map you will need to create it.

To create a Pixel map:

1. Create a .png image using a program such as Adobe Photoshop. The Pixel map can only be increased by a maximum scale factor of sixteen so if it is going to be applied to the whole screen make sure the Pixel maps resolution is high enough.

2. Make the areas to be see-through in the screen see-through in the Pixel map.

3. Make the areas to be illuminating in the screen white in the Pixel map.

4. Make the areas to be black in the screen black in the Pixel map.

See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software Also save the file to a supported file format.

The Pixel map will be applied to the screen on a pixel by pixel basis.
Scroll

The Scroll layer takes any bitmap and scrolls it in a horizontal direction across the output canvas, repeating it when necessary. It is particularly useful in situations where you want to generate scrolling content and want to modulate the scrolling speed in real time (for instance, so that content moves at the right speed for television cameras). In these situations, the process of making a video containing scrolling content is too slow.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expansions of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCOL** and **yCOL** values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xColl yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Speed

This controls the horizontal scrolling speed, measured in pixels per beat. Positive numbers scroll the image to the left; negative numbers scroll it to the right.

Border

The Scroll layer can generate a black border above and below the image, scaling the image down in the process. The `border` property controls the height of this border, in pixels.
SpinBitmap

The SpinBitmap layer displays a moving, rotating bitmap. It is able to perform continuous rotation, and can tile its image. Since a bitmap contains an alpha channel, this layer can display up to three layers of images, one behind the other.

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the [compositing layers topic](#).

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Preview Thumbnail
This property is a small preview window at the top of the editor showing you what the layer output looks like.

**Blend Mode**

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have \( \text{RGB} = 255 \) (white) and \( \text{alpha} = 0 \) (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at \( 255 \) RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: \( \text{white} \times \text{grey} = 0.5 \). White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colous. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effect is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HISVAPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCOL yCOL**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL , **Colour Y** controls saturation, and **Colour X** controls hue.
Tile

If set to `off`, the central image is not tiled and the `move speed` property has no effect.
Rotation speed

If this is set to 0, the rotation angle is controlled by the angle property. Otherwise, this property controls the rotation speed in degrees per beat.

Scale

This controls the scaling of the bitmap. Somewhat counter-intuitively, increasing the number reduces the size of the bitmap and increases the number of tiled copies.

Move speed

This controls the movement speed of the image, in each of the x and y axes, in pixels per beat.

Angle

If rotation speed is set to 0, this property controls the angle. Otherwise, it has no effect.

Depth

With values ranging from 0 to 2, depth controls the number of layers displayed. This only results in a visual effect if the bitmap is partially transparent, i.e. contains an alpha channel.

Depth blend

This controls the blend mode applied between the multiple layers. The options to choose from, are add, over, alpha or multiply. For more information on blend modes please see the blendMode section in the Common layer properties topic.
Offset

This shifts the entire field left or right, measured in pixels.
Strobe

The Strobe layer creates a flashing strobe effect over multiple screens.

Warning: The Strobe layer only works using a Feed map. In order to turn each screen into "one flash", scale each feed rectangle to 1 pixel and place each screen's feed rectangles next to each other on the Mapping canvas.

Some Generative layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down **ALT** and left click & drag between the source and destination layer.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colour. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Rate

Speed of strobing, in cycles per beat. 0 = stop flashing.

Proportion

The amount of time within the cycle that the strobe is on. 0.5 = half the cycle time is white.
**Attack**

Speed of attack (ramping from 0 to full brightness). 1 = instant, 0 = slowest.

**Decay**

Speed of decay (ramping from full brightness down to 0). 1 = instant, 0 = slowest.

**Cycle**

The cycle of the strobe. 0 = all screens flash together, 1 = screens flash in turn.

**Randomise**

The randomness of the cycle order. Yes = order of the screens is randomised, No = order of the screens is not randomised.

**Min brightness**

The minimum brightness outputted by the layer. 0 = 0%, 1 = 100%.
Trigger

The Trigger layer allows you to trigger still image content at specific points of time on your screen(s).

All Effect layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effect similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCOL yCOL**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Bitmap

Still image file to be placed in the key-frame editor.

Speed

The transition speed of the triggered content. Positive = bottom to top, negative = top to bottom.
Size $x, y$

The pixel size of the image.

Trigger

Every trigger keyframe triggers the chosen image.
Bugs

The Bugs layer is a simple generative layer that simulates a number of bugs.

A bug is a block of colour that moves across the canvas, leaving behind a fading trail of blocks. With each step, the bug may decide to change its direction or continue in the same direction.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol y Col**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Block size

This controls the size of each bug, in pixels. When the block size is higher than 1, a one pixel black border is drawn around each bug's square. The smaller the size, the finer the pattern generated.

Population

This controls the number of bugs. When set to 255, the maximum number of bugs is drawn.
Speed

This controls how many times per beat the bugs update their position.

Turniness

This controls the ability of each bug to change direction at each step. When set to the minimum value 0, the bugs will move in straight lines and never change direction; when set to the maximum value 255, the bugs will turn at every opportunity.

Field

This controls the number of directions the bugs can choose from. The min/max Field values are 0 and 9. Different Field values control the directions the bugs move in in different ways.

The following Field values cause the bugs to move in the following directions:

0 : horizontally right and left + vertically up and down

1 : horizontally right and left

2 : vertically up and down

3 : diagonally up and down

4 : diagonally down into horizontal right and left

5 : vertically up

6 : vertically down

7 : horizontally right

8-9 : horizontally left
Sooner or later, everybody needs chevrons. Recognising this fact of life, the disguise software includes a Chevron layer for all of your chevron needs.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

### Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPal) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol y Col**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
**Speed**

This controls the speed of the chevrons, in pixels per beat. Positive numbers move the chevrons to the left; negative numbers move the chevrons to the right.
Angle
This controls the angle of the chevron fins, in degrees. Smaller numbers move the angle closer to horizontal.

Spacing
This controls the spacing between the chevrons, in pixels.

Width
This controls the width, as a percentage of the spacing, of the first colour band.
Colour

The Colour layer is the easiest layer type; it simply generates a flat colour. Many other layers take over the properties of the Colour layer. For example, the Bitmap layer and the Video layer use colour properties to allow you to tint their output.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (**HSV PAL**) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCol** and **yCol** values (see the section for **xCol**, **yCol**).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

xColl y Col

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. Colour X controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. Colour Y controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the Colour Y value to 255 and use 0 for the Colour X value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, Colour Y controls saturation, and Colour X controls hue.
Generative Layers

The disguise software has several types of layers that can be used to generate content in real-time known as Generative layers.

Types of Generative Layers

- Bugs
- Chevron
- Colour
- Gradient
- Notch
- Radar
- RGBColour
- Scan
- Strobe
- Tennis
- Timecode Readout
Gradient

The Gradient layer draws real-time gradients composed of two colours.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultipies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in a distinctly darker or lighter area, but does not result in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

Exclusion

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

Colour Dodge

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

Hard Mix

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

Brightness

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

\textbf{xCol \ y Col}

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. \textbf{Colour X} controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. \textbf{Colour Y} controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the \textbf{Colour Y} value to 255 and use 0 for the \textbf{Colour X} value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, \textbf{Colour Y} controls saturation, and \textbf{Colour X} controls hue.
Type
Either *linear* or *radial*.

2nd Brightness
This controls the brightness of the second colour in the gradient.
2nd xCol, 2nd yCol

This controls the second colour in the gradient (both colours must be taken from the same palette bitmap).

Position x, y

This controls the position (ranging from 1 to +1 on both axes) of the the gradient. In case you are using the linear gradient, this is the midpoint of the line; If you are using the radial gradient, this is the center of the circle.

Length

Length can have a value ranging from 0 to 1 (1 = complete canvas width) and controls the width of either the linear gradient or the radius of the circle.

Angle

This is only meaningful when using linear gradients. It controls the angle of the line, in degrees.

Inner Radius

Radial offset before the gradient occurs. Only applies when the gradient Type is Radial.
Notch

Notch is a generative content creation platform that integrates with the disguise software.

Overview

The Notch layer allows users to use Notch Blocks exported from Notch Builder.

Notch content creation should be approached from a similar standpoint to that of rendered content, whereby the user specifies as much in advance as possible and test the content on a real world system before show time to reduce the likelihood of performance related issues.

⚠️ **Warning:** Extremely demanding Notch blocks can cause an oversubscription of available memory resources - click here for full advisory.

Some Generative layers take their source from other layer types - either content or generative - by use of an arrow. Linking two layers with an arrow defines the arrowed from layer as the source, and the arrowed to layer as the destination. If you have an arrow between a content layer and effect layer, it is said that the content layer is being 'piped in' to the effect layer. For more information on arrows, see the compositing layers topic.

To draw an arrow between two layers, hold down ALT and left click & drag between the source and destination layer.

On the Notch layer, you can specify which source the layer is using (either texture or an arrowed layer) by using the Video Loader parameter. The Video parameter of the layer will show a thumbnail of either the texture chosen, or the content coming from the arrowed layer (depending on the selection made)
Notch advice

**Please note:** It’s always recommended to combine multiple Notch blocks into one block, as oppose to using lots of small blocks in your project. Notch has functionality to combine blocks in Notch builder.

- Notch renders to the size of the resolution of the mapping being used on the Notch layer within the disguise software.

- Rotation is displayed in degrees within Notch, but shown as radians once exposed within the disguise software.

- The Y and Z axes are different in Notch and the disguise software, and need to be flipped/converted manually using an expression.

- The general consensus is that one should NOT use Universal Crossfade alongside Notch. Using Universal Crossfade adds twice as much load, and depending on the effect used it is unlikely to generate the desired effect.

- Only ever have one dfx file connected to a Kinect.

- When using a Notch block with Kinect input, in order to see this input the machine will need the Kinect SDK installed. You will also need to enable Kinect input via Devices in Notch Builder.

- For audio reactive, its worthwhile to define the audio device being used on the server in you notch projects audio device (device->audio device).

- When using sockpuppet - please advise the Notch content creators to create unified naming conventions for all exposed parameters (ie: FX1, FX2, Speed1, Speed2, Color1, Color2, Color3, etc.) or the end management will be difficult to manage.

- Any block that stores frames (i.e. frame delay) needs to be managed extremely carefully or it may eat up all memory resources. If vram resources are being eaten up inexplicably, it’s worth checking whether the Notch block is storing frames for use anywhere.
Also, as Notch blocks are not user definable in terms of DMX assignment order, it is always best to pre-determine the number of attributes one wishes to have exposed in the Master block.

**Workflows**

The Notch layer is specifically for playing back Notch blocks. Depending on your application and production needs, there are a number of workflows you can employ in order to integrate Notch effects into the disguise software. Below are a few recommended workflows for their respective applications. Bear in mind that these are stripped to the bare minimum elements for simplicity, and users are required to have a valid Notch Builder software with export capabilities in order to follow along.

For more information on Notch Builder, see here.

**IMAG effects**

These are probably the most straight-forward effects to implement on a show, as blocks are usually designed to be plug and play, with a video source and exposed parameters to control and interact with the effect. Below we denote the workflow.

**In Notch**

- Create a Notch effect with a node that accepts a video (usually Video Source)
- Expose the Video Source property and compile the block

**In disguise**

- Create a Notch layer, load the IMAG Notch block into it.
- Create a Video layer (or any layer you wish to output content from, i.e. generative layers).
Set Video Source to **Layer**.

Move the Notch layer above the other layer in the stack, then arrow the video layer to it.

You should now be able to see the IMAG effect applied to any content from the layer below. You can change the effects being applied under the parameter group’s Notch Layer parameter, if the effect is set up to use layers as individual effects.

**Things to note**

- The disguise software does not yet support Video selection without layer arrowing. The Video parameter under Video Sources is mostly unused, though it can still be used for placeholder images. The images displayed there are taken from the DxTexture folder.

- If multiple Notch layers are used and you wish the arrowed video to be the same for all of them, you will need to set up the exposed parameter’s Unique Identifier in Notch to be the same for all exposed video sources.

- If multiple layers are arrowed into an effect that accepts multiple sources, the source layer is chosen in order of selection (i.e. the first layer selected will be the first Video source, the second will be the second, etc) regardless of the order of the parameters themselves within the list.

**3D virtual lighting simulations**

These workflows often involve a 3D mapped object with projectors simulating light sources moving and affecting the object in real time. These effects rely on the virtual 3D space to match the real space and object, along with the coordinates systems of the disguise software and Notch.

**In Notch**

- Add a 3D object node (or a Shape 3D node, etc) and add it to the scene.

- Add a light source.

- Create a UV camera node to output the lit textures onto the object’s UVs.
Expose the appropriate parameters (light positions, object positions, etc).

Compile and export the block.

In disguise

Create a surface with the same object used in Notch.

Calibrate the projectors to the surface with your preferred calibration method.

Create a Notch layer.

Apply the Notch layer via Direct mapping to the object.

Move the lights around to see the object UVs being affected.

Notes

Warning: While it’s often advised to enable Deferred Rendering in Notch, it might negatively impact performance depending on the complexity of the scene. Use this functionality cautiously.

