

# DIGITAL PRODUCTION

MAGAZIN FÜR DIGITALE MEDIENPRODUKTION

NOVEMBER | DEZEMBER 06:2020



## Player!

Die besten Tools für  
Wiedergabe & Review

## Tools

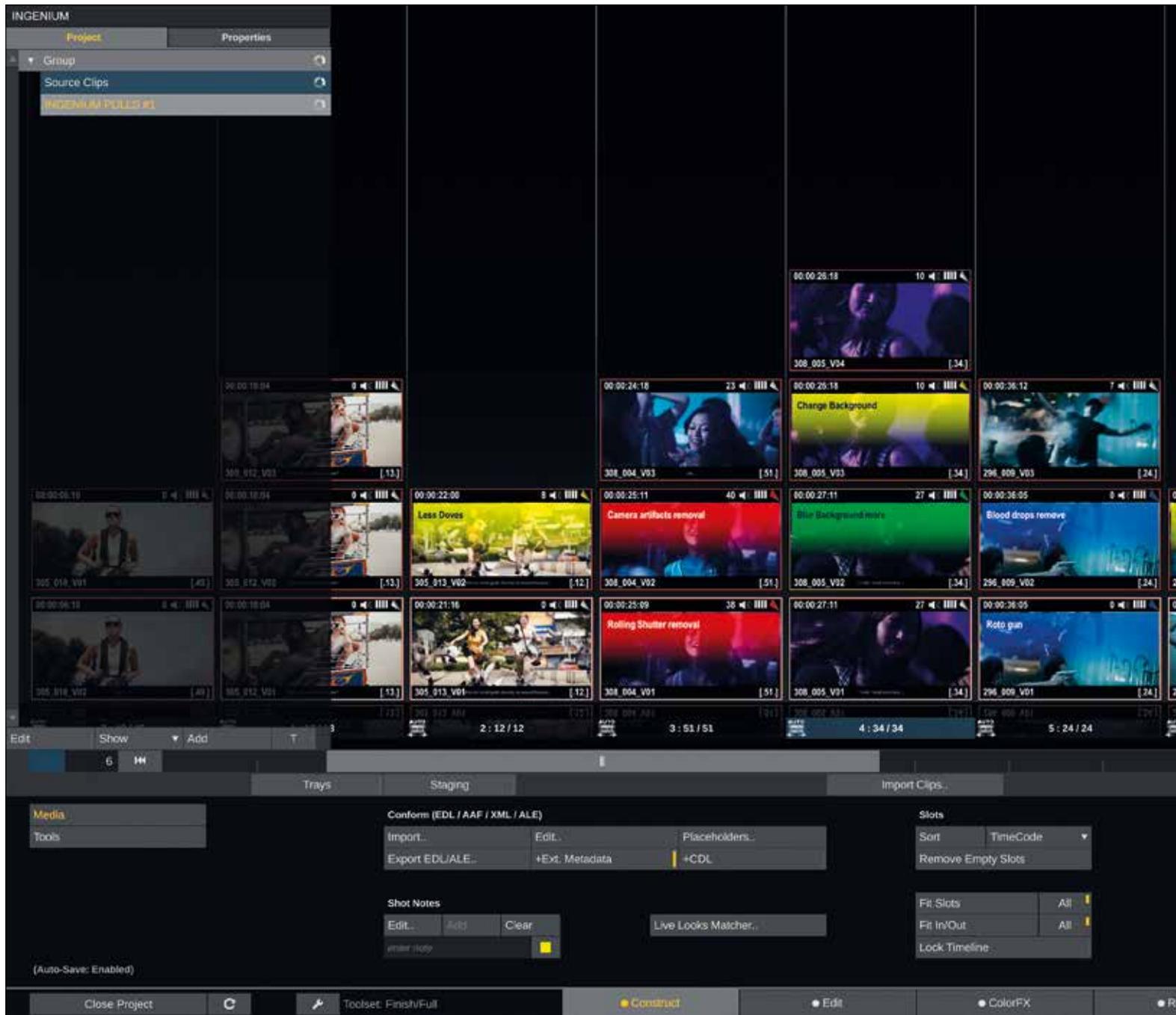
Blender 2.90, V-Ray,  
Arnold, ZBrush, Flame

## Praxis

Westworld S3, Project Blue  
Book, Zeitfeld, LedCaves ...

## und Tests

VFX-Dailies, Apple XDR  
Screen, Streaming



## VFX-Dailies in Assimilate Scratch

VFX-Workflows heutzutage werden immer anspruchsvoller. Es werden mehr und mehr Metadaten direkt am Set während des Drehs erfasst – Metadaten, die später beim Compositing verwendet werden. Im Gegensatz dazu wird das originale Rohmaterial aus verschiedensten Gründen jedoch noch lange nicht in den heutigen VFX-Workflows verwendet. Transcodiert muss es also werden – und am besten mit sämtlichen Metadaten intakt.

von Mazze Aderhold

**W**arum aber wird RAW-footage beim Compositing oft nicht verwendet? Da gibt es mehrere Gründe: RAW-Footage wird nicht immer, oder nur unzureichend, von den gängigen VFX-Tools unterstützt. Darüber hinaus sind die Datenmengen oft sehr hoch – im Compositing wird schließlich selten der gesamte Originalclip hergenommen, sondern nur ein Teil daraus.

Des Weiteren ist Kameramaterial denkbar ungeeignet für den VFX-Austausch, da es im Falle von RAW-Material lediglich von der Kamera erzeugt wird – nicht jedoch von Post-Tools. Optimiert sind die meisten Compositing-Tools für das OpenEXR-format. OpenEXR ist ein Bildsequenzformat, welches Bilddaten nicht nur in 16-, bzw. 32 bit float

speichert, sondern außerdem auch mehr Kanäle als nur RGB + Alpha speichern kann.

In einem EXR lassen sich mehrere sogenannte Layer unterbringen. Jeder Layer kann vier Kanäle speichern. Der erste Layer ist in der Regel der Beautypass: RGBA. Die darauffolgenden Layer speichern je nach Anwendungsfall etwa Depthmaps, UV-Maps, Displacementmaps, Reflectionmaps, Motionvectors, Cryptomattes und mehr – man kann so ziemlich alles darin unterbringen.

Dateigrößen lassen sich mittels verschiedener Kompressionsalgorithmen reduzieren: PIZ, ZIP, ZIPS, DWAA, DWAB und einige andere. Dabei sind die einen auf Bildkompression optimiert, andere auf Playbackspeed. Ein weiterer Vorteil von OpenEXR



Alle Screenshots: INGENIUM (2020, Camera Obscura, Vertrieb: ALIVE)

**Ingenium**

ESTHER MAAS JUDITH ROERSCH ADRIAN TOPOL

**INGENIUM**

Das Footage für diesen Artikel stammt aus „INGENIUM“ – mehr unter

► [www.ingenium-film.com](http://www.ingenium-film.com)

(2020, Camera Obscura, Vertrieb: ALIVE, Limitiertes Mediabook überall erhältlich ab 23.10.2020)

**Construct Tab: Clip-Import, Conform, Asset-Management und Shot Versioning:** Ältere Versionen eines VFX-Shots werden in Slots übereinander gestapelt. Die aktuellste Version liegt immer zuunterst. Rechts der Metadata Stack, der immer die Metadaten des aktuell angewählten Clips anzeigt.

ist, dass sich eine Unmenge an generischen, statischen sowie dynamischen Metadaten unterbringen lässt. Das hilft nicht nur beim Austausch von Shots zwischen Compositing-Tools, Abteilungen oder Posthäusern, sondern ist auch nützlich für Compositing Artists, die die virtuelle Kamera der Compositing-Software mit den framebasierten EXR-Metadaten verlinken – und damit authentisch Fokus- oder Brennweitenverläufe oder Kamerabewegungen nachbilden und steuern können.

