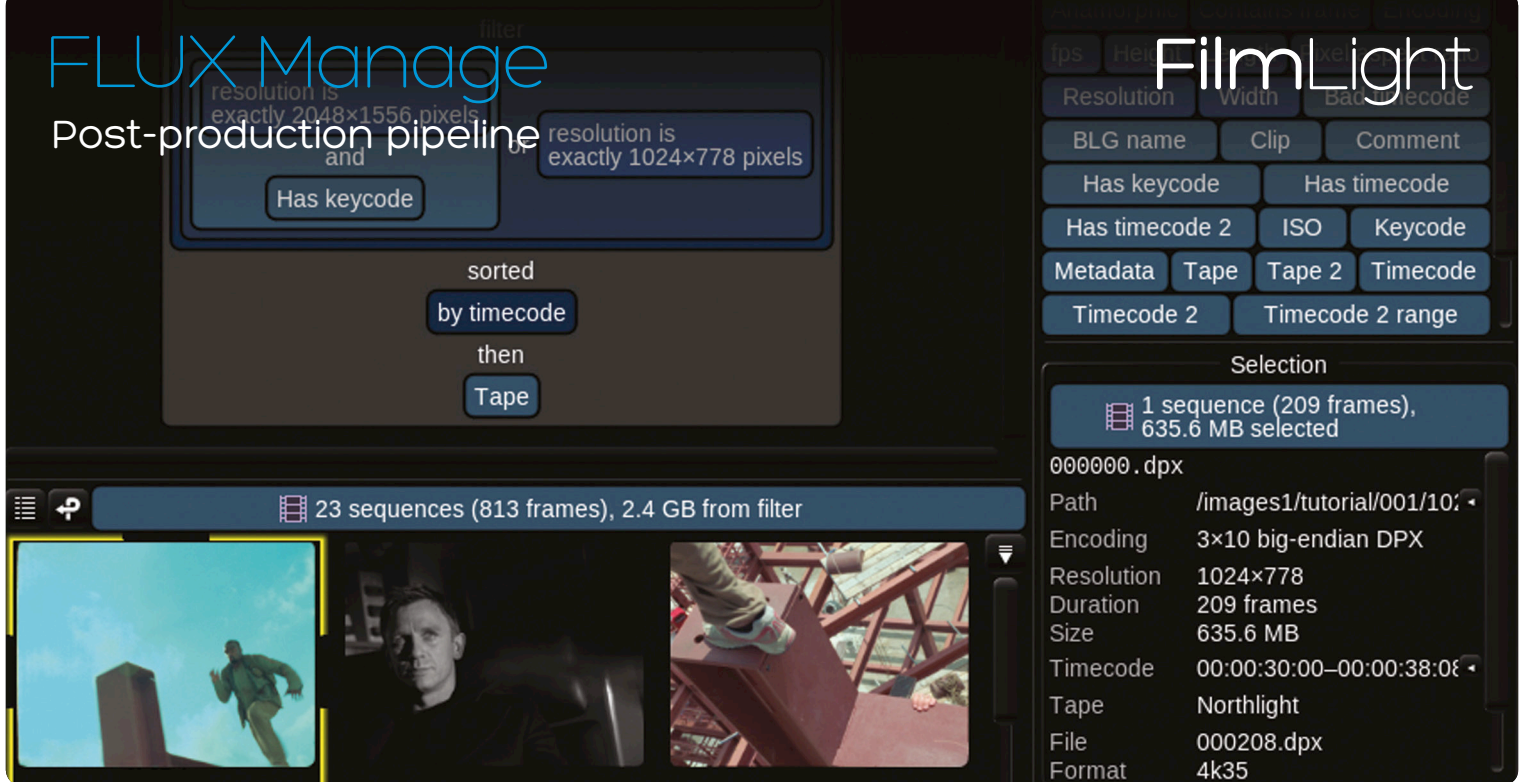# FLUX Manage
Post-production pipeline

FilmLight

Baselight 5.0 introduces a revolutionary new way to build an image factory

A Changed Business

Digital acquisition has changed the world of post-production. There are few practical or economic constraints when it comes to acquiring large quantities of high-resolution material; multi-camera projects, with shoot ratios previously unheard of, are being delivered to facilities with the same productivity expectations as there were for a bunch of VFX shots a few years ago. To be effective, the post house has to become an image factory.