In order for the Notch scene to match the scene within the disguise software, accurate measurements need to be taken on stage and an origin point reference needs to be determined from the start. Setting an origin point early in the process will make the line up process easier.

Lights can be linked to MIDI, OSC or DMX controls like every parameter in the disguise software, or can be keyframed and sequenced on the timeline.

The mapped object’s movement can be linked to automation or tracking systems, and the positional data can then be used to drive the exposed position and rotation parameters.
If multiple objects are in a scene, you will need to create a larger UV layout that accommodates each object in a separate UV area, and then match the overall lightmap resolution by setting the surface resolution within the disguise software. You can use the UV Output section of the 3D objects node in Notch to determine where in UV space a specific mesh will be output within the overall canvas.

Particles systems and tracking regions

A very common application of Notch is to use it alongside tracking systems such as BlackTrax in order to generate particles from specific points in space, to be either projected on a surface or displayed on an LED screen. Below is a broad outline of the workflow using an LED screen, the BlackTrax system, and region camera to specify the tracking regions:

**In Notch**

- Create a particle system (Minimum required: Particle Root, Emitter, Renderer).
- Create a Region Camera.
- Expose the position parameters of the Particle Emitter, as well as the Region Camera’s Top Left X and Y, and Bottom Right X and Y.
- Compile the block and export to disguise.

**In disguise**

- Create an LED screen. This can be placed anywhere in the virtual stage, though it is recommended to place the LED in the correct position to match the physical space.
- Ensure the BlackTrax system is connected and tracking data is being received from the beacons, then select a stringer to use as a tracked point for the Notch particles.
- Create a Notch layer and load the exported block into it.
- Set the Play Mode to Free-run, or press play on the timeline in order for the particles to begin spawning. They are a simulation, and only spawn over a span of time.
Right-click on the BT point being tracked to open a widget that displays the point’s current coordinates in 3D space.

With the Notch layer open and the Particle Emitter position parameters visible, navigate to the tracked point’s widget, hover over one of the position values, then Alt+left click and drag an arrow from there to the corresponding position parameter in the Notch block.

You should now see the particle effect either disappearing off-screen (if the beacon is not in range of the LED screen), or moving towards it.

If the world coordinates of Notch and disguise do not match, and the particle effect is limited to a particular screen or mapped area, a Region Camera can be used to mark the boundaries of the tracking region instead.

To set the region camera, simply measure the XY position of the top-left corner of the LED screen, and do the same for the bottom-right, and enter these values in the exposed Region Camera parameters.

Notes

An often quicker method of finding out the region camera values is to place a tracked BT point on the top left of the screen, and then the bottom right, and manually enter the xyz coordinates displayed on the tracked point’s widget, as they are sure to match precisely.

A common mistake that leads to oddities with tracking regions is when the wrong axes are used. As a rule of thumb, for vertical surfaces you’ll need to take the X and Y position of the top left and bottom right corners, whereas for horizontal surfaces (i.e. for effects built to be displayed on the ground) you will need X and Z. This is obviously also dependant on how the Notch block itself was built and the orientation chosen in the region camera node, so it’s important to double-check these details beforehand.

When using the region camera, particle size plays a fairly important role. It may be advisable to expose emitter size, particle size, and camera distance in order to achieve the desired result.

If multiple machines are outputting the same set of particles but are seeing a different result, it is because both particle roots are running separate instances of the simulation on two machines.
You can fix that by setting the Particle Root node for that emitter to Deterministic by ticking the box in the node editor in Notch Builder.

Notch layer properties

Notch layers are comprised of a set of default properties (detailed here) and additional properties that appear depending on what is actually in the Notch block. For explanation of properties beyond the default, please refer to your Notch content creator.

Effect

The Effect parameter defines which Notch DFX File the layer is looking at.

Blend Mode

Blend Mode controls how the output of the layer is composited with the layers below.

Brightness

This property (which appears as a light bulb icon) controls the brightness of the layer output.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Processing Size**

The resolution at which the Notch effect is processed.

There are two options:

- **Output size** - The resolution of the screen the effect is mapped to (not the mapping itself)
- **Input size** - The resolution of the effect as set in the Notch Builder.

**Dry-Wet Blend**

Global Intensity level for the effect on a scale of 0-255.

**Mode**

This specifies the playback mode.

There are three modes; each one has a specific behaviour that is useful for a different situation.
- **Locked**
  
  If the play cursor continues to play or stops at the end of a section, the video frame number locks to the timeline. When the play cursor holds at the end of a section, the video will play continuously.

- **FreeRun**
  
  If the play cursor continues to play or stops at the end of a section, the video will play continuously. Jumping around the timeline while the cursor is playing or has stopped does not affect which frame is being played.

- **Normal**
  
  When the play cursor stops, the video will also stop and the frame number will lock to the timeline position. When the cursor continues to play or holds at the end of a section, the video will play continuously. Jumping around the timeline while playing does not affect which frame is being played.

- **Paused**
  
  When playback reaches the end of the video clip, the clip will pause on the last frame.
Radar

The Radar layer is one of the simplest generative layers. It generates a moving beam with a bright leading edge that fades to a black trailing edge.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay
Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light
Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light
Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken
Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten
Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference
Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter **Content Mapping**.

**Palette**

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSV PAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the **DxTexture** folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the **xCol** and **yCol** values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

xColl y Col

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. Colour X controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. Colour Y controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the Colour Y value to 255 and use 0 for the Colour X value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, Colour Y controls saturation, and Colour X controls hue.
Speed

This controls the speed of the beam movement. Positive numbers go rightwards/downwards; negative numbers go leftwards/upwards.
Beam Size

This controls the size of the beam. When set to 255, the width of the beam is the same as the width of the canvas.

Direction

This is an option property; the two options are **horizontal** and **vertical**.

Timing

This controls how the **speed** property is interpreted. The options are:

- **Per sec**: the radar beam moves at \( x \) pixels per second, regardless of how the play cursor moves. This means that the radar beam position is not predictable anywhere on the timeline.

- **Per beat**: the radar beam moves at a speed related to the current bpm, so that it moves an integer number of complete sweeps per beat. Furthermore, the position of the beam depends on the timeline position, so the beam is guaranteed to be at the same position at the start of every beat.

- **Per frame**: the radar beam moves \( x \) pixels every frame.
**RGBColour**

The RGBColour layer is one of the most basic layers. It behaves in a very similar way to the Colour layer. Quite simply, it allows you to manually set RGBColour values to display a solid colour.

**Preview Thumbnail**

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter _Content Mapping._

**Red**

Controls the Red additive colour mix.

**Green**

Controls the Green additive colour mix.

**Blue**

Controls the Blue additive colour mix.
Scan

The Scan layer is a simple test pattern generator that is useful for checking output signal timings. It generates a single-pixel wide horizontal red line and a single-pixel wide vertical blue line. Each line moves forward by a single pixel per frame.

It sometimes happens that downstream DVI processing devices such as switchers, scalers or LED processors skew the timing of the DVI signal, causing the appearance of jerky playback. At such times it is useful to be able to generate a visual signal that is visibly smooth on the computer, but clearly shows signal skews on the output devices.

The Scan layer is also useful for checking if the 3D screen models have been correctly imported into the disguise software. If the model has incorrect texture coordinates, you will see a single row or column either disappearing or spanning two separate sub-objects.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. The result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
**Overlay**

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

**Hard Light**

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

**Soft Light**

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

**Darken**

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

**Lighten**

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

**Difference**

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

**Step**

This controls the movement speed of the lines. The number you fill in is the number of frames the lines moves forward per pixel. Increasing this number slows the line down. Setting `step` to 0 allows you to control the position of the lines manually using the `x` and `y` properties.

**x, y**

When `step` is set to 0, you can directly control the position of the lines by adjusting `x` and `y`. The position of the vertical blue line is controlled by `x`, while the position of the horizontal red line is controlled by `y`. Both are measured in pixels.

**Draw**

There are two options:

- **Lines**: this shows red and blue lines which intersect to form a crosshair (useful for LED screens).
- **Dot**: this shows a single white dot at position `x, y`. This option is more useful when calibrating singlepixel strings (useful for linear fixtures).

**Timing**

There are two options:

- **Pixels**: if set to this, the crosshair moves one pixel every (step) frames.
- **Time**: if set to this, the crosshair moves at (1/step) pixels per second.
Tennis

Tennis has been a staple of the The Disguise software since the early days, hidden until r17 and previously called Pong but renamed due to legal reasons. Best played using a midi controller.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiply out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity. The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base cols. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol y Col**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Options

Ball speed

This controls the speed of the generated ball, in pixels per beat.
Bat size
This controls the size of each bat.

Bat padding
This controls padding on each bat.

Control

Please note: We recommend using an external controller (such as a midi device with faders) to control the bats.

Left bat position
This controls the vertical location of the left bat.

Right bat position
This controls the vertical location of the right bat.
Readout

This layer allows you to assign the incoming timecode, remaining track time, section time or system time to a screen via a mapping to show on the output as well as in the stage.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.

Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
**Mask**

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

**Multiply-Fade**

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

**Multiply-Alpha**

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

**Premultiply Alpha**

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

**Colour Burn**

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

**Screen**

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were screened. This is useful for adding highlights to an image. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as it if were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click **palette** to open the Texture object library.
2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xColy Col**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Display

The Display property allows you to switch between readouts for the incoming timecode, MTC layer, Timeline, Section, Video timecode, Time and Output Debuggers, and System time readouts.
AnimateCameraControl

This layer allows you to script a fly-through of the stage so that you can create presentation videos, with or without audio.

Please note: while the Timeline play cursor is under the layers extent, the AnimateCameraControl layer overrides normal Stage navigation. To regain control of the normal Stage navigation, move the play cursor away from under the layer (or move the layer).

Please note: AnimateCameraControl layer gives you full control of keyframes but is also more complex and harder to operate than the AnimateCameraPreset layer, which is based on animating between a number of pre-made configurations.
Camera pivot

This controls the position of the **pivot** the point where the camera is looking at. This is a 3 dimensional position, i.e. it has x, y and z components. These are all stage space coordinates, expressed in meters.

**Please note:** in most cases the best approach is to have a static pivot point throughout the camera fly-through animation. Set the pivot point to the point of rotation, then animate the other properties. If not, you may get a sliding effect of the camera.

Camera rotation

This controls the rotation of the camera around the pivot point, in degrees around each of the x, y and z axes. Rotation around the x axis is equivalent to elevation, around the y axis is equivalent to heading, and the z axis rotates the camera around its viewing axis.

Distance from pivot

This controls the distance (in meters) from the camera to its pivot point. Increasing this number takes the camera away from the pivot point; decreasing it moves the camera closer to the pivot point.

View angle

A higher view angle will show more of the stage from a given point of view. View angle is measured in degrees.

Camera locked

This manipulates how the layer controls the camera.
Left-click **camera locked** to open the **camera locked** keyframe editor.

Left-click **always** to open the option properties.

Left-click **when playing** to set the layer so that it only controls the camera when the keyframe editor is playing, allowing you to move freely when the keyframe editor stops.

Left-click **always** to set the layer so that it always control the camera.
AnimateCameraPreset

The AnimateCameraPreset layer lets you create a camera fly through using a number of configurations which can be dropped onto the timeline. The disguise software will then animate the camera between the configurations.

Creating a camera position animation

1. Left-click **position**. This will open the **position** keyframe editor and a window where you can create a new camera position.

2. Left-click **new cameraposition** to highlight the text field.

3. Type in the name of your new camera, for example **east**.

4. Hit **Enter**. This will create the new camera position, add it to the **position** key-frame editor at the position of the currently selected Track bar, and open the Camera Position editor. The camera position key-frames can also be click-dragged from the Camerapositions manager to a specified point on the Position key-frame editor.
Process used to create a camera position, in this example position 1, and place it on the Position key-frame editor

5. Edit the values within the Camera Position editor (properties explained below).

6. Repeat the steps listed above to create as many new camera positions as required.

7. Hit Shift and < to jump to the first second of the track.

8. Hit Enter to trigger normal play mode and watch the camera animation.

Position

This property lets you create a new configuration of cameras you want to animate. Please see the section below Creating a camera position animation for step-by-step instructions on how to do this.

Camera locked

This manipulates how the layer controls the camera.

— Left-click camera locked to open the camera locked keyframe editor.

— Left-click always to open the option properties.

— Left-click when playing to set the layer so that it only controls the camera when the keyframe editor is playing, allowing you to move freely when the keyframe editor stops.

— Left-click always to set the layer so that it always control the camera.

As demonstrated above, you need to use the Camera Position editor as part of the process to animate a camera position.

To open the Camera Position editor:

— Right-click a camera position in the Camerapositions manager
Pivot

This controls the position of the pivot the point where the camera is looking at. This is a 3 dimensional position, i.e. it has components x, y and z; all are stage space coordinates, expressed in meters.

**Please note:** in most cases the best approach is to have a static pivot point throughout the camera fly-through animation. Set the pivot point to the point of rotation, then animate the other properties. If not, you may get a sliding effect of the camera.

Elevation

This rotates the camera up and down and is expressed in degrees. Note that the Elevation value is based around the cameras pivot point.

Heading

This rotates the camera left and right, in degrees. Note that the Heading value is based around the cameras pivot point.