## Der Workflow

Wie bekommen wir also das Rohmaterial aus der Kamera möglichst flott und metadaten-

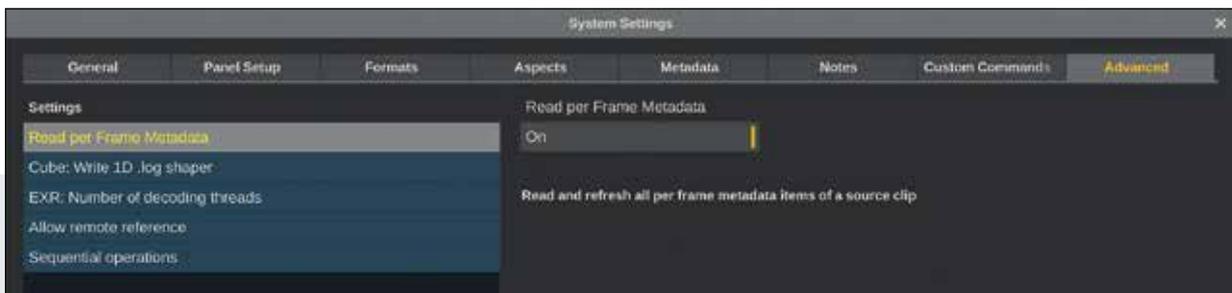
reich ins OpenEXR-Format? Hier kommt Scratch von Assimilate ins Spiel.

Scratch liest nicht nur annähernd alle modernen Kameraformate, sondern auch noch jegliche Metadaten, die da mitgeliefert werden. Für Arri RAW, Arri ProRes MXF und RED RAW werden sogar framebasierte Metadaten unterstützt – also etwa Fokus-, Blenden-, oder Brennweiteninformation, aber auch Schwenk-, Neig-, oder Kippbewegungen der Kamera pro Frame.

Darüber hinaus erlaubt es in seiner Funktion als Finishing-Tool, eine EDL (respektive XML, oder AAF) zu conformen – denn die bestimmt in der Regel, von welchen Shots VFX-Plates erstellt werden. Was ebenfalls in der EDL steht: Der VFX-Shotname.

Dieser unterscheidet sich vom tatsächlichen Dateinamen des Rohmaterials und besteht aus einer festgelegten Dateinamen-Struktur, anhand welcher der VFX-Shot inklusive künftiger Versionen im Shotmanagement getrackt werden kann. Der VFX-Shotname für jeden Sourceclip wird in Form eines Markers (oder für die älteren Semester: Locator) innerhalb der EDL übertragen. Das heißt für unseren Scratch-Workflow: Wir importieren und conformen das Rohmaterial anhand der tatsächlichen Dateinamen, exportieren es jedoch dank Metadatenmanagement mit neuem VFX-Shotname.

Warum man in diesem Workflow eher auf das doch sehr betagte EDL-Format, statt der moderneren XML- oder AAF-Formate setzt? Eigentlich ganz einfach: VFX-Plates spiegeln in der Regel ohnehin das reine Aufnahmeformat wieder – also ohne irgendwelche Effekte, Pan&Scan oder ähnlichem Zeug, was während des Schnitts auf die Offlineclips gepackt wurde. Das heißt, der eigentliche Vorteil von XML und AAF, nämlich gegenüber einer EDL alle möglichen Arten von Effekten zu transportieren, ist damit belanglos. Darüber hinaus ist AAF ein Binär-Format – das kann man nicht mit einem Texteditor öffnen und anschauen. Eine EDL ist noch ein ganzes Stück simpler und lässt sich aus allen gängigen NLEs exportieren, mit einem Texteditor öffnen und damit wunderbar in Scripts und dergleichen nutzen.



Über das Search-field unterhalb der Settings-Liste lässt sich das gesuchte Setting schnell finden.



Project Settings: Die Einstellungen zu Auflösung, Framerate und Default Output Data Format sind nur Defaults für neue Timelines, lassen sich aber jederzeit anpassen – genauso wie die eingetragenen Pfade.



Im Conform Window sind alle Events der EDL aufgelistet. Die Spalte ganz rechts in der Liste zeigt an, unter welchen Kriterien Matches gefunden wurden. Direkt daneben werden die Matches dann aufgelistet.

## Und los geht's – Projekterstellung

Wer eine Komplettübersicht über Daillies Workflows in Scratch haben will, der sollte sich vielleicht zunächst den gleichnamigen Artikel in der DP 01:20 zu Gemüte führen, da dieser den Daillies-Workflow etwas umfangreicher beleuchtet (inkl. Metadaten Burn-in, Audio-Syncing, LUT-/CDL-Matching und dergleichen), während wir uns hier primär auf VFX-Daillies beschränken. Aber fangen wir an:

Als erstes aktivieren wir ein Setting in Scratch unter System Settings im „Advanced“ Tab: „Read per Frame Metadata“ – natürlich nur, wenn wir auch Frame-basierte Metadaten auslesen wollen. Die Funktion hat einen leichten Einfluss auf die Transcoding Geschwindigkeit, weshalb sie standardmäßig deaktiviert ist.

Dann aktivieren wir noch ein zweites Setting: „Disable Color management“. Scratch flaggt jeden Shot, basierend auf seinen Metadaten und/oder Debayering-Settings mit einem Color Space und einer EOTF

(Transferfunktion). Bei Arri RAW wäre das zum Beispiel „Alexa Wide Gamut“ und „Arri LogC“. Gleichzeitig flaggt Scratch auch die Monitorausgänge (UI-Monitor, Dual Head direkt an der GPU und etwaige SDI-Ausgänge) – standardmäßig mit sRGB und Gamma 2.2. Schaut man sich nun einen LogC-geflaggten Clip an, konvertiert Scratch das Bild automatisch nach Gamma 2.2, um es „korrekt“ auf dem UI-Monitor (und allen anderen Monitoren) anzuzeigen. Ähnliches passiert beim Rendering: Jedes Output Node hat ebenfalls ein Color Space und EOTF

Dropdown. Basierend darauf geschieht unter Umständen eine Wandlung beim Export – vom Clip (Alexa Wide Gamut/LogC) zum Output Node (z.B. Rec709/Gamma 2.4). Das lässt sich alles natürlich kontrolliert einstellen, aber um die Sache zu vereinfachen, deaktivieren wir das Color Management hier und vermeiden so ungewollte Farbraumwandlungen.