The visual effects, editorial or grading applications are not suitable for this role; nor are the general purpose IT industry tools, such as conventional file browsers, hierarchical storage systems and data deduplication tools. The ad hoc pipeline tools developed for another age will have to be heavily augmented or put aside. Post-production needs its own industrial throughput digital asset system–a system aware of the relationship between image files, sequences and edit lists, capable of tracking the high volume and quick churn of data, and providing the bandwidth for real-time operation.

Sets of Sets

Data management in post-production–above the single shot–is all about sets of data. An EDL is a set of images, while all the versions of all the EDLs in a project are a super-set. An action on those sets (for example, removing everything on the filesystem except what is in these projects) is a set intersection problem. FLUX Manage lets you build and save common set-based operations graphically to move, copy or delete a precise slice of your data store.

Uniquely Standard

There have been many offerings to manage post-production assets, but they have had limited success because they either:

» Use a proprietary filesystem.

» Put all the assets in a database.

» Rely on a loose coupling between the 'metadata view' (interesting to you) and the 'filesystem view' (required for any application access).

FLUX Manage, when coupled with FilmLight's indexed filesystem as part of FLUX Store[†] or any Baselight system, approaches this problem in a unique way. It makes use of a standard filesystem–in this case, a FilmLight version of XFS–but augments it with metadata kept in synchronisation using custom-developed filesystem callbacks. That is to say, FLUX doesn't need the overhead it takes to open a file to find its timecode, ISO setting, or any other metadata within any post-production media.

As a result, you can search large filesystems for matching metadata instantaneously (as in a conform, for example), all without the overhead of traversing the filesystem hierarchy (accelerated search requires FLUX indexing capability*).

But this storage is still a standard filesystem that can be manipulated and managed with all normal command line functions (such as cp, mv or rm) without the metadata view becoming inconsistent. This inability to subvert the metadata view is the key to a uniquely robust structure. You can use it with the same confidence to manipulate a filesystem using image metadata (e.g. timecode), as you would with filesystem metadata (e.g. filename).

[†] See the *FLUX Store* datasheet for more information.

* Items marked with an asterisk (*) will be made available in subsequent Baselight 5.0 point releases.

See What You Have

A conventional file browser is unfit for video and film production. Drilling down directories to find that shot you know is somewhere on one of the systems on the network is grossly inefficient.

This is a visual industry, and a visual browser, combined with a graphical user interface to refine the search, is much more effective. Somewhere on one of your systems, you know there is a short sequence of frames of R3D material shot at ISO 250 on a RED Dragon from over 6 months back. The fastest way to find it is with an interactive and fluid browser that can simply and quickly build a matching gallery of images with associated metadata, and allow you to flick through them and play the likely suspects.
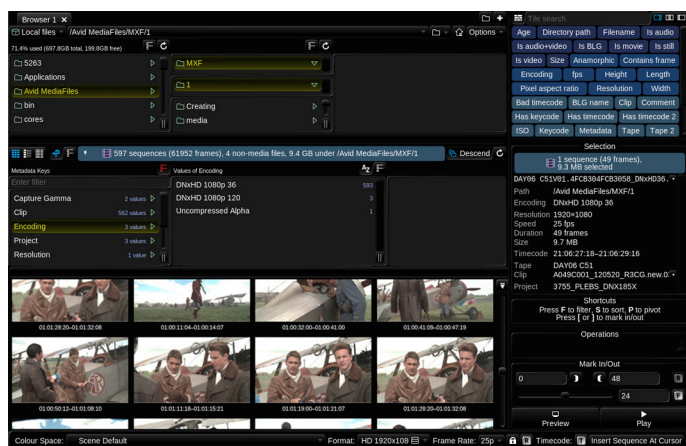
FLUX Manage makes it easy to find just what you're looking for

Difficult Becomes Simple

Error-prone data management tasks that previously relied on in-house scripting tools, or considerable experience with command line shells and languages, become accessible to all.

For example, with FLUX Manage you can easily:

» Move all of the 2K DPX sequences that are longer than 1000 frames, match the name *raw*, and are older than 6 months.

» Copy all of the ARRIRAW files with ISO 800 under a directory.

» Remove everything from the filesystem that is a proxy and is not used in jobs ABC & DEF, leaving ten frame handles in case the edit slips.