Roll

This rotates the camera around its own z-axis, in degrees.
Distance to pivot

This moves the camera closer to or further away from its pivot point, in meters.

View angle

A higher view angle will show more of the stage from a given point of view. View angle is measured in degrees.

Previsualisation Layers Overview

Delete this text and replace it with your own content.
AnimateObjectPreset

The AnimateObjectPreset layer allows you to simulate moving screen assemblies within the Stage Visualiser by placing AnimateObjectPreset key-frames onto the key-frame editor. The disguise software will then animate the object(s) between the configuration key-frame positions.

Please note: at this time, disguise does not output control instructions to Screen movement systems. The AnimateObjectPreset layer is for pre-visualisation purposes only.

1. Create an AnimateObjectPreset layer
2. Create ObjectPosition configurations
3. Add screens or objects to the configurations
4. Configure the properties (position etc) of the configurations

5. Playback the configuration keyframes on the timeline to animate objects around the stage.

Creating a AnimateObjectPreset animation

1. Create an AnimateObjectPreset layer and assign a mapping type that contains all of the screen or objects that you want to control. For information on mapping types please see the chapter Content Mapping.

2. Open the Config key-frame editor by left-clicking config and create and place a Configuration key-frame. The process used to create/sequence Screen Configuration key-frames is the same as any other layer type that uses key-frame editors, for example the AnimateCameraPreset layer. For an example on how to create and animate key-frames using the key-frame editor please follow the step-by-step instructions in the AnimateCameraPreset topic.

3. Use the Configuration editor to edit the position of the screen or object. Please read the Screen Configuration properties section for more information on how to do this.

4. Create as many Configuration key-frames as required and place these on the Config key-frame editor. Press the Play button and the disguise software will animate the screen or objects between the Configuration key-frame positions.

Axes

There are five options to choose from:

All: this affects the properties position, rotation and animate.
**Position only:** this affects only the `position` property.

**Rotation only:** this affects only the `rotation` property.

**Animate only:** this affects only the `animation` property of the screen type **Moving mesh** which is currently not in an official disguise release.

**Position and rotation:** this affects only the `position` and `rotation` properties.

**Config**

This property lets you create a new configuration of screens or objects that you want to animate.

**Master**

The `master` property allows you to exert fine control of the movement speed between configuration positions. By default, the disguise software generates a linear animation between configuration points. However, if there are any keyframes on the master timeline between the configuration endpoints, the disguise software will use the master curve to animate between the endpoints.

**Actions**

**Please note:** when using the AnimateObjectPreset layer import properties make sure your layer names and object names have no spaces or characters such as dashes or slashes etc otherwise the layers and screens will not import. In addition, the export and import properties wont work if there are no MTC references on the track. Therefore put a note on the track at the start for example saying MTC:0.0.0.0. These issues will be fixed in a future release.

There are six options to choose from:

**Export:** makes a list of time based positions of screens when they change. This list is saved as a `.txt` file called `screenpos_yourlayername`.

**Export all in track:** Export all points from the track.
**Import**: Imports all previously exported points.

**Import all in track**: Imports all points for a specific track.

**Export cues**: makes a time list of when specific configuration names are key-framed. This list is saved as a .txt file called `screencues_yourlayername`.

**Export configs**: makes a list of configurations. This list is saved as a .txt file called `screenconfigs_yourlayername`.

The .txt files are saved in a new folder called **table** inside the specific Project folder. For more information on where the specific Project folder is located please see the sub-chapter **Projects location**. You can open the .txt file with Microsoft Excel or Open Office.

You need to use the Configuration editor as part of the process to animate a screen or object.

**To open the Configuration editor:**

- Right-click a configuration from the ObjectConfigurations manager

**Add / Remove Screens or objects**

1. To add a screen to the Screen Configuration editor left-click the + button. This will open the Screens manager.
2. Left-click the object you want to add, in this example Surface 1. This will add Surface 1 to the Screen Configuration editor.

3. Edit the Screen Configuration editor's position or rotation properties.

4. To remove a screen from the Configuration editor left-click the button and select the corresponding screen from the Screens manager.

**Position**

This controls the screen position in the 3D space. There are three numbers laid out horizontally: respectively the x (left/right), y (up/down) and z (into/out of the screen) coordinates.

**Rotation**

This controls the rotation of the screen mesh, in degrees. The x component controls the rotation around the x axis; the y component around the y axis; and the z component around the z axis.
TargetPreset

The TargetPreset layer is similar to the TargetControl in that it allows to control the movement of moving head lights. Each light in the target screen (which must be of the screen type DmxLights) has its own target point.

⚠️ Warning: the fixture defined in the DMXLight needs to set to aimable. If not, the TargetPreset layer will have no effect.

The TargetPreset (and TargetControl) layer allows you to control how these target points are generated from the lights base positions. The main difference is that while the TargetControl layer allows to keyframe every frame of the offset, scale and rotation values, the TargetPreset layer is based on setting a number of configurations of the target offset, scale and rotation and then interpolate the movements between the configurations. Notice that the fixture defined in the DMXLight needs to set to **aimable**. If not, the Target layer will have no effect.

Please read the sub-chapter DmxLights for information on how to create a DmxLights screen. Afterwards please read the sub-chapter Creating a fixture for information on how to add a fixture to a DmxLights screen.
Targeting/animating fixtures

1. Create a TargetPreset layer.

2. Target the fixtures by assigning a mapping type with the DMXLights you want to control. Make sure to set the fixture defined in the DMXLight to **aimable**.

3. Open the TargetPreset keyframe editor by left-clicking **target** and create and drag/drop a Target keyframe (configuration). The process used to create and sequence Target keyframes is the same as any other layer type that uses key-frame editors, for example the **AnimateCameraPreset** layer.

4. Right-click the the Target keyframe to edit the position of the fixture targets see below for more info.

5. Create as many Target keyframes as required and place these on the keyframe editor.
6. Press the Play button. The disguise software will now animate the light's movements between the Target positions.

**Mapping**

Determines what mapping of DMXLight screens to use. Notice that any mapping type can be created using DMXLight screens.

**Target**

This property lets you create a new configuration of fixture targets, each specifying a target position (offset, rotation, scale) set which can be animated. Please see the section below for step-by-step instructions on how to do this.

You need to use the Target editor as part of the process to animate movinghead fixtures.

**Opening the Fixture Targets editor**

- Right-click a target in the Targets manager.

<table>
<thead>
<tr>
<th></th>
<th>fan down</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Offset</strong></td>
<td>0, -25, -28</td>
</tr>
<tr>
<td><strong>Rotation</strong></td>
<td>0, 0, 0</td>
</tr>
<tr>
<td><strong>Scale</strong></td>
<td>6, 1, 1</td>
</tr>
</tbody>
</table>

**Offset**
This controls the **offset** of the fixture target, in meters, from the base light position.

**Rotation**

This controls the **rotation** of the fixture target, in degrees. The x component controls the rotation around the x axis; the y component around the y axis; and the z component around the z axis.

**Scale**

This controls the **scale** of the fixture target. 1 = no change, >1 = scale up. If all are set to 0, the lights will focus on a point; if two are set to zero, the lights will focus on a line.
TargetObject

The TargetObject layer is similar to the TargetControl in that it allows to control the movement of moving head lights. The only difference being that instead of using virtual target points the TargetObject allows you to use an object as the target.

⚠️ Warning: the fixture defined in the DMXLight needs to set to aimable. If not, the TargetObject layer will have no effect.

Please read the sub chapter DmxLights for information on how to create a DmxLights screen. Afterwards please read the sub chapter Creating a fixture for information on how to add a fixture to a DmxLights screen.

For step-by-step instructions on how to control moving head lights using the Target layer please see the sub chapter Targeting fixtures.

Mapping
Determines what mapping of DMXLight screens to use. Notice that any mapping type can be created using DMXLight screens.

Target

This property allows you to select the object that you want to use as the target.

Offset

This value is added to the (x, y, z) stage position of each fixture. To move the fixture targets left and right, change the x property, to move them up and down, change the y property and to move them in and out, use the z property.

Scale

This scales the target points in each of the three axes x, y and z. Values higher than 1 increase the size of the target grid, causing the beams to splay outwards; values lower than 1 reduce the size of the target grid and setting the scale to 0 causes all beams to converge.

Rotation

This rotates the target grid around each of the x, y and z axes (angles specified in degrees).
TargetControl

The TargetControl layer is similar to the TargetPreset in that it allows to control the movement of moving head lights. Each light in the target screen (which must be of the screen type DmxLights) has its own target point.

Warning: the fixture defined in the DMXLight needs to set to aimable. If not, the TargetControl layer will have no effect.

The TargetControl layer allows you to control how these target points are generated from the lights base positions. The main difference is that while the TargetPreset layer is based on setting a number of configurations of the target offset, scale and rotation and then interpolate the movements between the configurations, the TargetControl layer allows to keyframe every frame of the offset, scale and rotation values. This gives the user more precise control of the movements but may be more difficult to use for an inexperienced user. Notice that the fixture defined in the DMXLight needs to set to aimable. If not, the TargetControl layer will have no effect.

Please read the sub-chapter DmxLights for information on how to create a DmxLights screen. Afterwards please read the sub-chapter Creating a fixture for information on how to add a fixture to a DmxLights screen.
Offset

This value is added to the (x, y, z) stage position of each fixture. To move the fixture targets left and right, change the x property, to move them up and down, change the y property and to move them in and out, use the z property.

Scale

This scales the target points in each of the three axes x, y and z. Values higher than 1 increase the size of the target grid, causing the beams to splay outwards; values lower than 1 reduce the size of the target grid and setting the scale to 0 causes all beams to converge.

Rotation

This rotates the target grid around each of the x, y and z axes (angles specified in degrees).
For step-by-step instructions on how to control moving head lights using the Target layer please see the sub-chapter Targeting fixtures.
Bitmap

Using the Bitmap layer is the easiest way to display still images. The Bitmap layer takes over certain properties from the Colour layer and the Video layer. To see where to place media files in Windows see the Placing media files for a project page. To see what video file formats the disguise software support see the Supported file formats page.

Please note: In release r13 onwards it is recommended to use a video layer for still images. For more information see the Placing media files for a project page.

The Bitmap layer supports the use of Arrows, allowing content layers to be piped into the Bitmap layer. For more information please see the sub-chapter Composing layers using Arrows.
Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are expanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask

Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade

The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha

Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha

Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn

Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.

The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen

This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colours. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding highlights to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

Mapping

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the xCol and yCol values (see the section for xCol, yCol).

To change the current palette bitmap:

1. Left-click palette to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

**Colour Shift properties are:**
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min,max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When \textit{gamma} is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): \(g=0.5\), \(g=1\), \(g=1.5\)

**Transition**

Determines the crossfade (in seconds) between two adjacent bitmaps placed in the keyframe editor of the Bitmap layer.

**Threshold**

The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.
Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key colour rgb

The value of the key colour, 0-255, causing the key colour to become transparent.

Scale mode

Provides mapping options to apply to content of a certain resolution which does not match the resolution of the targeted canvas.

Scale

This controls the size of the image. When set to 1, the image is displayed at its normal size. Increasing the scale increases the size of the image; to display at double size, set scale to 2. To halve the image size, set scale to 0.5.

Aspect

This controls the aspect ratio of the image. When set to 1, the image is displayed at a 1:1 aspect ratio. Increasing the aspect value increases the height of the image in relation to its width.

Position

This controls the horizontal (x) and vertical (y) position of the image. When both are set to 0, the centre of the image is centered in the output canvas. Setting x to 1 moves the center to the left edge of the canvas; setting it to +1 moves the center to the right edge of the canvas. Setting y to 1 moves the center
to the top edge of the canvas; setting it to +1 moves the center to the bottom edge of the canvas.

**Rotation**

This controls the rotation of the bitmap, in degrees.

**Crop**

The **crop** pulldown tab contains a set of properties (top, left, bottom, right) that allow you to remove pixels from the edges of the bitmap. Each property specifies the number of pixels you want to remove from the corresponding edge. For example, setting top to 10 will clip 10 pixels off the top edge.
Legacy video

The Legacy video layer is used to play pre-rendered Quicktime video files. To see what video file formats the disguise software support see the Supported file formats page. To see where to place media files in Windows see the Placing media files for a project page.

Preview Thumbnail

This property is a small preview window at the top of the editor showing you what the layer output looks like.
Blend Mode

BlendMode controls how the output of the layer is composited with the layers below. Layers are rendered in a bottom-up order: layers at the top can modify the output of the layers below.
Changing brightness of a content layer in the disguise software is actually controlling the value of the alpha of the layer. Even when displaying a HAP video, the software will composite the layer with a controllable layer of alpha - one per layer.
There are two ways of expressing alpha in an image:

Straight alpha is an alpha channel which functions just like RGB. Alpha acts as a fourth channel of information per pixel which is as independent of the other three as R, G and B are of each other. For example, with straight alpha it is possible to have RGB = 255 (white) and alpha = 0 (fully transparent) on the same pixel. Internally generated content, such as gradient layer, are generated with straight alpha. This is the preferable style of alpha and considering the cleaner method of the two.