Als nächstes erstellen wir unser Projekt: Dort ist es wichtig, Media- und Render Directory richtig anzugeben. Im Media Directory sollte sich nach Möglichkeit das gesamte Source Footage befinden, welches importiert werden soll. Selbstverständlich kann man von jedem x-beliebigen Ort Footage importieren, jedoch lässt sich besagtes Footage besser und schneller relinken, wenn es innerhalb des Media Directorys liegt. Für den Fall, dass das Media Directory mal verschoben wird, oder hinter einem anderen Laufwerksbuchstaben verschwindet. Settings wie Auflösung, Framerate und Color Space sollten auf das primär gewünschte Exportformat eingestellt werden – ebenso das Default-Dateiformat: 16 oder 32 bit float OpenEXR. Alle diese Settings sind jedoch lediglich Defaults für neu erstellte Timelines. Scratch erlaubt es, Timeline Settings (und auch die Projektsettings, die den Default bestimmen) jederzeit zu ändern und auch mehrere Timelines unterschiedlichen Formats in einem Projekt unterzubringen. Wir bestätigen mit einem Klick auf OK und öffnen das Projekt.

## Construct Tab: Conform und Metadatenmanagement

Wir befinden uns zunächst im Construct Tab, dem ersten von vieren am unteren Bildschirmrand. Im Construct Tab findet hauptsächlich Media Management statt: Also Import, Conform, Organisation. Und damit geht's auch gleich los – wir importieren das Footage. Dazu können wir am linken Bildschirmrand über den Project Tree Groups erstellen, und darin wiederum Timelines. Über einen Klick auf den „Import Clips..“-Knopf importieren wir das Rohmaterial entweder in die ausgewählte Timeline, oder können das Ganze auch über den „New Timeline“-Button automatisiert laden und von Scratch die Groups und Timelines erstellen lassen, basierend auf den Ordnernamen. Sobald das Footage in einer Timeline ist, sollten wir kurz gegenchecken, ob unser Timelineformat (Auflösung, Aspect, Framerate) dem der Sourceclips entspricht. Das Timelineformat finden wir unten rechts am Bildrand – das Format der Sourceclips im Metadata-Stück ganz rechts. Sollte dort eine Differenz bestehen, kann man ganz einfach einen Clip von der Timeline draggen und auf die Time-

```

FROM_EDITOR.edl - Editor
Datei Bearbeiten Format Ansicht Hilfe
TITLE: Ingenium VFX-Pull1s
FCM: NON-DROP FRAME
000001 AR61C003_180302_R72A V C 09:07:48:23 09:07:56:24 01:04:48:04 01:04:56:05
+ just a comment
*LOC: 01:04:51:24 BLACK 305_010
000002 AR62C004_180302_R72A V C 11:34:20:15 11:34:23:07 01:04:56:05 01:04:58:22
*LOC: 01:04:58:01 RED 305_020
000003 AR61C011_180302_R72A V C 10:18:02:08 10:18:05:05 01:04:58:22 01:05:01:19
*LOC: 01:05:00:02 BLUE 305_060
000004 AR61C007_180302_R72A V C 09:56:01:14 09:56:05:22 01:05:01:19 01:05:06:02
*LOC: 01:05:03:21 BLACK 309_020
000005 AR61C014_180302_R72A V C 10:32:23:08 10:32:24:07 01:05:06:02 01:05:07:01
*LOC: 01:05:06:12 BLACK 309_030
  
```

Eine EDL lässt sich einfach in einem Texteditor öffnen und editieren. Dabei kann man sowohl den Schnitt selbst, als auch Inhalte von Comments und Markern anpassen.

lineformat-Anzeige unten rechts dropfen, damit die Timeline fortan dasselbe Format hat, wie die darin liegenden Clips. Als nächstes importieren wir die EDL vom Offline-Edit – das geht über den „Import..“-Button im Conform-Bereich. Im darauffolgenden Conform Window sind dann alle sogenannten Events in der EDL aufgelistet. Links unten kann man Scratch mitteilen, welche Framerate die EDL hat und ob ein Timecode-Offset berücksichtigt werden soll.

Rechts daneben wählt man die Match-Kriterien aus und wo im Projekt sich Scratch die Clips suchen soll: Aktuelle Timeline, Group, oder im ganzen Projekt. Das ist auch der Grund, warum wir das Footage bereits vorab in seiner Gesamtheit geladen haben: Für EDLs, die keinerlei Dateinamenangaben besitzen, ist das zwingend erforderlich. Bei AAFs und XMLs kann beim Conforming alternativ auch auf einen Ordner verwiesen werden, ohne vorher das Footage ins Projekt laden zu müssen.

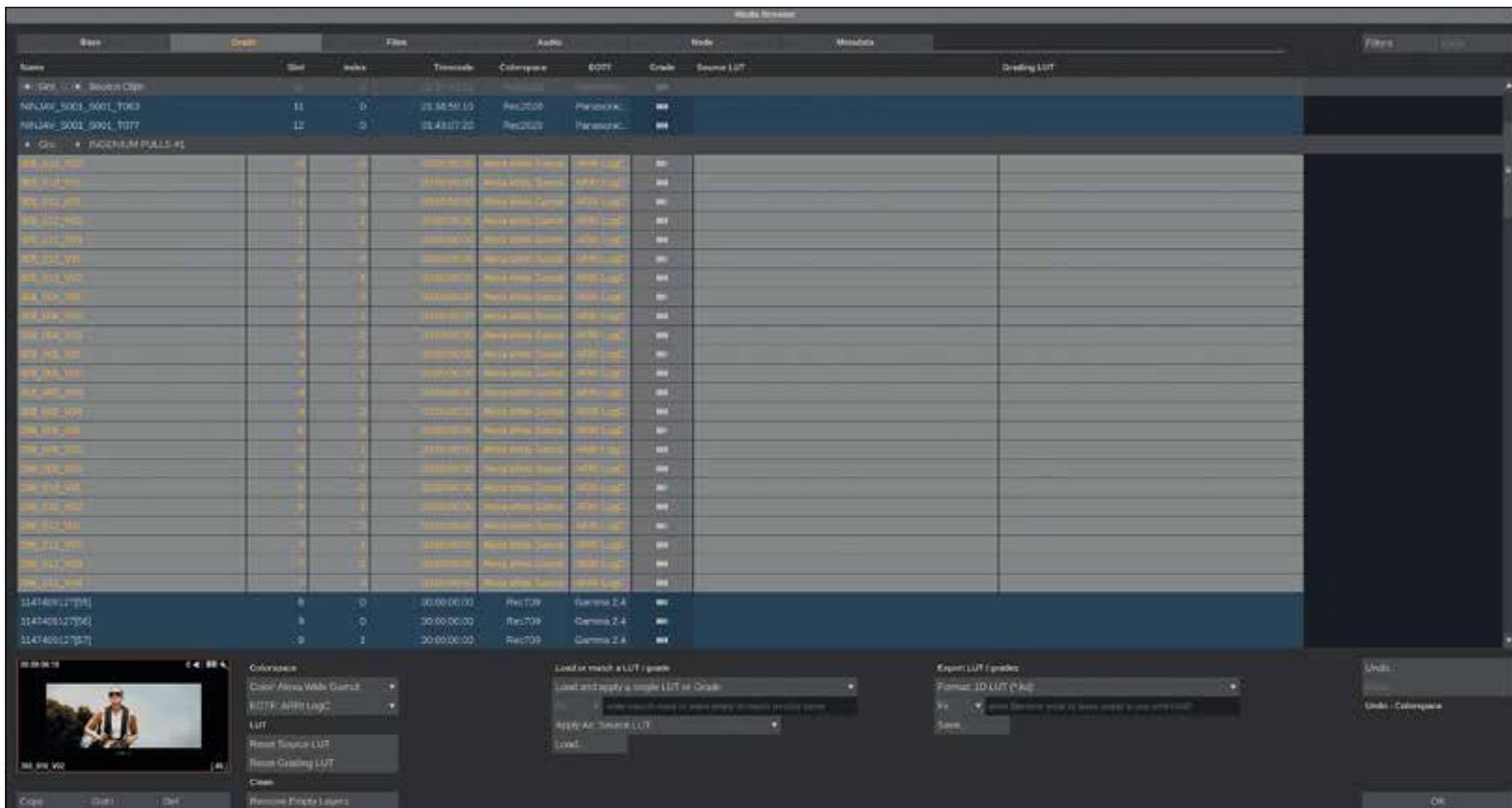
Für die Match-Kriterien reicht es in der Regel aus, Timecode und Clipname anzuwählen – aber das ist vom jeweiligen Projekt (oder Cutter/Schnittassistent) abhängig. Sobald die Match-Kriterien ausgewählt sind, kann man rechts daneben den Matching-Prozess starten. Sobald der durch ist, werden die gefundenen Matches für jedes Event rechts in der Spalte aufgelistet. Hier kann man auch ungewollte Matches entfernen, falls nötig. Das zuunterst liegende Match ist der primäre Clip, der zum Conformen der Timeline hergenommen wird. Alle darüber liegenden Clips werden lediglich als Versions über dem Primärclip abgelegt – und das auch nur, wenn darunter der „Incl. Duplicates“-Button aktiviert ist. So behält man sich die Möglichkeit vor, während dem anschließenden Conformcheck doch noch das primäre