» Synchronise every image used in these ten Baselight or Daylight scenes onto another system.

Commonly, some useful tasks are so onerous that you simply don't do them–and that is before taking into account the risk of disruptive finger trouble. Now, you can save these tasks as a template for others on FLUX Manage's shelf, so even facility staff who would find the query itself intimidating can effectively manage the storage themselves.

Make a Difference

FLUX Manage is not a complete facility pipeline. We've planned additional functionality, but many companies have unique, advantageous in-house systems that we could not hope to replicate. Underneath the FLUX Manage graphical user interface is 'fluxc', the flux scripting language executed by the GUI. This allows you to develop a completely custom workflow on top of the FLUX underpinnings if you need to.

There is little point in every boutique, studio, or VFX house developing code to handle the primitive operations, each learning along the way the errors you can make with drop-frame timecode, correctly sorting in frame order the files raw.9999.exr and raw.10000.exr, or that a DPX file can have edge code running backwards if the camera film was loaded in reverse. These and a hundred other mistakes are a waste of time and money to discover in the middle of a high-pressure production.

Instead, the primitive functions–which we all acknowledge should work sensibly in the same way everywhere–can be combined to build any workflow. FilmLight has developed and honed these functions for more than a decade, and they are now available to FLUX users as a foundation to build complex queries and operations. Examples of the fluxc primitives that can be scripted up today are shown in the table on the following page.

These commands can be used to perform common operations, which you can run as a batch job as an alternative to using the GUI.

For example:

» Find all the images used on Baselight 'grading1' in 'joba' with ten frame handles added to the material.

```
FLUX> scenes "grading1:joba" modify addhandles(10,10)
```

» Find all the R3D image sequences under directory /vol/images/jobc longer than 1000 frames, shot with a MYSTERIUM S35 sensor.

```
FLUX> under "/vol/images/jobc" where encoding "R3D" and
longerthan 1000 and meta("Sensor", "MYSTERIUM S35")
```

» Create a set of material not used in 'joba' to remove from the system, keeping ten frame handles in case the edit slips.

```
FLUX> (under "/vol/images") except (scenes "grading1:joba"
modify addhandles(10,10))
```
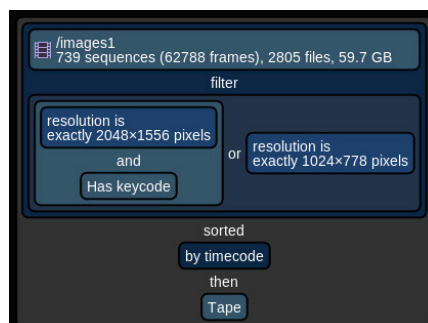