Premultiplied alpha takes the alpha and applies it to the values of the RGB channels per pixel. The more transparent something gets with premultiplied alpha, the closer it gets to black - as if the content is sitting on a black table. Premultiplied is the default output of Adobe Photoshop or AfterEffects. the result is our pixel at 255 RGB (full white) with 0 alpha would be calculated as a black pixel in the final image.

Here are explanations of what each individual blend mode will do.

**Over**

Makes a layer fully opaque. Premultiplies all alpha onto the RGB value of each pixel. Alpha = black, so adjusting the brightness of a layer in Over blend mode makes it darker.

**Alpha**

Default blend mode. It will apply alpha values as a transparency if the alpha is present. Brightness changes will make the layer more or less transparent.

**Add**

Adds the value of each RGB pixel together. Always creates a brighter result. Values clamp at 255.

**Multiply**

Reads the level of each subpixel as a level between 0.0 and 1.0, and multiplies source with blend. The result is always a darker image overall. For example: white x grey = 0.5. White turns transparent, black takes precedence. Alpha is applied in the same fashion as the Over Blend mode.
Mask
Choose mask when you want to multiple out content upwards in the stack of layers, rather than downwards. In order to make this work you have to use the same mapping on the layer set to mask and the layer on top.

Multiply-Fade
The same as Multiply, but will make use of the alpha channel to calculate transparency in the source and blend layers. Since maximum transparency is premultiplied, alpha results in black. This will ignore black created through premultiplying.

Multiply-Alpha
Multiply with straight alpha. This mode assumes that the alpha has not been premultiplied onto the RGB values and will not apply a correction to the semi-transparent pixels.

Premultiply Alpha
Like how Multiply-fade will ignore the darkening caused by premultiplying, this blend mode will do the same with Alpha.

Colour Burn
Blend mode increases the contrast to darken the base colour while reflecting the blend colour. The darker the blend colour, the more intensely the colour will be applied in the base image. White as the blend colour produces no change. Using the colour burn blend mode can produce some harsh results at full opacity.
The colour burn blend mode can be used to make tonal and colour adjustments to a layer.

Screen
This blend mode looks at each channels colour information and multiplies the inverse of the blend and base colour. The result is always a lighter colour. Screening with black leaves the colour unchanged. Screening with white produces white. The effect is similar to projecting multiple images on top of each other - where bright white is fully opaque, black is fully transparent and 50% grey is 50% transparent.
Overlay

Multiplies or screens the colours, depending on the base colour. Patterns or colours overlay the existing pixels while preserving the highlights and shadows of the base colour. The base colour is not replaced, but mixed with the blend colour to reflect the lightness or darkness of the original colour.

Hard Light

Multiplies or screens the colours, depending on the blend colour. The effects is similar to shining a harsh spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were multiplied. This is useful for adding shadows to an image. Painting with pure black or white results in pure black or white.

Soft Light

Darkens or lightens the colours, depending on the blend colour. The effect is similar to shining a diffused spotlight on the image. If the blend colour (light source/top layer) is lighter than 50% grey, the image is lightened as if it were dodged. If the blend colour is darker than 50% grey, the image is darkened as if it were burned in. Painting with pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white.

Darken

Looks as the colour information in each channel and selects the base or blend colour - whichever is darker - as the result colour. Pixels lighter than the blend colour are replaced and pixels darker than the blend colour do not change.

Lighten

Looks at the colour information in each channel and selects the base or blend colour - whichever is lighter - as the result colour. Pixels darker than the blend colour are replaced and pixels lighter than the blend colour do not change.

Difference

Looks at the colour information in each channel and subtracts either the blend colour from the base colour or the base colour from the blend colour depending on which has the greater brightness value.
Blending with white inverts the base colour values and blending with black produces no change.

**Exclusion**

Creates an effects similar to but lower in contrast than the difference mode. Blending with white inverts the base colour values. Blending with black produces no change.

**Colour Dodge**

Looks at the colour information in each channel and brightens the base colour to reflect the blend colour by decreasing contrast between the two. Blending with black produces no change.

**Hard Mix**

Adds the RGB channels of the blend colour to the RGB values of the base colour. If the resulting sum for a channel is 255 or greater, it receives a value of 255; if it is less than 255 it receives a value of 0. Therefore all blended pixels have RGB channels of either 0 or 255. This changes all pixels to primary additive colours (RGB), white or black.

**Brightness**

This property (which appears as a light bulb icon) controls the brightness of the layer output.
If the layers blend mode is set to Alpha, then reducing the brightness to 0 also reduces the opacity of the layer to 0. This can be useful when you want to dissolve from one layer to the next. In that case, you can place the new layer above the old layer and increase its brightness level.

**Mapping**

The mapping property controls how the layer output is mapped onto the screen(s) in the Stage level.
For information on mapping, including how to use the different mapping types offered by the disguise software, please see the chapter Content Mapping.

Palette

This points to the still image file that defines the bitmap from which the tint colour is taken. The default palette (HSVPAL) consists of the complete range of hues and saturations. Selecting this property will open the Texture object library, which shows all of the still image files saved on your local hard-drive in the DxTexture folder.

To control the location within the current palette bitmap, and thus control the colour, you have to edit the \textit{xCol} and \textit{yCol} values (see the section for \textit{xCol}, \textit{yCol}).

To change the current palette bitmap:

1. Left-click \textbf{palette} to open the Texture object library.

2. Left-click the still image file you want to use for the palette bitmap.

If you want to use a palette bitmap other than the standard still images provided in the disguise software, you will need to use a custom still image file.
See the Placing media files for a project sub-chapter to understand where to place a custom still image file and how to access it in the disguise software. Also save the file to a supported file format.

**xCol, yCol**

These properties control the coordinates within the current palette bitmap where the output colour is sampled from. The default value is 0,0 which points at the color white (if you are using the HSVPAL palette). Multiplying white with the colors of the chosen content simply generates the original content colors. **Colour X** controls the horizontal position, where 0 is the leftmost edge and 255 is the rightmost edge. **Colour Y** controls the vertical position, where 0 is the top edge and 255 is the bottom edge.

For example, to saturate the video clip red, change the **Colour Y** value to 255 and use 0 for the **Colour X** value. These coordinates refer to the color red in the palette which is being multiplied with the colors of the existing content.

When you are using the default palette HSVPAL, **Colour Y** controls saturation, and **Colour X** controls hue.
Colour Shift

The colour shift property allows you to perform a number of colour-correction operations on the output of a layer.

Colour Shift objects can also be attached to individual video files within the Video layer.
Colour Shift property performs colour-corrections on the output of a visual layer

**Colour Shift properties are:**

- **Brightness (shift)**: 0
- **Contrast (scale)**: 1
- **Saturation scale**: 1
- **Hue shift**: 0
- **RGB controlled**: Together
- **Red minimum**: 0
- **Red maximum**: 255
- **Red gamma**: 1
- **Green minimum**: 0
- **Green maximum**: 255
- **Green gamma**: 1
- **Blue minimum**: 0
- **Blue maximum**: 255
- **Blue gamma**: 1
Saturation scale

When set to 1, no change is applied to the image. When set to 0, the image is desaturated to black and white. When set higher than 1, the image is hypersaturated.

![Saturation scale images](image)

Saturation scale (from left to right): s=0, s=0.5, s=1, s=2

Hue shift

**RGB controlled**

This property controls how the min/max levels and gamma corrections are applied. If set to *together*, then the min, max and gamma values are the same for the red, green and blue channels. If set to *separate*, the min,max and gamma values can be controlled separately for red, green and blue. The latter setting allows you to apply fine colour balance controls to the image.

**Min**

This sets the lowest brightness level found in the image. All pixels at this level are scaled down to zero brightness. Increasing this value enforces shadows and dark levels in the image and can be used to reduce low-level compression artifacts in an image or video frame.

Increasing *min* equally across red, green and blue enforces shadows, whereas increasing *min* on an individual channel has the effect of shifting the colour balance away from that colour. For example, increasing *red min* shifts the image towards cyan.
min = 00 (left), min = 96 (right)

red.min = 0 (left), red.min = 96 (right)

**Max**

This sets the highest brightness level found in the image. All pixels at this level are scaled up to the maximum level, i.e. 255. Decreasing this value brightens any highlights in the image. This is useful when the source image is too dark.

max = 255 (left), max = 260 (right)

If you set min to 255 and max to 0, the image (or single channel) will be inverted.
Invert all (left), invert red (right)

**Gamma**

When \textit{gamma} is set to 1, no change is made to the image. Reducing gamma brightens highlights while increasing gamma darkens lowlights. The maximum and minimum brightness levels remain the same.

Gamma scale (from left to right): \(g=0.5\), \(g=1\), \(g=1.5\)

**Media**

The Video property points to the still media file that defines the content to be used on the layer. Selecting this property will open the VideoClip object library, which shows all of the still image files saved on your local hard-drive in the \texttt{VideoFile} folder.

To change the current piece of content:
1. Left-click **Video** to open the VideoClip object library.

2. Left-click the media file you want to use.

See the Placing media files for a project sub-chapter to understand where to place your content and how to access it in the disguise software. Also save the file to a supported file format.

**Speed**

Selects playback speed. When this is set to 1, the video is played at its normal framerate; when set to 2, the video is played at double speed, and so on.

**Please note:** Playback speed is set at the point where the video keyframe is encountered in the video keyframe editor. Once this is set, the playback speed does not change, even if it varies over time. To change the playback speed, drop a speed keyframe at the point where the video thumbnail appears in the video keyframe editor.

**Mode**

This specifies the playback mode. There are three modes; each one has a specific behaviour that is useful for a different situation.

**Normal**

When the play cursor stops, the video will also stop and the frame number will lock to the timeline position. When the cursor continues to play or holds at the end of a section, the video will play continuously. Jumping around the timeline while playing does not affect which frame is being played.

**Locked**

If the play cursor continues to play or stops at the end of a section, the video frame number locks to the timeline. When the play cursor holds at the end of a section, the video will play continuously.
Freerun
If the play cursor continues to play or stops at the end of a section, the video will play continuously. Jumping around the timeline while the cursor is playing or has stopped does not affect which frame is being played.

At end point
This specifies what happens when playback reaches the end of the video clip. There are three options to choose from:

Loop
When playback reaches the end of the video clip, the clip will loop immediately back to the beginning and will start playing again from there.

Ping-pong
When playback reaches the end of the video clip, the clip will play in reverse back to the beginning.

Pause
When playback reaches the end of the video clip, the clip will pause on the last frame.

Transition time
The transition property specifies the dissolve time, in beats, when switching from one video clip to the next on the timeline. By default, this is set to 0, so the player cuts immediately from one clip to the next. Setting this value to 1 causes the player to dissolve from one clip to the next over 1 beat.

Volume
Some video clips may contain audio. You can fill in any number between 0 (mute) and 255 (full) in the text field to set a constant volume, or open the property's key-frame editor to fade volume in or out.
Output

If the video has an audio track, this controls which audio output device the audio is played to.

Threshold

The threshold of the keying. Colours closer than (threshold) to the key colour become transparent.

Hardness

The hardness of the keying. Lower numbers produce softer edges and higher numbers produce harder edges between transparent and opaque areas.

Key color RGB

The value of the key colour, 0 255, causing the key colour to become transparent.

Scale

Controls the size of the video

Aspect

Controls the aspect ratio of the video, higher numbers make the frame taller.

Pos x y

Position of the video; x = horizontal position, y = vertical position. 0 = centre of canvas, -1 = top edge of canvas, +1 = bottom edge
Rotation
Rotation (in degrees) of the video.

Crop left, right, top, bottom
Crops the frame from the respective edge. 0 = no crop, 1 = entire frame crop.

Vignette
Adds a vignette effect by softening the edges of the crop.
Keyframing overview

Keyframing is used to animate a layer's property over time.

What is the Keyframe editor?

Keyframing enables you to animate a layer's property over time. A layer's property can be accessed from its layer editor. If a layer's property cannot be animated, you may be able to use the Open layer.

Before reading this sub-chapter it is recommended that you read the sub-chapter Editing layers which explains the Layer editor.

Opening a Keyframe editor

To open a Keyframe editor:

- Left-click a Layer property to open the Keyframe editor. Opening a Keyframe editor will close any other open Keyframe editors.

Keyframe toggles

- Beginning with r20, Keyframe toggles have been added, allowing you to reset all keyframes for an attribute within a layer. Enable and disable keyframes easily by clicking on the keyframe toggle icon. For more information on Keyframe Toggles, please visit this link.
Opening multiple Keyframe editors

Hold Ctrl whilst left-clicking the properties of a layer editor to open multiple Keyframe editors.

Keyframe editor types

There are three different types of Keyframe editors. Examples of these three types are described below using the example of a Bitmap layer and its editor, and in the three following sub-chapters. Please read the Editing objects sub-chapter for more information on object property types.
**Numeric**

A numeric Keyframe editor allows you to animate a numeric value when editing a numeric property, in this example `brightness`.

**Option**

An option Keyframe editor gives you a fixed option list to choose from when animating an option property, in this example `blendMode`.

**Object**

An object Keyframe editor refers to another object when animating an object property.