Match gegen ein anderes auszutauschen. Bevor wir nun jedoch vorschnell auf den „Assemble“-Button klicken, sollten wir schnell noch über die wichtigsten Settings gehen: „To New Timeline“ sollten wir aktivieren, damit nicht unsere bestehende Timeline, die lediglich als Source-Clip Speicher dient, verbogen wird. Ebenfalls sollten wir „Use Placeholder“ aktivieren – sollte mal ein Event nicht mit einem Clip gematcht werden, erstellt Scratch hier einen Placeholderclip, der sich auch im Nachhinein noch mit einem Sourceclip linken lässt. Direkt daneben der „Apply CDL“-Button: Enthält die EDL vielleicht CDL-Grades für jeden Clip? Und wenn ja, sollen diese Grades in Scratch direkt nachgebaut und angewandt werden? Die wirklich wichtigen Settings sind aber die folgenden:

Sollen Comments importiert werden? Comments sind in der EDL schlicht Einträge, die mit einem Sternchen beginnen, gefolgt vom eigentlichen Comment. Das kann alles Mögliche sein – wird aber oft nicht für den VFX-Workflow benötigt. Als nächstes: „Import Markers“. Ein Marker findet sich in einer EDL als Sternchen mit darauffolgendem „LOC“, Timecodeangabe, Farbe und Inhalt. Eben was einen Marker (oder „Locator“) im Avid so ausmacht:

```
*LOC: 01:04:51:24 BLACK 305_010
```

Scratch selbst unterscheidet bei Comments und Markern nicht: Beide enden als sogenannte „Notes“ in ihrer Farbe (Comments sind immer gelb) auf dem jeweiligen Shot. Aber selbst Marker wollen wir nicht zwingend als Notes auf unseren Shots haben – denn da helfen sie uns nicht wirklich. Etwas weiter links findet sich ein Dropdown, welches standardmäßig „Upd. Name: None“ zeigt.



Später beim VFX-Review mit dem Kunden ist es oft notwendig, LUTs und CDLs batchmäßig auf Clips zu laden und gegebenenfalls wieder zu exportieren. Das geht hier im Media Browser.

Das stellen wir um auf „Upd. Name: Markers“. Was jetzt passiert, wenn wir den „Assemble“-Button klicken, ist, dass Scratch sich den VFX-Shotname aus dem jeweiligen Marker für jeden Clip zieht und als Clipname verwendet.

Damit sind unsere Clips also direkt nach dem Conform schon umbenannt und im Grunde bereit, wieder als EXR exportiert zu werden. Bevor es aber soweit ist, sollten wir noch schnell einen Blick in das mächtigste Tool in Scratch werfen: Den Media Browser. Der Button dorthin versteckt sich schüchtern am rechten Bildschirmrand. Erst wählen wir jedoch mit **Ctrl + A** alle Clips in der Timeline an, und drücken dann auf den Media Browser Button.

Im Media Browser selbst gibt es verschiedene Tabs: Im Base Tab kann man Clipnames, Reel-IDs, Timecodes und mehr anpassen. Bei nicht-numerischen Metadaten, wie etwa Reel-ID und Clipname kann man sich die in Scratch verfügbaren #-Codes zunutze machen und die jeweiligen Felder damit füllen. Mehr zu den #-Codes, sobald wir beim Exportieren sind. Das Grade Tab erlaubt es uns, batchweise LUTs und/oder CDLs auf alle angewählten Shots zu laden und ggf. auch wieder davon zu exportieren.

Dabei können wir sogar LUT-Matching betreiben, indem wir statt „Load and apply a single LUT or Grade“ die „Match on shot name or search mask“-Funktion wählen. Jetzt können wir Scratch beibringen, wie es die Dateinamen der zu ladenden LUTs zu interpretieren hat, um sie auf die jeweiligen Shots zu matchen: Etwa #sname (wenn