Drag-and-drop filters to build complex queries

fluxc scripting commands

»	+	(alias: plus) Add two sets of sequences and files together, or add two numbers
»	=	Assign to a variable
»	addhandles	Add handles to start and end of each sequence
»	all	(alias: return) Has no effect
»	and	(alias: & &&) True only if all supplied tests are true
»	bydate	Sort by modification date
»	byduration	Sort by sequence duration
»	byencoding	Sort by image encoding
»	bykc	Sort by keycode
»	byname	Sort by filename
»	bypath	Sort by path
»	bytape	Sort by tape name metadata
»	bytc	Sort by timecode
»	bywidth	Sort by image width
»	copy	Copy sequences and files to another directory
»	define	Define a function
»	dumpxml	Dump an XML representation of sequences and files
»	encoding	encoding("x") Encoding is "x"
»	erase	Erase sequences
»	except	(alias: -) Remove one set of sequences and files from another
»	false	Always false
»	filter	(alias: where) Filter sequences and files by given test
»	foldermatch	folder(r) Sequence is in a folder matching #r#
»	frame	frame(in) Contains frame number "in"
»	fullhandles	(alias: allhandles) Expand sequences to full handles
»	goodtc	Sequence has valid continuous timecode
»	hasaudio	Sequence has audio
»	haskc	Sequence has [some] keycode
»	hastc	Sequence has [some] timecode
»	hastc2	Sequence has [some] timecode 2
»	height	height(n) Sequence is exactly n pixels high
»	inresdir	Sequence resolution matches resolution directory it's in
»	inwrongresdir	Sequence is in resolution directory but its resolution doesn't match
»	isblg	Sequence is a BLG
»	iso	Sequence has a given ISO setting
»	length	length(n) Sequence is exactly n frames long
»	longerthan	longerthan(n) Sequence is longer than n frames
»	mb	Return the approximate uncompressed size of the supplied sequence
»	meta	meta("key",value) Metadata value "key" is value
»	metamatch	metamatch("key",#value#) Metadata value "key" is a string matching #value#
»	modify	(alias: apply) Modify a set of sequences by one or more operations
»	movie	Sequence is a movie
»	namematch	name(#x#) Name matches #x#
»	narrowerthan	narrowerthan(n) Sequence is narrower than n pixels wide
»	newerthan	newerthan(n) Sequence is newer than n days
»	noproxies	Remove proxy sequences
»	not (alias: !)	Invert the following test
»	numframes	Return number of frames in supplied sequence
»	olderthan	olderthan(n) Sequence is older than n days
»	onlyproxies	Keep only proxy sequences
»	or (alias: | ||)	True if one or more of the supplied tests are true
»	plus	Add two sets of sequence and files together
»	renderblg	Copy a set of sequences, rendering them using a BLG
»	resolution	resolution(w,h) Sequence is exactly w x h pixels
»	reverse	Reverse a sort order
»	scene	Return sequences and files used by one Baselight scene
»	scenes	Return sequences and files used by all Baselight scenes in given job or folder
»	sequences	Return only the sequences from the supplied sequences and files
»	shorterthan	shorterthan(n) Sequence is shorter than n frames
»	show	Display a textual summary of what follows
»	sort	(alias: sorted) Sorts a set of sequences into order
»	sum	Apply a numeric function to each sequence and sum the results
»	tape	tape("x") Tape name is "x"
»	tapematch	tape(#x#) Tape name matches #x#
»	tc	tc("tc") Sequence (as far as we know) contains timecode tc
»	tcall	tcall("tc1","tc2") Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]
»	tcfps	Sequence timecode has a given fps
»	true	Always true
»	type	Sequence is a given type
»	under	Return sequences and files under the given directory
»	widerthan	widerthan(n) Sequence is wider than n pixels wide
»	width	width(n) Sequence is exactly n pixels wide

Leave a Trail

FLUX Manage uses the render queue from Baselight and Daylight to marshal its work. The queue itself is held in a database, where lists of tasks are worked through in reassignable priority order, persistent through power outage, machine or network failures.

Jobs that have finished remain in the database with associated reporting and exit status. This provides an audit trail of what has occurred, ready for re-use months later when a similar process has to be repeated, or simply as a tool to help understand exactly what has been done, or how much resource it required.

Customise with XML

You can also access the FLUX indexed filesystem as XML, allowing you to integrate a customised workflow or production database. This is available by either connecting to the flux service directly using your scripting language of choice (e.g. perl), or simply by requesting XML from a fluxc query.

Building the Image Factory

Data wrangling for today's complex productions of 100:1 shooting ratios and 1000+ VFX shots needs to happen on an industrial scale and with industrial efficiency. If not, the pure volume of data to be managed swamps the creative effort. This is not an easy task, and the first release of FLUX in Baselight 5.0 will not provide a complete solution to this problem. But building on a decade of development for Baselight, the product will grow with future releases as the industry evolves.

Already, FLUX uses Baselight's underlying streaming image server and I/O system which allows multiple 2K streams–or even 4K streams–across multiple 10GbE connections, to be laid out unfragmented at their destination.

Future developments will include incorporating Baselight's renderer to be used with custom operations on sets of data, for colour transforms, format conversions, burn-ins, transcoding and so on–a further step in building the image factory pipeline that is becoming a basic requirement to maintain profitability in digital post-production today.

```
FLUX> dumpxml under "/vol/images/jobc" where olderthan 10
<?xml version="1.0" encoding="UTF-8" ?>
<response>
 <rootdirectories>
  <rootdir path="/vol/images/Dior">
   <dir name=".">
    <dir name="raw_qts">
     <movie name="A002R29K_0000000.mov" mtime="1338213430" ftype="QuickTime Movie" fpshdr="25" />
      <frames>
       <fr num="0" tc1="01:03:48:05" TapeName1="A002R29K" X="1920" Y="1080" />
       <fr num="15" tc1="01:03:48:20" TapeName1="A002R29K" X="1920" Y="1080" />
</frames>
...
```