**Navigating Keyframe editors**

1. Left-click a Track bar in the Timeline to play forwards or backwards in the currently selected Track. Notice that as you navigate the Timeline the currently open Keyframe editor will adjust accordingly. The Keyframe editor shows a bar of time just before the current Timeline position and a certain number of bars after it. The number of bars depends on the screen width and the current zoom factor.

2. Left-click the Keyframe editor and hold Alt whilst moving the mouse scroll wheel to zoom in or out. The mouse must remain focused on the Keyframe editor to successfully zoom.

**Common Keyframe properties**

All three Keyframe editor types have the following things in common:

- They consist of a sequence of **Keyframe** or **key values**, which can be either **option** choices, **numeric** values or **objects**. In the case of numeric properties, the value between the Keyframe is generated according to a line or curve that passes through the Keyframe.
A beat of time appears as a single vertical line. Each group of four lines represents a single Track bar and is marked with a bar number just below it. The background colour of the Keyframe editor matches the colour of the corresponding Track bar in the Timeline.

Just above the Keyframe editor is a small triangle with a timecode just above it. This is the 'Now' pointer; it indicates the Timeline position that is currently being rendered to the stage. Editing a Keyframe moves the now pointer to the time of that Keyframe, thus allowing you to instantly see the effect of changing the value.

Keyframe Options

Right click on the border of a keyframe timeline and the Options window will appear with the following options:

- **Delete keyframes after**
- **Delete keyframes before**
- **Normal**: When there are looping keyframes after a section break and the playhead hits that section break, it will play the keyframes after the section break anyway.
**Locked:** When there are keyframes after a section break and the playhead hits that section break, it respects the section break and will not play the keyframes.

**Fit to Contents:** Shortens the layer length to the same length of time as the last keyframe of that layer.
Numeric keyframes

Numeric keyframes are used for properties that use a numeric range such as 0-255 or 0-1.

Create a numeric keyframe

1. Left-click the keyframe editor of a layer at the point where you want to create a new keyframe. This will create a keyframe with the same value as the previous keyframe in the timeline.

Move a numeric keyframe

1. To move a keyframe, click it and drag it upwards, downwards, left or right. You can also adjust the keyframe using the mouse scroll wheel.

2. You can add a keyframe and position it in a single action; just click in the timeline and drag the new keyframe without releasing the mouse button.

Remove numeric keyframes

1. To remove one keyframe, rightclick it.

2. To remove multiple selected keyframes, hold down Shift and drag-select the keyframes. Hit the Delete button on the keyboard.
Editing min/max values

As with standard numeric properties, you can set min/max constraints to limit the positions of your Keyframes to a certain range.

1. Right-click a numeric property from the Layer editor to open the Numeric property editor.

   ![Brightness properties](image)

   - **Minimum**: 0
   - **Maximum**: 255
   - **Step**: 1
   - **Damping**: 0
   - **Expression**: = self

2. Type the min/max values you want to set in the **min / max** text fields.

3. Edit the value in the numeric property's text field in the Layer editor. If you input a value beyond the range specified in the min/max text fields, the value will automatically default to the min/max value (whichever is nearer) when you click anywhere on the screen. However, you can override this by manually moving the Keyframes on the Keyframe editor.

   For more information on **step**, **damping** and editing numeric properties in general please read the 'Editing numerical properties section in the Editing objects sub-chapter.

Right-click the keyframe editor to set the curve for all keyframes, or select a subset by holding shift and dragging a selection box over them. This will open an option menu that allows you to change the interpolation type (i.e. the shape of the curve that goes through the key values). There are three options:
Select

The output value remains static until the next keyframe.

Linear

The output follows a straight line between adjacent keyframes.
Cubic

The output value follows a cubic curve with zero in and out speed.
Object keyframes

The object keyframe editor consists of keyframes appearing as object thumbnails. The object library for that type opens up just below the keyframe editor, allowing you to drag thumbnails onto it.

Creating an object keyframe

Either drag a thumbnail from the object library onto the keyframe editor or left-click a thumbnail in the library to create a new keyframe at that current position.

Moving an object keyframe

Left-click the thumbnail in the keyframe editor and drag it left or right.
Removing object keyframes

- Rightclick the object thumbnail in the keyframe editor.
Option keyframes

Option property keyframe editors also consist of a series of keyframes, but each keyframe has an option choice. Option property keyframe editors behave similarly to numeric keyframe editors.

Creating an option keyframe

- Left-click the keyframe editor at the point where you want to create a new keyframe. This will create the keyframe and open an options menu for editing the keyframe.
Modifying an option keyframe

1. Left-click the keyframe and drag it left or right to change its time.

2. To select a different option, left-click the keyframe and release the mouse button immediately to open up the option list so you can choose a different value.

Removing an option keyframe

- Rightclick the keyframe to remove it.
Multiple keyframes

Editing actions apply to all three types of keyframe editors mentioned above and can be applied to a single keyframe editor or multiple keyframe editors of multiple layers.

Selecting a range of keyframes

1. Hold down the left Shift key and drag a rectangle around the keyframes you want to select inside the keyframe editor. Keyframes that fall under the selection rectangle are displayed as solid white dots.
2. To open more than one keyframe editor, hold down the **Ctrl** key when selecting the properties.
3. With multiple keyframe editors open, you can now select and edit keyframes from more than one keyframe editor simultaneously.
Editing a range of keyframes

Rightclicking inside the keyframe editor with a selected range of keyframes allows you to change the interpolation type between only the selected keyframes. The popup menu gives you a list of options to choose from.
Repeat a range of keyframes

1. Hold down the left Shift key to select a whole number of bars.
2. Rightclick the keyframe editor.
3. Select either repeat to end of layer or repeat to end of section.

Cut / copy / paste

1. Hold down the left Shift key to select a range of keyframes.
2. Rightclick the keyframe editor and select cut keyframes or copy keyframes.
3. Move to a different section of the timeline, rightclick the ribbon and select `paste keyframes`.

Clear a range of keyframes

1. Hold down the left `Shift` key to select a range of keyframes.

2. Rightclick the keyframe editor and select `delete keyframes`.
Compositing layers

Arrows provide a way of composing multiple layers together. For instance, you could take the output of a Video layer and pipe it into a Scroll layer, to scroll the video from side to side.

Multiple layers can be chained together, so for example, you could take the output of the Scroll layer and pipe it into a SpinBitmap layer.

1. To create an arrow between two layers, make sure the source layer is below the destination layer in the layer stack (because layers are rendered from the bottom up).

2. Position the cursor over the source layer, hold down the Alt key, press the mouse button and drag away from the source layer. You will see an arrow appear that is following the cursor.

3. Now move the mouse cursor to the destination layer and release the mouse button. This will create a new arrow between the source and destination layers.
Scrolling video content created by using an Arrow to pipe a Video layer into a Scroll layer
4. To remove an arrow, hold down the **Alt** key and right click the root of the arrow.

**Arrow-supported layers**

All content layers can be used for piping in content into another layer. However, not all layers support being piped into.

Currently, only the Bitmap and Effect Layers can be piped into.
Copying and pasting layers

A layer can be copied and pasted in the current track or even copied and pasted onto another.

Copying a layer

1. Right-click the layer to open the Layer menu.
2. Left-click **copy**.
Copying a layer will copy all of the layers keyframe events. It is recommended that you rename the layer to avoid future confusion. However you cannot copy and paste groups of layers. To do this, you can export and import layers. Please read the section Exporting / importing layers for more information on this feature.

Pasting a layer

1. Right-click the Track bar where you want to insert the layer on the Timeline. This will open the Track bars menu.

2. Left-click paste layer 'LayerName' to insert the layer copy at the currently selected point on the Timeline.
Deleting multiple layers

1. Hold down the left Shift key and left-click and drag a selection rectangle in the Content Layer section over the layers you want to select.

2. Right-click the selection rectangle to open the Multiple Layer menu.

3. Left-click delete. The selected layers will be immediately deleted from the Content Layer section.
Deleting layers

Deleting a layer

1. Right-click the layer to open the Layer menu.

2. Left-click delete. The layer will be immediately deleted from the Content Layer section.
Duplicating layers

Duplicating a layer

1. Right-click the layer to open the Layer menu.

2. Type the new layer name in the duplicate text field (if different from the default layer name) and hit Enter. The default layer name in the text field will be the name of the layer you are duplicating with a number added to the end.
Extending layers

You may want to extend a layer so that it renders content for a longer section of the Timeline.

Extending a layer

1. Left-click the layer to open two small tabs at the layers edges.

2. Hover your mouse cursor over the layers edges. You will notice the cursor change from a cross hair to a double ended arrow.

3. Left-click and drag the layers extender tabs to the left or right to extend the layer.

Extending a layer does not modify its keyframe events, they remain in the same position. If you reduce the extent of a layer, any key-frame events outside the layers extent remain there; if and when you extend the layer again, those key-frame events become accessible again.
Grouping layers

When working with large numbers of layers, it is convenient to group related layers together. This reduces the vertical height of the Timeline and makes it easier to move large numbers of layers.

Please note: that to arrow a layer into an existing group, the layer needs to be on top of the group in the compositing order before arrowing. Arrowing a layer from elsewhere on the timeline will not work.

You can also import and export a group of layers to the current track, or even onto another track in the Timeline. Please read the 'Importing and exporting layers section above for more information on this very useful feature.

Grouping layers

1. Hold down the left Shift key and left-click and drag a selection rectangle in the Content Layer section over the layers you want to group.

2. Right-click the highlighted layers to open the Layer menu.

3. Type the name of your group into the group text field and hit Enter. The grouped layers will turn orange.
Editing grouped layers

Left-click the + to expand the group layer.

The layers are revealed below the group layer and can be edited in the normal way.
Collapsing a group of layers

Click the - button on the Group layer.

Ungroup a grouped layer

1. Right-click the Group layer to open the Layer menu.
2. Left-click **ungroup**.

Removing a layer from the group

1. Make sure the group is expanded
2. Right-click on the layer you wish to remove from the group
3. Left-click on **Extract**

Adding a layer to a group

Hold down the **Alt** key and left-click drag an arrow from the layer to be group, to the group layer.

The layer will be added to the group layer.
Save as and importing layers

Layers can be saved to the layers library, from where they can be imported back onto the Timeline in another location.

Please note: Importing a layer makes a copy of the saved layer; changing the layer that you saved to the layer library does not change the copies.

This allows you to transfer layers or groups of layers in the current track, or even from one track into another track. This is extremely useful when sequencing, as you can store these saved layers in the Layers Library as templates to avoid having to repeat actions when creating layers.

Please note: This functionality used to be called Import & Export in previous releases.

Saving a layer

1. Right-click the layer to open the Layer menu.

2. Type the name of your exported layer (if different from the default layer name) in the Save as text field and hit OK. This will save the layer in the Layers Library.
Saving a layer will copy all of the layers keyframe events.

Importing a layer

1. Right-click the Track bar at the point where you want to import the layer on the Timeline. This will open the Track bars menu.

2. Left-click **import layer**. This will open the Layers Library.
3. Left-click the saved layer from the Layers Library.
The layer will be added to the track at the currently selected point on the Timeline.
Isolating layers

Sometimes it is useful to only see the output of a single layer when you have a stack of layers.

To see the output of a single layer:

1. Hold down the I key and left-click the layer. The other layers turn grey to indicate that the selected layer has been isolated.
2. Left-click anywhere in the Content Layer section to return the screen to the normal view.

Or alternatively

Right click on the layer and select the Isolate option.
Moving layers

Moving and re-ordering layers is particularly useful when you have a stack of layers.

Layers arranged in a stack are rendered in bottom up order: layers at the top are composited over layers at the bottom.

Moving/re-ordering a layer can be done in three ways.

Option 1

- Left-click and drag the layer left or right. You cannot drag the layer up or down. This method can be used to move multiple layers when you have a selection rectangle. Please read the Selecting multiple layers section below for more information on selection rectangles.

Option 2

1. Right-click the layer to open the Layer menu.

2. Left-click move up or move down.

Option 3

- Left-click the layer and hold Ctrl + Alt whilst using the keyboard arrow keys to move the layer up, down, left or right. Moving a layer moves all keyframe events.
Muting layers

It is possible to mute a layer to temporarily disable it, allowing you to see layers behind it in a stack of layers.

Muting a single layer:

1. Hold down the M key and left-click the layer.  
   Or alternatively

2. Right click on the layer and left click the Mute option.

The layer will turn dark grey to indicate that the selected layer has been muted.

Unmuting a single layer:

1. Hold down the M key and left-click the layer again.  
   Or alternatively

2. Right click on the layer and left click the unmute option.
The layer will return to its original colour.
Renaming layers

To rename a layer:

1. Right-click the layer to open the Layer menu.

2. Type the new layer name in the rename layer text field and hit Enter. The layers name will update instantly. A good layer naming format is Layer-type + description.
Selecting multiple layers

To select multiple layers:

- Hold down the left Shift key and left-click and drag a selection rectangle in the Content Layer section over the layers you want to select. This allows you to select a range of layers.

Selecting multiple layers will copy the selected layers keyframe events.