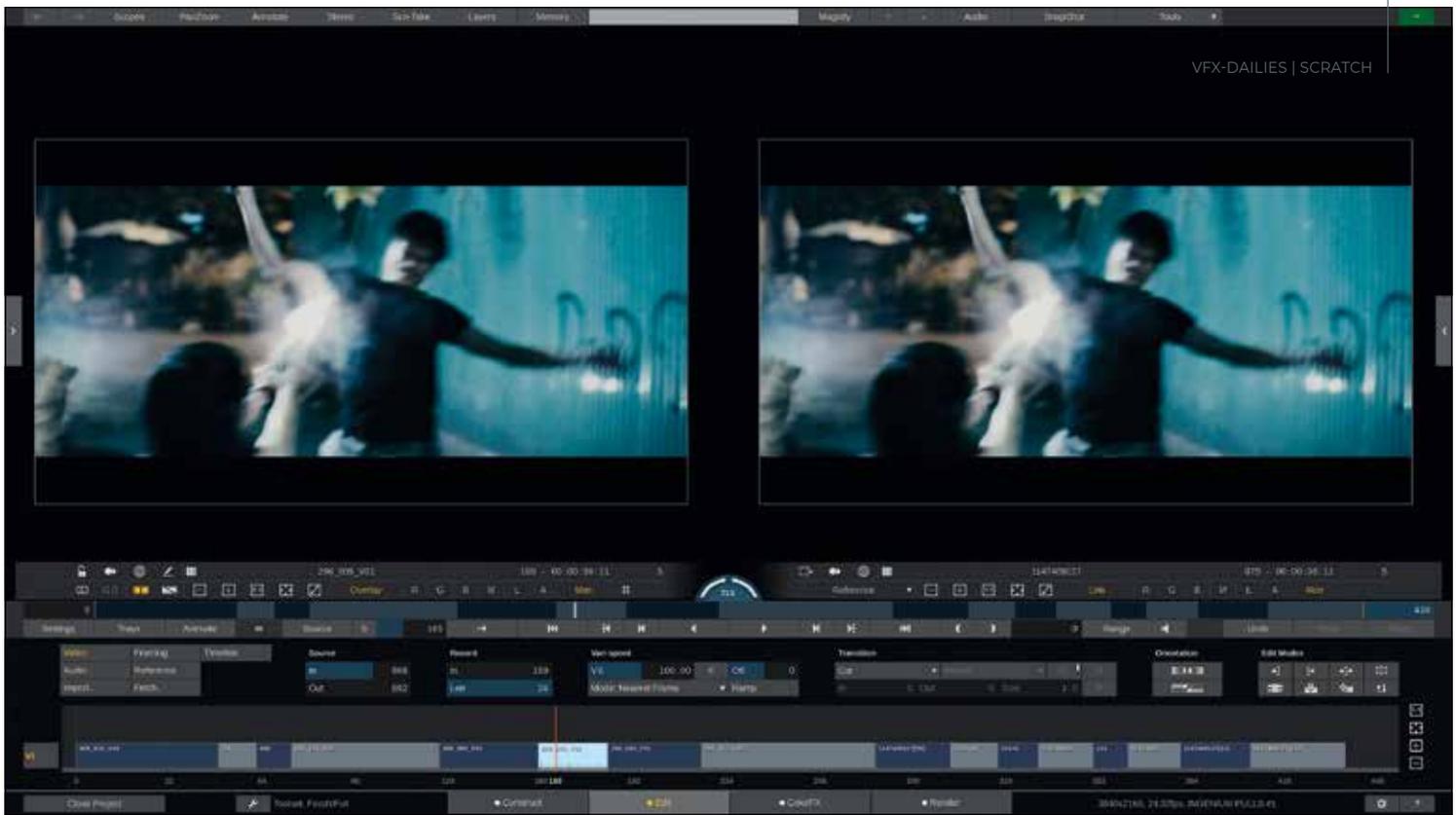
der Dateiname der LUT dem VFX-Shotname entspricht) oder #scene (wenn die Dateinamen der LUTs szenenbasiert sind) – oder auch eine Kombination mehrerer #-Codes. Diese Funktion ist jedoch in erster Linie hilfreich, wenn die fertigen Composites aus der VFX zurückkommen und dann automatisch Grades auf die Clips gepackt werden, um sie mit dem Kunden zusammen beurteilen zu können. Im Files-Tab können Shots, Audio-dateien und LUTs relinked werden, falls mal was offline geht und im Audio Tab kann extern aufgenommenes Audio gesynct werden. Da das eher unüblich bei VFX-Workflows ist, gehen wir an dieser Stelle nicht näher darauf ein. Im Node Tab können die Decoding- bzw. Debayering-Settings für alle angewählten Clips eingestellt werden. Bei ProRes Quicktime gibt’s da wenig bis gar nichts zu tun – wenn aber das Sourceformat Arri RAW, RED RAW oder irgendein anderes RAW-format ist, hat man hier die Möglichkeit, den Farb-raum, Gamma, sowie die Debayeringqualität und, abhängig vom jeweiligen RAW-Format, eine Reihe anderer Settings einzustellen. An dieser Stelle sei angemerkt, dass die kommende Version 9.3 mit einstellbaren Projekt-defaults für alle RAW-Formate ausgeliefert wird, sodass man sich damit nicht mehr nach jedem Import herumschlagen muss. Bleibt zuletzt noch das Metadata Tab. Hier können wir Clip-Metadaten modifizieren, löschen, oder hinzufügen. In der unteren Hälfte des Media Browsers lassen wir uns die für uns interessanten Spalten anzeigen, oder erstellen und füllen zur Linken davon vier #-Codes. Soviel zum Media Browser.

## Edit Tab: Conform überprüfen

Bevor es an den Export geht, schauen wir noch schnell ins Edit Tab: Falls Timewarps auf irgendwelchen Shots liegen, die wir natürlich nicht in unseren VFX-Plates wollen, können wir diese im Video Menü entfernen. Genauso können wir auch Dissolves und andere Transitions entfernen, falls nötig – in der Regel wurde aber beides schon im Schnittsystem vor dem EDL-Export von der Schnittassistentin erledigt, sodass wir uns darum im Conform nicht mehr kümmern müssen. Im Reference Menü können wir einen Referenz-Clip laden, um unseren Conform zu überprüfen und sicherzustellen, dass alles ordnungsgemäß in unserer Timeline liegt.

Beim Laden des Referenzclips springt Scratch automatisch in den Dual-View Modus. Diesen kann man auch händisch über Hotkey „D“ triggern. Alternativ kann man in den Split-View Modus via Hotkey „S“ wechseln, um den Conform zu überprüfen. In der Mitte, genau zwischen den beiden Views, lässt sich ein Offset zwischen Timeline und Referenz einstellen, falls nötig. Im Version Stack auf der rechten Seite finden wir die Duplicates – also alternative Shot-Matches vom Conform. Der primäre Clip, der auch für den Export verwendet werden wird, liegt immer zuunterst. Sollten wir der Ansicht sein, dass ein anderes Match besser passt, draggen wir es einfach von seiner jeweiligen Position auf die unterst-mögliche.

Man kann im Übrigen in jedem der Thumbnails durch den Clip scrubben, indem man an der oberen Kante draggt.



Im Edit Tab prüfen wir den Conform anhand eines Referenz-Clips im Dual View Modus. Hier können wir ggf. Timewarps oder Transitions entfernen, sollten die noch in der EDL enthalten gewesen sein.



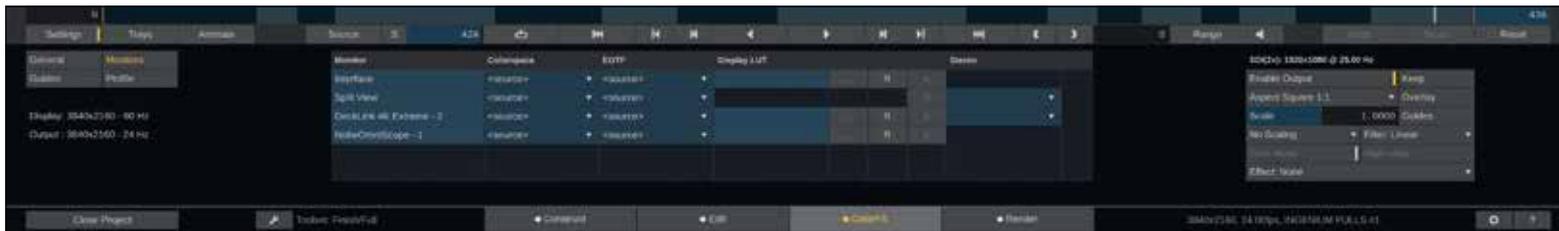
Scratch erlaubt es, beliebig im Viewport zu zoomen und Elemente wie die Scopes anzuordnen. Diese helfen bei der technischen Beurteilung und der Metadata Stack rechts listet alle relevanten Metadaten.

### ColorFX Tab: Grades, Metadaten und Bild überprüfen

Weiter zum ColorFX Tab: Hier können wir im Numeric Menü etwaige CDL-Grades gegenchecken – rechterseits können wir vom Version Stack zum Metadata Stack wechseln,

um uns die Metadaten eines jeden Clips anzeigen zu lassen und über die Top Menu Bar am oberen Bildschirmrand können wir uns Scopes einblenden, um das Bild weiter technisch zu beurteilen. Dazu noch schnell die wichtigsten Hotkeys: Alt + Drag up/down, um im Viewport zu zoomen und Space +

Drag, um das Bild im Viewport zu verschieben. Ebenfalls in der Top Menu Bar finden wir einen Magnifier, mit dem man Pixel-Values genau untersuchen kann. Der Color Pot in der Mitte der Top Menu Bar zeigt diese dann an. Mit einem Klick in diesen Color Pot kann man einstellen, ob man die Werte in



Wer sich im Color Management austoben möchte, dem sei geraten, Scratch im Monitor Menü mitzuteilen, was seine Monitore am Eingang erwarten. „Source“ bedeutet, dass keine Wandlung stattfindet und Clips immer 1:1 angezeigt werden.

8 bit, 10 bit, 16 bit oder in floating point angezeigt bekommen möchte.

Für den Fall, dass das Bild insgesamt zu kontrastig erscheint, obwohl man doch mit logarithmischem Footage arbeitet, hier ein Hinweis: Das passiert, wenn man das Color Management nicht vorher deaktiviert hat. Wer jedoch Color Management nutzen will, der sollte es dann auch ordnungsgemäß aufsetzen – fürs Monitoring passiert das unter Settings (Hotkey F5) im Monitor Menü. Hier sind alle Monitor-Ausgänge aufgelistet. Stellen wir den UI-Monitor zum Beispiel auf „Source“, findet keine Farbraumwandlung für diesen Monitor statt.

Haben wir einen Projektor am SDI-Ausgang hängen, der P3 D65 mit einem Gam-

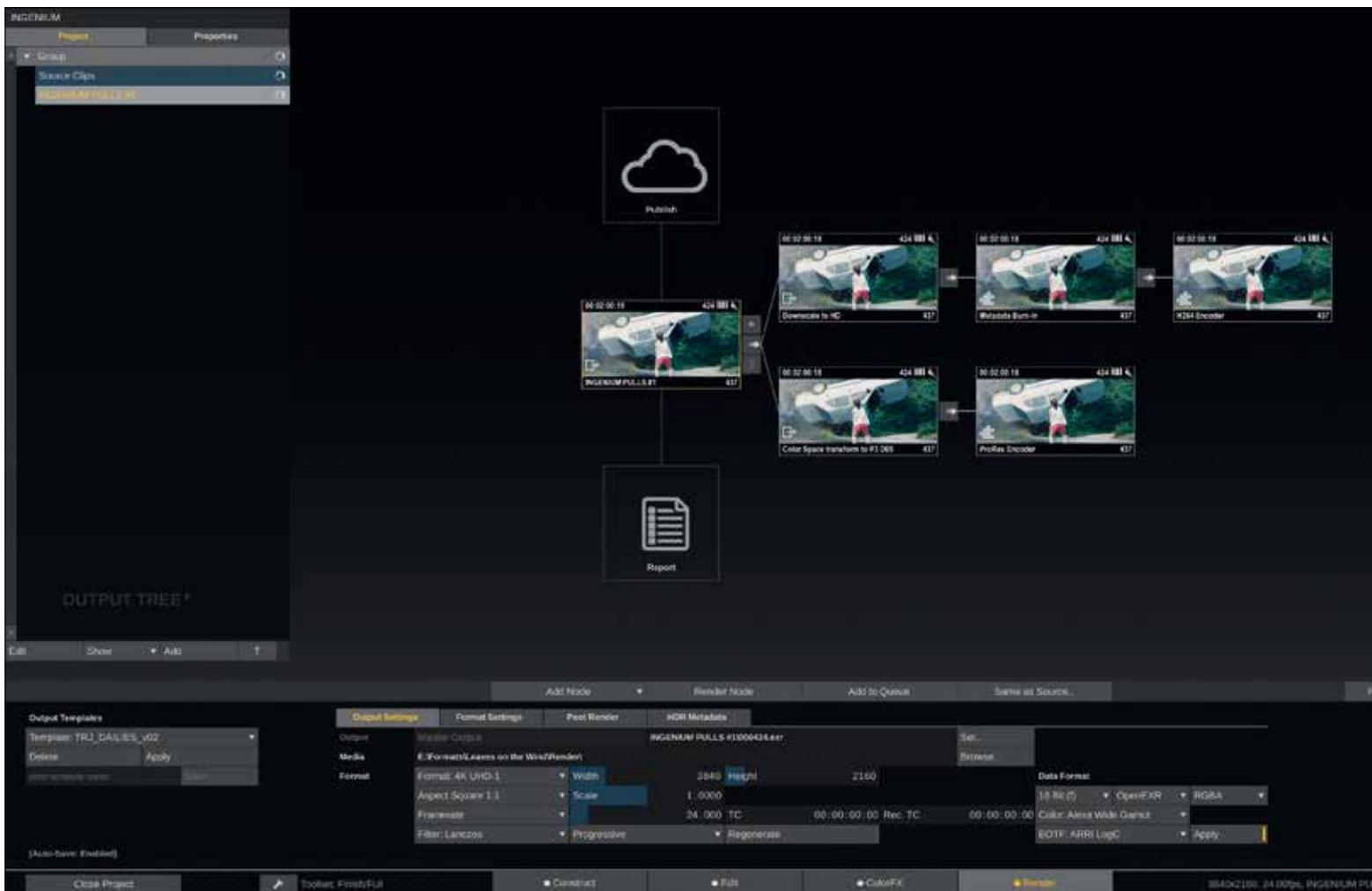
ma von 2.6 erwartet, können wir das hier für den jeweiligen SDI-Ausgang einstellen. Es wird immer derjenige Clip umgewandelt, auf dem wir gerade stehen. Der momentan angenommene Farbraum des Clips lässt sich entweder über den Metadata Stack, oder das Media Menü unter ColorFX herausfinden. Wenn Color Management deaktiviert ist, so wie Eingangs gezeigt, sollten alle Monitore standardmäßig auf „Source“ stehen.

### Render Tab: Output Tree bauen und exportieren

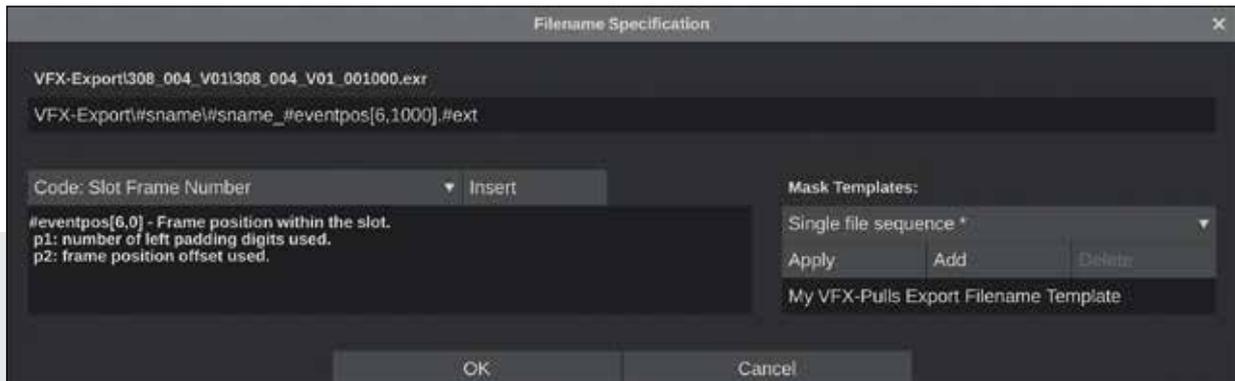
Nun, da Conform, Metadata Management und Bild QC durch sind, geht es ans Exportieren. Im Render Tab können wir uns mit-

tels Output Nodes verschiedenster Art einen Output Tree anlegen. Den kann man sich in etwa so vorstellen wie Timeline Nodes in Resolve – allerdings mit der zusätzlichen Funktionalität, dass es auch noch Encoder Nodes zum File-Export gibt.