- Left-click anywhere in the Content Layer section to clear the selection rectangle.
Splitting layers

To split a layer:

1. Left-click a Track bar at the point on the Timeline where you want to split the layer.

2. Right-click the layer to open the Layer menu.

3. Left-click split. This will split the layer at the currently selected Timeline position into two shorter layers.

Splitting a layer does not modify its keyframe events, they remain in the same position. If you split the layer, any key-frame events outside the layers extent remain there; if and when you extend the layer again, those key-frame events become accessible again.
Sockpuppet overview

What is Sockpuppet

Sockpuppet fully implements DMX & OSC controls into the disguise software. This feature enables the user to fully control all layers and their parameters from a lighting desk or OSC application. As a result, the operator is no longer restricted to timeline based sequencing, but can now program their project using the lighting desk’s cue stack, or a combination of both; by having certain layers controlled from the lighting desk and others controlled from the Timeline. In short, Sockpuppet works by treating all of the layer types as fixture personalities.

Compatibility

Sockpuppet is fully compatible with any lighting desk capable of outputting Art-Net or sACN, as well as any OSC platform capable of sending OSC control strings.

Additionally, for added usabilityXML personalities can be exported from the disguise software and imported into any XML compatible console.

Known Limitations

Frame accurate sync across multiple machines cannot be guaranteed. This is because sockpuppet playback responds locally to control packets received on each machine, rather than the director receiving the control data and distributing it synchronously across the machines in the session. This is to guarantee as low latency response as possible.

In setups where a single screen is stitched from outputs across multiple servers, sockpuppet cannot guarantee synchronous playback across the whole surface.
Network setup - console

First ensure the lighting desk and server are networked together and able to ping each other. You will need to assign the IP address and Subnet mask on both the lighting desk's Art-Net port and the server's Art-Net port. These will normally be addressed as shown below:

- server - DMX port - 2.0.0.x - 255.0.0.0
- lighting desk - Art-Net port - 2.0.0.x - 255.0.0.0

Please note: Each device needs a unique IP address within the same IP range and subnet

Network setup - OSC

First ensure the OSC app and server are networked together and able to ping each other. You will need to assign the IP address and Subnet mask (ensuring they are both set to the same IP and Subnet range) on the system hosting the OSC application and the server's network port that is receiving the OSC data.

- Set OSC application send IP
- Set OSC application receive IP
- Set the OSC application outgoing port
- Set the OSC application incoming port
- Set the disguise software OSC device send IP
- Set the disguise software OSC device receive IP
- Set the OSC the disguise software outgoing port
- Set the OSC the disguise software incoming port

**Confirm the connection**

To check the two devices can communicate with each other, a command line called **ping** should be used.

**Please note:** If using sockpuppet on a Director/Actor system, you will need to give each machine its own Art-Net or OSC connection as each machine needs to receive the data from the control source.

**Setup**

Once network connectivity has been confirmed between the server and the lighting desk, the next step is to create a DMX or OSC device in the disguise software.

Read more about DMX setup [here](#).

Read more about OSC devices [here](#).
Sockpuppet overview

What is Sockpuppet

Sockpuppet fully implements DMX & OSC controls into the disguise software. This feature enables the user to fully control all layers and their parameters from a lighting desk or OSC application. As a result, the operator is no longer restricted to timeline based sequencing, but can now program their project using the lighting desk’s cue stack, or a combination of both; by having certain layers controlled from the lighting desk and others controlled from the Timeline. In short, Sockpuppet works by treating all of the layer types as fixture personalities.

Compatibility

Sockpuppet is fully compatible with any lighting desk capable of outputting Art-Net or sACN, as well as any OSC platform capable of sending OSC control strings.

Additionally, for added usability XML personalities can be exported from the disguise software and imported into any XML compatible console.

Known Limitations

Frame accurate sync across multiple machines cannot be guaranteed. This is because sockpuppet playback responds locally to control packets received on each machine, rather than the director receiving the control data and distributing it synchronously across the machines in the session. This is to guarantee as low latency response as possible.

In setups where a single screen is stitched from outputs across multiple servers, sockpuppet cannot guarantee synchronous playback across the whole surface.
Network setup - console

First ensure the lighting desk and server are networked together and able to ping each other. You will need to assign the IP address and Subnet mask on both the lighting desk's Art-Net port and the server's Art-Net port. These will normally be addressed as shown below:

- server - DMX port - 2.0.0.x - 255.0.0.0
- lighting desk - Art-Net port - 2.0.0.x - 255.0.0.0

Please note: Each device needs a unique IP address within the same IP range and subnet

Network setup - OSC

First ensure the OSC app and server are networked together and able to ping each other. You will need to assign the IP address and Subnet mask (ensuring they are both set to the same IP and Subnet range) on the system hosting the OSC application and the server's network port that is receiving the OSC data.

- Set OSC application send IP
- Set OSC application receive IP
- Set the OSC application outgoing port
- Set the OSC application incoming port
- Set the disguise software OSC device send IP
- Set the disguise software OSC device receive IP
Set the OSC the disguise software outgoing port

Set the OSC the disguise software incoming port

Confirm the connection

To check the two devices can communicate with each other, a command line called ping should be used.

Please note: If using sockpuppet on a Director/Actor system, you will need to give each machine its own Art-Net or OSC connection as each machine needs to receive the data from the control source.

Setup

Once network connectivity has been confirmed between the server and the lighting desk, the next step is to create a DMX or OSC device in the disguise software.

Read more about DMX setup here.

Read more about OSC devices here.
Bank Editor

Please note: it is important to note for best results have the console set up to work in decimal rather than percent.

What is the bank editor?

The bank editor enables us to assign DMX values to the various resources needed within a layer, examples of such resource parameters include, video or bitmap files, mapping types or HTML bookmarks. This process is managed on a bank and slot basis.

Banks & slots

Within the layer personality, sockpuppet will assign two channels to each resource parameter, the first of these channels will be for the resource bank, and the second will be for the resource slot. For example within the video layer personality, channel 8 is for the video bank & channel 9 is for the video slot.

Example: If we had a video asset assigned to video slot 5 in video bank 3, we would need to output the following from the lighting console to recall it.

channel 8 @ 3, channel 9 @ 5.

This means that we can have a maximum of 255 video banks and within each of those we can have 255 video slots, giving us a total of 65025 assignable video slots (255*255) for each resource type.
Main Interface elements

To open the Bank editor right click on the state menu (top left) and underneath the sockpuppet tab left click "bank".

At the top of the window we have the currently selected resource type, in this case, video clip. Left click Video Clip to open a menu for other resource types. You can populate the resource bank with resources by dragging them from the resources view in the bottom of the window, to empty bank slots.

Preferences

The bank editor preferences can be opened by left clicking the small white triangle in the top right corner of the window.
View type

Here we can select the way in which we want to view the bank editor. Grid will display the slot window in a grid format and List will display the slot window as a list.

Columns

Specifies the number of columns displayed by default.

Parameter tracking

By left clicking on this field and setting it to "on" we will have a visual representation of which slot we currently have selected; so as the console programmer scrolls through the slots and banks on the encoders, a red square will be drawn over the slot we currently have selected.

Auto-populate

When disabled, it stops assets being added automatically based on their file name.
Search function

This enables you to search through the assets already assigned to a slot, and will highlight them with a light blue square if they match the search criteria.

Working with banks

The banks are all configured and managed in the tabs bar above the slot layout. By default we will have bank 0 already created. In order to open the bank options window right click on an existing bank or on bank 0.

Rename

Here we can specify a custom name for our bank, type it into the field and click ok or press the return key.

Duplicate

Here we can copy the contents of our selected bank into another, specify the destination bank number in the field and click ok or press the return key.
New bank

Here we can create a new bank. To do this enter the number for your new bank and click ok or press the return key.

Auto-populate banks from directory

Please see below, "under adding assets to slots"

Reset bank

This will return the bank to its default state, and empty any populated slots.

Remove missing

Allows you to remove missing media from a selected bank.

Delete bank

By left clicking here, we will completely delete the bank from our bank editor.

Adding assets to slots

Manually add assets to slots

To manually add an asset to a slot, simply select the resource type you wish to assign, such as textures, left click on the thumbnail in the library and drag and drop it over the slot you wish to assign it to. Repeat this process for all of your content. If you have your content organised into files and boxes (click here to read about organising your content) then you can quickly populate your slots with whole...
boxes at a time. To do this simply hold down Alt on the keyboard and left click over the box you wish to use, this will turn our cursor into a white arrow which we can drag from our box to the first slot we wish to use, release and it will populate the slots with the contents of that box.

**Using auto-populate to add assets to slots**

Another widely used method of assigning DMX values to assets, is adding numerical values to folders and files within your assets library. The disguise software also supports this workflow and is managed via the auto-populate field in the bank options (see above for how to access).

**Auto-populate file structure:**

Each of the content folders located within the objects/videofile folder needs to be labelled numerically, for example: 001_Foldername

Files within this folder should follow this same numeric labelling scheme: 001_filename
Auto-populated results within the Bank Editor

**Creating a directory**

When using this method to populate your banks and slots, you will need to start by numbering your folders and files within the project directory in Windows.

Navigate to the resource folder you wish to use (we will be using DxTexture as an example) and number your folders within the DxTexture folder, these will become your bank numbers. Within each of these folders you will need to number your asset files, these will become your slot numbers.

**Clearing assets from slots**

In order to clear a slot we will need to overwrite it with the "none" thumbnail from our resource library. To do this simply left click on the none thumbnail and drag it over a populated slot. We could also use the reset bank option to clear out the whole bank.

**Banks with OSC**

As of r17.3, banks can also be used with OSC sockpuppet.

- The path is always the same: /d3/layer/Typeoflayer/Nameoflayer - Example: /d3/layer/Video/video
- It's always one message (One address/path + one or two integers)
Option one: One path and one integer with bank and slot encoded into one number (e.g., bank 1 slot 1 is \(256 + 1 = 257\)).

The formula for a single int is \(\text{bank} \times 255 + \text{slot}\) (e.g., bank 0 slot 1 is \(0 \times 255 + 1\)).

Option two: One path and two integers with bank in the first integer and slot in the second integer.
Patch assignments

Sockpuppet treats all of disguise’s layer types as fixture personalities. The Patch Assignments window is where we patch or assign our layers on our Timeline to incoming DMX values or channels. For OSC the process is slightly different.

DMX

Patching a layer to DMX control

1. Access the patch assignments window by right-click on the d3 icon in the top left corner of the interface.

2. Navigate down to the Sockpuppet tab.

3. Beneath this you will find two fields, the first will be Patch assignments and the second Bank assignments.

4. Left-click Patch Assignments to open the patch assignments window.
<table>
<thead>
<tr>
<th>Viewing universe</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>33</td>
<td>34</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
</tr>
<tr>
<td>65</td>
<td>66</td>
</tr>
<tr>
<td>81</td>
<td>82</td>
</tr>
<tr>
<td>97</td>
<td>98</td>
</tr>
<tr>
<td>113</td>
<td>114</td>
</tr>
<tr>
<td>129</td>
<td>130</td>
</tr>
<tr>
<td>145</td>
<td>146</td>
</tr>
<tr>
<td>161</td>
<td>162</td>
</tr>
<tr>
<td>177</td>
<td>178</td>
</tr>
<tr>
<td>193</td>
<td>194</td>
</tr>
<tr>
<td>209</td>
<td>210</td>
</tr>
<tr>
<td>225</td>
<td>226</td>
</tr>
<tr>
<td>241</td>
<td>242</td>
</tr>
<tr>
<td>257</td>
<td>258</td>
</tr>
<tr>
<td>273</td>
<td>274</td>
</tr>
<tr>
<td>289</td>
<td>290</td>
</tr>
<tr>
<td>305</td>
<td>306</td>
</tr>
<tr>
<td>321</td>
<td>322</td>
</tr>
<tr>
<td>337</td>
<td>338</td>
</tr>
<tr>
<td>353</td>
<td>354</td>
</tr>
<tr>
<td>369</td>
<td>370</td>
</tr>
<tr>
<td>385</td>
<td>386</td>
</tr>
<tr>
<td>401</td>
<td>402</td>
</tr>
</tbody>
</table>
View Types

Left click on the view type field to open the list of available view options.

**Patch grid**

Displays all 512 channels in a grid format. You are able to patch to any free channels here or view which channels you have already populated with a patch. Left-click on Patch Grid to view different options to change the layout of the patch window.

Please note: this is the view type you will need to be in to patch your layers.

**Patch list**

This shows us a list of all of the current layers patched and their start channels within any selected universe (selectable from the Viewing Universe field).

**Global patch list**

This follows the same format as the Patch List, however this will show us a full list of all layers patched regardless of which universe they start on.

**Universe occupancy**

This will show us in grid format which of our universes we currently have layers patched to (a blue square over any universe indicates we have layers patched).

**Patching layers**

To patch layers within the disguise software ensure:
The Patch Assignments window is open & in the patch grid view type.

The layer you want to patch is added to the Timeline. Click here for more information on working with layers.

Hover the cursor over the DMX channel you want to patch to in the Patch Assignments window.

Select the DMX channel while holding down ALT to generate a white arrow, which should be dragged down to the layer on the Timeline that you want to patch to.
This will create a green block over the channels in our patch grid that the layer will take up, and also turn the layer on the timeline green to signify the layer is patched and only controllable through Sockpuppet DMX.