Zu Beginn besteht der Output Tree lediglich aus dem sogenannten Main Output Node. Das repräsentiert im Übrigen unsere Timeline bzgl. Auflösung, Aspect und Framerate. Wenn wir an diesen Parametern auf dem Main Output Node etwas verändern, verändern wir damit unsere Timeline-Parameter. Gehen wir davon aus, dass unser Source Footage UHD-Auflösung hat und unsere Timeline dem entspricht. Wir können nun das Main Output Node



Im Render Tab können wir uns unseren Output Tree mit beliebig vielen Abzweigungen anlegen, um Deliverables in verschiedenen Auflösungen, Farbräumen und Formaten zu exportieren.



Im Filename Specification Menü können wir den Export Dateinamen dynamisch anhand von #-Codes anlegen. Direkt darüber wird uns praktischerweise eine Preview der Filemask angezeigt.

direkt als EXR-Sequenz exportieren – oder vorher noch ein wenig auf die Möglichkeiten des Output Trees eingehen. Das kleine Plus-Icon rechts vom Main Output Node erlaubt es uns nämlich, z.B. ein Transformer Node anzuhängen.

Ein Transformer Node kann beispielsweise die Auflösung verändern (etwa um mit einem anschließenden H264/H265 Encoder Node kleine H264 Preview Clips in HD zu rendern), oder das Material in einen anderen Farbraum wandeln.

Jedes Node hat ein eigenes Format Settings Tab, in welchem die Node-spezifischen Settings eingestellt werden müssen. Bei einem Image Node ist es etwa das Input-Sizing, bei einem ProRes Encoder Node wäre es die Codec-Auswahl und Audio Exportkonfiguration.

Ebenfalls nützlich für Preview Clips: Das Burn-in Node. Sobald es hinzugefügt wurde, kann man mit einem Doppelklick auf das Node ins ColorFX Tab wechseln (wichtig: Um Output Nodes im Player anschauen und verändern zu können, muss man sie zwingend doppelklicken, um damit ins ColorFX Tab zu gelangen). Dort kann man es über das Burn-in Menü konfigurieren – mit

den schon bekannten #-Codes. Damit lassen sich dynamisch Timecodes, Datum, Clipname, Auflösung und so weiter einblenden. Wieder zurück zum Output Tree: Hier können wir mehrere Zweige im Tree anlegen für verschiedene Deliverables. Am Ende eines jeden Zweigs sollte dann ein Image- oder Encoder Node stehen, der den eigentlichen File-Export übernimmt.

In der Regel wollen wir aber unverfälschte Plates in Originalauflösung rendern – also zurück zum Main Output Node. Auflösung, Framerate und Aspect sollten bereits passen und der Originalauflösung entsprechen. Auch das Exportformat 16/32 bit float OpenEXR sollte sich aufgrund unserer Project Settings schon korrekt eingestellt haben.

Wenn wir vom Output Settings Tab ins Format Settings Tab wechseln, können wir noch den Kompressionsalgorithmus festlegen, den wir beim Exportieren für die EXRs verwenden wollen.

Zurück im Output Settings Tab fehlen jetzt noch zwei Dinge: Export Filename und Pfad.

Der Hauptrender-Pfad ist entsprechend den Project Settings eingestellt, kann aber auf Wunsch auch woanders hin umgebogen werden. Es empfiehlt sich jedoch, das dann besser in den Project Settings zu ändern, damit es standardmäßig für jede neu erstellte Timeline so aufgesetzt wird. Der aktuelle Filename für den Export wird oberhalb des Pfads angezeigt und entspricht zunächst einmal nicht dem, was wir uns für unsere VFX-Plates vorgestellt haben. Mit einem Klick auf den Filename kommen wir ins Filename Specification Menü.

In der Regel strahlt uns jetzt die folgende Filemask an: #name/#frame[6].#ext. Da wird also zunächst der Name des aktuellen Nodes verwendet (was in diesem Falle dem Timeline-Namen entspricht), um einen Ordner gleichen Namens zu erstellen – und dort wird dann die EXR-Sequenz mit einem simplen 6-stelligen Framecount gerendert. Sogas kann natürlich kein Mensch gebrau-

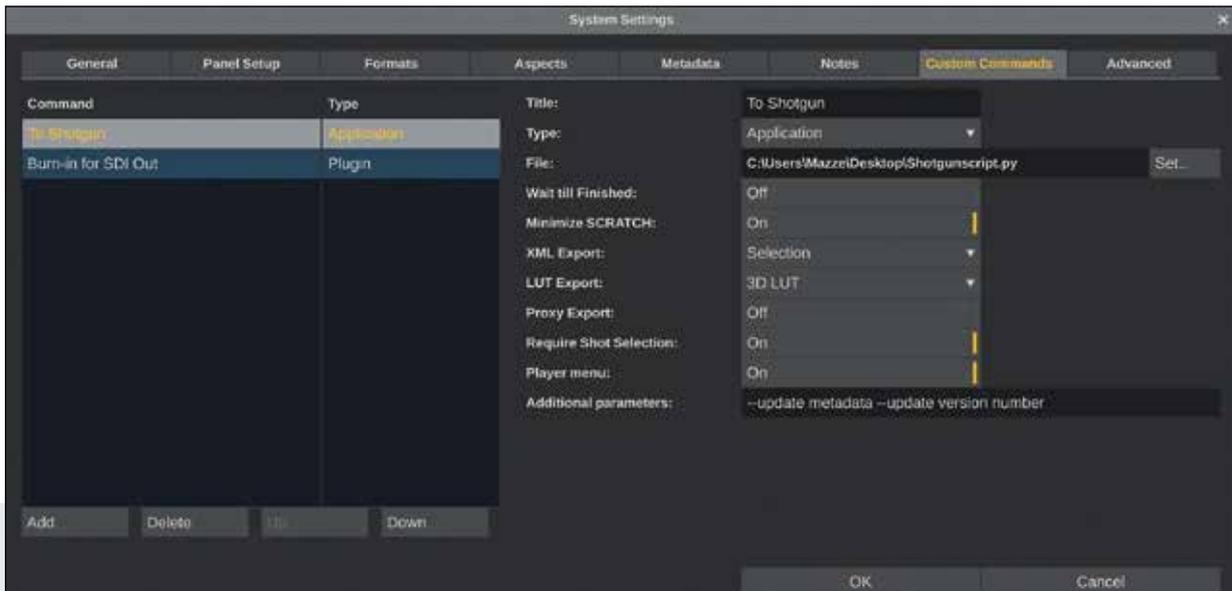
chen – nicht nur weil der Filename recht aussagelos ist, sondern auch weil Scratch in diesem Falle lediglich eine lange EXR-Sequenz rendern würde, statt einzelne Shot-Sequenzen.