We will also be presented with some key information regarding our patch. This will include the start address, a user specified name (this will be "untitled" by default) the layer type, and the personality version.
Please note: Any previous keyframing / sequencing done on the layer will be overwritten and deleted by the default values sent from the desk once the layer is patched.

Patch settings

To access this patch settings menu, right click on the green patch block, this will open the patch settings.

Name

Here we can give the layer a user friendly name, this might be particularly useful if we have a high number of video layers patched for example, and want to easily distinguish which ones are which.

Universe

Here we can specify the universe we want to patch our layer to.
Start channel

Here we can specify the start channel we want to patch our layer to.

Patch info

Here we are presented with some basic information as to where and how many layers our patch assignment is connected to.

Show Personality

By left clicking this, we are presented with a window displaying the layers personality. This is a table highlighting which incoming DMX channel is assigned to what function or parameter within the layer, in addition to this we are shown the default values of channels, their ranges, and the title displaying the layer type and its total footprint size.

Show Personality XML

Left clicking Show personality XML will do two things. The first will present the user with a window showing the video layer personality in XML form. The second action performed will be to create a new folder in the d3 projects folder, this will be named "output". (see the below file directory)

Computer > Media (E:) > d3 Projects > Project Name > output

This folder will be populated with an XML file containing the Personality data for the selected layer. This can then be transferred via USB stick onto your XML compatible lighting console.

Please note: You will need to follow this process for each layer type patched. A new XML document will be created within this output folder, for each new layer type that the user opens the show personality XML function on.
Personality Versions

Here we will find an archive of the previous personality versions used in our project. A personality profile may be updated / changed with a new release of the disguise software, this function will allow you to roll back to the personality version that you originally programmed your show with, meaning you will not have to update any of the programming on your console.

Delete Patch

By left clicking on delete patch we will remove the selected layer from our patch assignment. This will return it to its standard operating mode (controlled through the timeline & keyframbale)

OSC

Patching a layer to OSC control

1. Right click a layer on the timeline

2. Left click Patch to OSC
3. Choose OSC device

The layer will now turn green, indicating that it is controlled by an external sockpuppet system.
View the personality

1. To view the personality of the layer for OSC, right click the layer and choose Edit OSC patch.

2. Left click View personality

The default OSC personality opens.
Export the personality

1. Repeat the previous steps to view the OSC patch.

2. Once the patch is open, left click the Export button at the bottom of the window.
Sockpuppet personality editor

The **Sockpuppet** personality editor is a feature of the disguise software that allows users to edit the default personality of a patched layer.

It's important to note that currently, the functionality is limited to patched layers and you cannot store a custom personality for use with other layers of the same type.

The functionality is useful for setting 16-bit control of some control channels, as well as removing parameters from the personality.

### Accessing & using the personality editor

1. Patch a layer as specified in the [patch assignments topic](#).

2. Right click the patched layer in the patch assignments widget.
3. In the DMX Patch Settings widget, right click the **DmxPersonality** object.
The personality editor will open. You can now edit values in the editor which will be stored to the personality in use for that layer.
<table>
<thead>
<tr>
<th>Channel Offset</th>
<th>Type</th>
<th>Field</th>
<th>Display Name</th>
<th>Group Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>options</td>
<td>blendMode</td>
<td>blendMode</td>
<td>Default</td>
<td>0</td>
<td>18</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>8-bit</td>
<td>brightness</td>
<td>brightness</td>
<td>Default</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>bank/slot</td>
<td>mapping</td>
<td>mapping</td>
<td>Default</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>RGB-colour</td>
<td>RGB colour</td>
<td>RGB colour</td>
<td>Default</td>
<td>0</td>
<td>1</td>
<td>255</td>
</tr>
<tr>
<td>4</td>
<td>bank/slot</td>
<td>video</td>
<td>video</td>
<td>Media</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>16-bit</td>
<td>speed</td>
<td>speed</td>
<td>Media</td>
<td>-4</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>options</td>
<td>mode</td>
<td>mode</td>
<td>Media</td>
<td>0</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>options</td>
<td>at end point</td>
<td>at end point</td>
<td>Media</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>16-bit</td>
<td>loop inFrame</td>
<td>loop inFrame</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>16-bit</td>
<td>transition time</td>
<td>transition time</td>
<td>Media</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>8-bit</td>
<td>volume</td>
<td>volume</td>
<td>Audio</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>11</td>
<td>bank/slot</td>
<td>Output</td>
<td>Output</td>
<td>Audio</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>12</td>
<td>16-bit</td>
<td>brightness (shift)</td>
<td>brightness (shift)</td>
<td>ColourShift</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>13</td>
<td>16-bit</td>
<td>contrast (scale)</td>
<td>contrast (scale)</td>
<td>ColourShift</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>16-bit</td>
<td>saturation scale</td>
<td>saturation scale</td>
<td>ColourShift</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>16-bit</td>
<td>hue shift</td>
<td>hue shift</td>
<td>ColourShift</td>
<td>-360</td>
<td>360</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>options</td>
<td>RGB controlled</td>
<td>RGB controlled</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>17</td>
<td>8-bit</td>
<td>red min</td>
<td>red min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>18</td>
<td>8-bit</td>
<td>red max</td>
<td>red max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>8-bit</td>
<td>red gamma</td>
<td>red gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>8-bit</td>
<td>green min</td>
<td>green min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>21</td>
<td>8-bit</td>
<td>green max</td>
<td>green max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>22</td>
<td>16-bit</td>
<td>green gamma</td>
<td>green gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>23</td>
<td>8-bit</td>
<td>blue min</td>
<td>blue min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>24</td>
<td>8-bit</td>
<td>blue max</td>
<td>blue max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>25</td>
<td>16-bit</td>
<td>blue gamma</td>
<td>blue gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>8-bit</td>
<td>key colour.r</td>
<td>key colour.r</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>27</td>
<td>8-bit</td>
<td>key colour.g</td>
<td>key colour.g</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>8-bit</td>
<td>key colour.b</td>
<td>key colour.b</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>29</td>
<td>options</td>
<td>clip_type</td>
<td>clip_type</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>30</td>
<td>16-bit</td>
<td>size</td>
<td>size</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>16-bit</td>
<td>scale.x</td>
<td>scale.x</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>32</td>
<td>16-bit</td>
<td>scale.y</td>
<td>scale.y</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>16-bit</td>
<td>pos.x</td>
<td>pos.x</td>
<td>Move</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>16-bit</td>
<td>pos.y</td>
<td>pos.y</td>
<td>Move</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>35</td>
<td>16-bit</td>
<td>rotation</td>
<td>rotation</td>
<td>Move</td>
<td>-180</td>
<td>180</td>
<td>0</td>
</tr>
<tr>
<td>36</td>
<td>16-bit</td>
<td>left</td>
<td>left</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>37</td>
<td>16-bit</td>
<td>right</td>
<td>right</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>16-bit</td>
<td>top</td>
<td>top</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>39</td>
<td>16-bit</td>
<td>bottom</td>
<td>bottom</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>40</td>
<td>options</td>
<td>on clock lost</td>
<td>on clock lost</td>
<td>Timecode</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>8-bit</td>
<td>hours</td>
<td>hours</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>8-bit</td>
<td>minutes</td>
<td>minutes</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>8-bit</td>
<td>seconds</td>
<td>seconds</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>8-bit</td>
<td>frames</td>
<td>frames</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>45</td>
<td>8-bit</td>
<td>threshold</td>
<td>threshold</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>46</td>
<td>16-bit</td>
<td>hardness</td>
<td>hardness</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>47</td>
<td>16-bit</td>
<td>cropSoftness</td>
<td>cropSoftness</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>48</td>
<td>16-bit</td>
<td>frame index</td>
<td>frame index</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
<tr>
<td>49</td>
<td>16-bit</td>
<td>trim frames</td>
<td>trim frames</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
<tr>
<td>50</td>
<td>16-bit</td>
<td>layer outFrame</td>
<td>layer outFrame</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
</tbody>
</table>
Use the actions tab to show or export the personality in either text or grandMA2 format.

4. Editing and creating a custom DMX personality:
   
   **Removing channels**
   
   Channels can be removed from a personality as well to create a custom DMX profile. Right-click on the empty space between a column in the DMXPersonality editor and select Remove Properties.
<table>
<thead>
<tr>
<th>Channel Offset</th>
<th>Type</th>
<th>Field</th>
<th>Display Name</th>
<th>Group Name</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>18</td>
<td>bank/slot</td>
<td>Output</td>
<td>Output</td>
<td>Audio</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>16-bit</td>
<td>brightness (shift)</td>
<td>brightness (shift)</td>
<td>ColourShift</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>16-bit</td>
<td>contrast (scale)</td>
<td>contrast (scale)</td>
<td>ColourShift</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>24</td>
<td>16-bit</td>
<td>saturation scale</td>
<td>saturation scale</td>
<td>ColourShift</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>26</td>
<td>16-bit</td>
<td>hue shift</td>
<td>hue shift</td>
<td>ColourShift</td>
<td>-360</td>
<td>360</td>
<td>0</td>
</tr>
<tr>
<td>28</td>
<td>options</td>
<td>RGB controlled</td>
<td>RGB controlled</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>29</td>
<td>8-bit</td>
<td>red min</td>
<td>red min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>8-bit</td>
<td>red max</td>
<td>red max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>31</td>
<td>16-bit</td>
<td>red gamma</td>
<td>red gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>33</td>
<td>8-bit</td>
<td>green min</td>
<td>green min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>34</td>
<td>8-bit</td>
<td>green max</td>
<td>green max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>35</td>
<td>16-bit</td>
<td>green gamma</td>
<td>green gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>37</td>
<td>8-bit</td>
<td>blue min</td>
<td>blue min</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>38</td>
<td>8-bit</td>
<td>blue max</td>
<td>blue max</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>39</td>
<td>16-bit</td>
<td>blue gamma</td>
<td>blue gamma</td>
<td>ColourShift</td>
<td>0.1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>41</td>
<td>8-bit</td>
<td>key colour.x</td>
<td>key colour.x</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>8-bit</td>
<td>key colour.y</td>
<td>key colour.y</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>43</td>
<td>8-bit</td>
<td>key colour.z</td>
<td>key colour.z</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>44</td>
<td>options</td>
<td>clip_type</td>
<td>clip_type</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>45</td>
<td>16-bit</td>
<td>size</td>
<td>size</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>47</td>
<td>16-bit</td>
<td>scale.x</td>
<td>scale.x</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>49</td>
<td>16-bit</td>
<td>scale.y</td>
<td>scale.y</td>
<td>Move</td>
<td>0</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>51</td>
<td>16-bit</td>
<td>pos.x</td>
<td>pos.x</td>
<td>Move</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>53</td>
<td>16-bit</td>
<td>pos.y</td>
<td>pos.y</td>
<td>Move</td>
<td>-1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>55</td>
<td>16-bit</td>
<td>rotation</td>
<td>rotation</td>
<td>Move</td>
<td>-180</td>
<td>180</td>
<td>0</td>
</tr>
<tr>
<td>57</td>
<td>16-bit</td>
<td>left</td>
<td>left</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>59</td>
<td>16-bit</td>
<td>right</td>
<td>right</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>61</td>
<td>16-bit</td>
<td>top</td>
<td>top</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>63</td>
<td>16-bit</td>
<td>bottom</td>
<td>bottom</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>65</td>
<td>options</td>
<td>on clock lost</td>
<td>on clock lost</td>
<td>Timecode</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>66</td>
<td>8-bit</td>
<td>hours</td>
<td>hours</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>67</td>
<td>8-bit</td>
<td>minutes</td>
<td>minutes</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>68</td>
<td>8-bit</td>
<td>seconds</td>
<td>seconds</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>69</td>
<td>8-bit</td>
<td>frames</td>
<td>frames</td>
<td>Timecode</td>
<td>0</td>
<td>255</td>
<td>0</td>
</tr>
<tr>
<td>70</td>
<td>8-bit</td>
<td>threshold</td>
<td>threshold</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>71</td>
<td>16-bit</td>
<td>hardness</td>
<td>hardness</td>
<td>ColourShift</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>73</td>
<td>16-bit</td>
<td>cropSoftness</td>
<td>cropSoftness</td>
<td>Crop</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>75</td>
<td>16-bit</td>
<td>frame index</td>
<td>frame index</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
<tr>
<td>77</td>
<td>16-bit</td>
<td>trim frames</td>
<td>trim frames</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
<tr>
<td>79</td>
<td>16-bit</td>
<td>loop outFrame</td>
<td>loop outFrame</td>
<td>Media</td>
<td>0</td>
<td>65535</td>
<td>0</td>
</tr>
</tbody>
</table>
Creating a custom DMX personality

The Channel Offset, Display Name, Group Name, and Min/Max/Default Values can also be customized in this editor and exported.
Sequencing with Sockpuppet

There are a number of changes to the way in which layers are programmed or sequenced when running in sockpuppet mode. The following section will outline these differences.

Dormant layers

Overview

Dormant layers is a feature to enable the stacking of large numbers of layers on the timeline. The issue with this previously was that each layer on the timeline required a certain amount of system resources in order to run, which would quickly add up when working with large numbers and result in a significant drop in performance. Now when a layer is set to be dormant it will not require any system resources, and therefore mean we can have large stacks of layers on our timeline and not have to worry about performance issues.