Wie bringen wir Scratch nun also dazu, einzelne Shot-Sequenzen zu rendern? Ganz einfach: Die Filemask selbst diktiert es. Im Moment besteht die Filemask lediglich aus dem Namen der Timeline (der für alle Shots gleich ist) und dem Framecount (der notwendig ist, denn sonst erhielte ja jeder EXR Frame den exakt selben Dateinamen und würde den vorher geschriebenen Frame direkt überschreiben). Sobald wir jedoch ein Kriterium (sprich: #-Code) einbringen, was sich von Shot zu Shot verändert – etwa den Clipname – ändert sich auch der Export-Dateiname während des Renders und so werden dann separate Shot-Sequenzen erzeugt. Ähnlich funktioniert das ganze natürlich auch mit ProRes-, DNx-, oder H264-Rendern – nur dass man dort keine Framecounts verwenden sollte.

Eine sinnvolle Filemask für unseren EXR-Export wäre etwa:

VFX-Export/#sname/#sname\_#eventpos[6,1000].#ext

Hier sagen wir Scratch zunächst ganz simpel, dass es doch bitte einen Ordner mit Namen „VFX-Export“ erzeugen soll. Dort hinein möge es bitte einen Ordner entsprechend dem „Source Name“ (also #sname) erstellen. Der Source Name ist hier der Clipname im Projekt, der tatsächlich den VFX-Shotname aus der EDL widerspiegelt – dieser ändert sich mit jedem Shot und so erstellt Scratch für jeden Shot auch einen neuen Ordner mit dessen VFX-Shotname. In diesen Ordner wird nun die eigentliche EXR-Sequenz gerendert – ebenfalls mit dem VFX-Shotname und zusätzlich einem slot-basierten Framecount (#eventpos – damit startet der Framecount für jeden Slot in der Timeline neu).



Ein Custom Command kann so ziemlich alles sein: Vom Plug-in bis zum Script ist alles möglich. Damit wird Scratch einfach an Nuke, Shotgun oder andere Software angebunden, um einzelne Shots, oder gar komplette Timelines zu übergeben.

Dieser soll maximal 6 Ziffern lang sein und hat einen Offset von 1000 Frames – startet also immer bei Frame 1000.

Für den Fall, dass wir diese Filemask bald wiederverwenden wollen, können wir uns dieselbe hier noch als Template abspeichern. Mit einem Klick auf OK ist alles bestätigt und wir können entweder auf „Render Node“ klicken, oder wahlweise das Node über den „Add to Queue“-Button in die Render Queue packen und noch weitere Output Nodes von diesem Tree, oder anderen Timelines hinzufügen, bevor in der Render Queue auf „Start“ klicken.

Scratch rendert Outputs einer Timeline immer parallel – das ist effizient, da der Source Frame nur einmal gelesen werden muss und dann dynamisch an alle Encoder verteilt wird. Output Nodes verschiedener Timelines werden nacheinander abgearbeitet.

Das Schöne an Scratch: Es rendert im Hintergrund. Wir können in der Render Queue also auf „Continue“ klicken und anderswo im Projekt weiterarbeiten. Was wir auch machen können, um künftige Transcodings zu vereinfachen, ist unseren Output Tree als Template abzuspeichern. Dazu wählen wir zunächst das Main Output Node ganz links an, um dann in der unteren linken Ecke des Bildschirms den Namen des zu speichernden Templates einzugeben und auf „Save“ zu klicken.

Gespeichert wird immer der Tree zur rechten Seite des momentan angewählten Nodes (ohne jedoch das angewählte Node selbst!) – daher sollte man vor dem Speichern das Main Output Node anwählen, um auch wirklich den gesamten Tree zu speichern. Wenn wir das Projekt schnell ver-

lassen (Achtung, beim Verlassen des Projekts stoppt der Render), können wir das eben abgespeicherte Template auch in den Project Settings verankern, damit jede neu erstellte Timeline gleich mit unserem vordefinierten Output Tree daherkommt – inklusive Burn-ins, Resizes, Filemasks und allem was dazugehört. Sobald die fertigen Shots von der VFX zurückkommen, kann man sie nun als Versions in dieselbe Timeline laden und so „before & after“-Screenings mit dem Kunden machen.

Ebenso kann man weiter neue Versionen eines VFX-Shots (auch automatisiert via Scripts) in die Timeline laden und so alle Versionen eines Shots übereinander in einem Slot in der Timeline lagern. Damit hat man – etwa während eines Kundenscreenings – Zugriff auf jede Version eines Shots. Notes, welche man auf jeden Shot legen kann, können ebenfalls als PDF oder XML exportiert werden.

## Automatisierung durch Scripting

Das war nun also der manuelle Weg vom VFX-conform bis zum fertigen Export. Scratch verfügt jedoch zusätzlich noch über eine XML-scripting Schnittstelle, mit der wir vieles von dem, was wir jetzt händisch durchexerziert haben, automatisieren können.

Dazu kann man in den System Settings ein „Global Watch Folder“ angeben. Sobald dort eine XML drin liegt, macht sich Scratch umgehend daran, dieselbe umzusetzen. Das Anwenden und Absenden von Scratch XMLs kann durch sogenannte „Custom Commands“ gelöst werden, die wir in den System Settings anlegen können.

Ein Custom Command ist ein benutzerdefinierter Button, der fast überall im Scratch-UI platziert werden kann. Die Funktion kann vielfältig sein: Ein vordefiniertes Plug-in laden, eine Webpage aufrufen, ein System Event triggern, oder auch eine Anwendung oder eben Script starten – an welches dann eine Scratch-XML übergeben wird, um dieselbe zu parsen. Eine solche Scratch XML kann dann vom jeweiligen Script auch wieder via Watch Folder zurück an Scratch gegeben werden, um Projekte zu erstellen, Footage zu laden, Metadaten hinzuzufügen, CDL- und LUT-Grades automatisch anzuwenden, Transcodes zu starten und dann wieder erneut eine XML zu exportieren, die in weiteren externen Scripts verwendet werden kann – etwa um die Notes vom letzten Kundenscreening ins Shot Tracking System zu übertragen. Scripting in Scratch heben wir uns jedoch für einen anderen Artikel auf. Der geeignete Leser kann die Dokumentation für Scratch-XML inklusive Anwendungsbeispiele und Samplescripts in der Zwischenzeit auf der Supportseite von Assimilate ([www.assimilatesupport.com](http://www.assimilatesupport.com)) finden und sich damit ans Script-Schreiben begeben.

> ei



Mazze Aderhold ist freiberuflicher Application Engineer und Workflow-Spezialist. Er arbeitet hauptsächlich als Head of Support und Product Manager für Assimilate, aber auch direkt mit verschiedenen Posthäusern zusammen. 2015 gründete er Angry Face, eine Firma, bei der sich alles um Hardware für Color Grading dreht. Hierfür hat er sowohl ein Gehäuse (das Rageboard™) sowie hochwertige eloxierte Aluminiumringe für Tangents Element Panel entwickelt, die er weltweit vertreibt.

[www.mazze-aderhold.de](http://www.mazze-aderhold.de)  
[www.angry-face.com](http://www.angry-face.com)