Setting a layer to be dormant

To make a layer dormant simply set the brightness to 0. This will change the colour of the layer to a dark grey to signify the layer is now dormant. Raise the value above zero to make the layer active.

RGB colour overview

When running in sockpuppet mode the colour pallette system with its X and Y values are ignored and instead replaced by a RGB colour system. This is to reflect the typical way of working with colour on a lighting console.

Please note: there is a know issue whereby the RGB colour control system will not be reflected in the layer editor within the disguise software, this will be fixed in a later release.

Video playback modes overview

When running a layer in sockpuppet mode there are a number of behaviour changes and additions to the video play modes. The functions are listed below:
**Normal**

Play forward

**Locked**

Play forward: no difference from Normal

**Reset**

Reset to the first frame in the clip and hold

**Pause**

Pause at the current frame

**Inframe**

Go to a specific frame

**Outframe**

Set a specific outframe

**Reverse**

Play clip backwards

**Timecode**

Playback will chase timecode

**LoopInFrame**

Set a specific inframe for a video clip loop
**LoopOutFrame**

Set a specific outframe for a video clip loop

**Toggle Sockpuppet**

The toggle sockpuppet feature allows us to remove the sockpuppet functionality on any parameter within a patched layer.

If for example, we have a Bitmap layer patched and we want to have DMX control over everything apart from the brightness parameter, which we want to keyframe. We can now right click on the parameter we want to toggle, and click "toggle sockpuppet" this will now take my parameter out of the sockpuppet mode.
Universal crossfade overview

What is Universal crossfade?

Universal crossfade allows you to smoothly dissolve / fade between any two points in a show.

Up to r11.3, the only way to transition from one point in a track to another, or to another track, was a hard cut. In r12 onwards, you can jump from any point in a track to any other point, or to another track, with a smooth dissolve.

Crossfade duration

Controlling the duration of the dissolve can be done in three ways:

- Globally across the entire project: open the state manager by right clicking the d3 icon on the top left of the gui, navigate to Project Settings and see option Global Crossfade Duration.

- Per-track: right-click on the track title bar, select Special settings, switch Default Crossfade to Fade or TrackSection, and select the transition time you want to use.

- Per section: right-click on the first bar of the section, under the Crossfade tab, select either Fade or TrackSection and configure the parameters accordingly.

Fade transitions

Fade transitions is the mode universal crossfade has always used.
These can be set at either a global level, track level or section level. Each overriding the last in order of precedence.

To access the fade settings:

1. Right click a section

2. Set the crossfade from Undefined to **Fade**.

3. Set the duration to be whatever time you wish the fade to occur over.
4. The first second of each section shows a triangular indentation to indicate that jumping into the section will initiate a dissolve of that duration. The duration is always controlled by the destination section.

The fade is started under two circumstances:
Jumping to another point on the timeline

Holding at the end of a section and going forward into the next section

Please note: that playing across a section boundary will not start the dissolve. Only an action that forces the playhead to another location on the timeline will trigger the crossfade.

Crossfade loop

When in loop section playmode, the playhead can crossfade to the beginning of the current section x seconds before the end of the section. Where x is the duration of the fade or the length of the specified track section used to render the transition. This helps create a more seamless fade back to the start of a section rather than a hard cut.

1. Right click on a section and set the transition to fade.
2. Check Loop XFade box.
3. The section will now fade back to the beginning using the duration set.

Sequenced transitions

As of r17.1, you can sequence a transition instead of using universal crossfade which only allows a straight dissolve. Note that sequenced transitions can only be used at the section level currently.

Individual RGBA components are actually transitioned independently.

During a transition, all layer types are expected to work except those which deal with transport commands or status. Examples include:

- TransportVolumeLocal
- TransportBrightnessLocal
- TransportControl
Audio is played back during transitions. This is to allow an audio component to the transition.

Prefetch is not performed - any 'heavy' video files will likely fail on the first frame. A workaround would be to offset the transition video forward by a frame or two.

**Setting up a sequenced transition**

1. Create a track called “transitions”.
2. Create sections within the track that define your transition lengths.
3. Within your section, sequence a video file or texture starting black, and ending white. Playmode should be set to “pause at end”.
4. Give the section a name by using a note.
5. On your main show track, right click a section to define a section based crossfade. This can also be set via Telnet when using the Multi-transport API.
6. Choose “Track Section” in place of the crossfade override.
7. Choose the transition section to use.
Universal crossfade overview

What is Universal crossfade?

Universal crossfade allows you to smoothly dissolve / fade between any two points in a show.

Up to r11.3, the only way to transition from one point in a track to another, or to another track, was a hard cut. In r12 onwards, you can jump from any point in a track to any other point, or to another track, with a smooth dissolve.

Crossfade duration

Controlling the duration of the dissolve can be done in three ways:

- Globally across the entire project: open the state manager by right clicking the d3 icon on the top left of the gui, navigate to Project Settings and see option Global Crossfade Duration.

- Per-track: right-click on the track title bar, select Special settings, switch Default Crossfade to Fade or TrackSection, and select the transition time you want to use.

- Per section: right-click on the first bar of the section, under the Crossfade tab, select either Fade or TrackSection and configure the parameters accordingly.

Fade transitions

Fade transitions is the mode universal crossfade has always used.

These can be set at either a global level, track level or section level. Each overriding the last in order of precedence.
To access the fade settings:

1. Right click a section

2. Set the crossfade from Undefined to **Fade**.

3. Set the duration to be whatever time you wish the fade to occur over.
4. The first second of each section shows a triangular indentation to indicate that jumping into the section will initiate a dissolve of that duration. The duration is always controlled by the destination section.

The fade is started under two circumstances:
Jumping to another point on the timeline

Holding at the end of a section and going forward into the next section

Please note: that playing across a section boundary will not start the dissolve. Only an action that forces the playhead to another location on the timeline will trigger the crossfade.

Crossfade loop

When in loop section playmode, the playhead can crossfade to the beginning of the current section x seconds before the end of the section. Where x is the duration of the fade or the length of the specified track section used to render the transition. This helps create a more seamless fade back to the start of a section rather than a hard cut.

1. Right click on a section and set the transition to fade.
2. Check Loop XFade box.
3. The section will now fade back to the beginning using the duration set.

Sequenced transitions

As of r17.1, you can sequence a transition instead of using universal crossfade which only allows a straight dissolve. Note that sequenced transitions can only be used at the section level currently.

Individual RGBA components are actually transitioned independently.

During a transition, all layer types are expected to work except those which deal with transport commands or status. Examples include:

- TransportVolumeLocal
- TransportBrightnessLocal
- TransportControl
TrackJump

Readout

Audio is played back during transitions. This is to allow an audio component to the transition.

Prefetch is not performed - any 'heavy' video files will likely fail on the first frame. A workaround would be to offset the transition video forward by a frame or two.

Setting up a sequenced transition

1. Create a track called “transitions”.
2. Create sections within the track that define your transition lengths.
3. Within your section, sequence a video file or texture starting black, and ending white. Playmode should be set to “pause at end”.
4. Give the section a name by using a note.
5. On your main show track, right click a section to define a section based crossfade. This can also be set via Telnet when using the Multi-transport API.
6. Choose “Track Section” in place of the crossfade override.
7. Choose the transition section to use.
Indirections

Indirections are keyframes that can be replaced remotely when the user sends a specified command to the system.

Overview

Indirections are primarily used for switching out pre-programmed content on the fly. They can be used for situations where the user may wish to change objects under certain scenarios such as triggering of an external system or changing playback to match different talent performing on stage.

Indirections are controlled by an Indirection controller, of which there are five types:

**Manual**

Holds a single resource that the Indirection uses. They can be manually controlled via the UI or via the new Indirections API.

**List**

The list controller type holds a list of objects and has an index for which object they refer to. The list can be controlled manually via the UI or via the Indirections API.

**OSC**

The OSC controller holds an address, for example - "/d3/indirection/name" and an OSC device. It's controlled by sending a string to The disguise software via OSC in the following format:

**Address:** /d3/indirection/nameOfIndirectionToControl

**Message:** sample/george.jpg

**DMX**

The DMX controller holds a channel and a universe. Its controlled by sending a value on that channel/universe that corresponds to a DMX bank & slot.

**UDP**

The UDP controller holds a string-keyed list of objects. It's controlled by sending a raw UDP message with the specified string to select the associated object.
Indirections can also be accessed via a dedicated API, documentation for which can be found at http://localhost/docs/v1/index.html whilst you have The disguise software running.

**Warning:** OSC, DMX and UDP Indirection controllers do not hold their values through restarts of the disguise software.

**Warning:** OSC and DMX indirection controllers can be set to any object that you've specified or put in a bank/slot which means you must take care to select the correct type of object. If you set a keyframe indirection to an object that does not match the keysequence it is sequenced in (i.e. you set your Indirection to reference a mapping when it expects a videoClip) the indirection will instead switch to holding None.

**Workflow**

- An Indirection acts somewhat like a "dynamic" keyframe
- The dynamic keyframe has a controller attached to it
- The controller gets values sent to it
- Based on those values the resulting output can be changed temporarily

**Examples**

**Manual**

Manual indirections are extremely useful if you want to make a quick change to a defined resource via the The disguise software user interface. For example, swap out a logo during a corporate event.
1. Open a content layer, such as Video

2. Open the keyframe editor for the resource you wish to change, i.e. videoClip

3. Create a new Indirection

4. Give the Indirection a unique name

5. In the Indirection create a controller

6. Choose the manual controller type

7. In the manual controller, choose a resource to assign to the controller

8. Once this is done, the resource can be changed and the layer is updated.
List

List is a useful controller for when you have a small selection of resources you wish to change out or swap between. For example toggling between two pieces of media for when a different performer is covering for someone during a show.

1. Open a content layer, such as Video
2. Open the keyframe editor for the resource you wish to change, i.e. videoClip
3. Create a new Indirection
4. Give the Indirection a unique name
5. In the Indirection create a controller
6. Choose the list controller type

7. In the list controller, choose resources to assign to the controller
   a. Resources can be re-ordered like other lists in the disguise software. Drag the items up and down to re-order.
   b. The index value cycles through resources, so the ordering of the list is important.
   c. The Index value can have an expression which can be linked to an external device, for example a midi controller.

8. Once this is done, the resource can be changed and the layer is updated.
OSC

The OSC indirection is useful if you want to create a more complex front end interface on a device such as an iPad, or apply logic to your Indirections.

1. Open a content layer, such as Video

2. Open the keyframe editor for the resource you wish to change, i.e. videoClip

3. Create a new Indirection
4. Give the Indirection a unique name

5. In the Indirection create a controller

6. Choose the OSC controller type

7. In the OSC controller, choose values to assign to the controller
   a. Set the OSC device that the controller is receiving data from.
   b. Set the OSC address that the controller is listening to.

8. Once this done, the resource can be changed and the layer is updated.
DMX

DMX is useful for being able to change Indirections via a lighting desk, much in the same way sockpuppet works, but without needing to set up a full sockpuppet programming workflow.

1. Open a content layer, such as Video

2. Open the keyframe editor for the resource you wish to change, i.e. videoClip

3. Create a new Indirection

4. Give the Indirection a unique name

5. In the Indirection create a controller
6. Choose the DMX controller type

![Select type of indirection controller]

7. In the DMX controller, choose values to assign to the controller
   a. Set the universe & address
   b. Ensure the resources you wish to control are exposed in the bank assignments, much like sockpuppet.

8. Once this is done, the resource can be changed and the layer is updated.

![dmx]

## UDP

The UDP Indirection is useful for more complex use cases where an external system is running logic to decide which Indirections are being used.

1. Open a content layer, such as Video
2. Open the keyframe editor for the resource you wish to change, i.e. videoClip
3. Create a new Indirection
4. Give the Indirection a unique name

5. In the Indirection create a controller

6. Choose the UDP controller type

7. In the UDP controller, choose resources to assign to the controller
   
   a. Set the key(s) for your resource(s)

8. Once this is done, the resource can be changed and the layer is updated.
Generating textures

Texture types

**Composite Texture**
Combines multiple textures into one. To blend the textures together, you can select different blend modes.

**Gradient Texture**
Gradient texture allows for generative gamma and gradient length options.

**Soft Edge texture**
Create as many points as you want and configure what gamma level and level of gradient you need to achieve the perfect blend between your projector outputs.

**Solid texture**
Texture that only allows one solid color.

**Lookup texture**
This texture allows you to edit the brightness curve of a softedge texture.

**PreComp Texture**
This texture acts as a virtual screen that content can be written to. See the [PreComp topic](#) for more information.
Creating textures

To create a texture:

1. Open a Bitmap layer, a Spinbitmap layer or any other texture based layer.
2. Left-click `palette` from the layer editor.
3. Left-click `new texture` to highlight the text field.
4. Type in the name of your texture, for example `theatre surface`.
5. Hit `Enter`. This will open a menu of five different types of textures.

Exporting textures

To Export a texture:

1. Open your texture from the `DxTexture` editor by right clicking on it.
2. Left-click the `Actions` tab at the bottom of the editor.
3. Alter the name if needed, and then hit `Enter` or left click OK. The texture will be written to the output folder of the `Project folder`. 