

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

A Changed Business

Digital acquisition has changed the world of postproduction. There are few practical or economic constraints when it comes to acquiring large quantities of highresolution 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 postproduction 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: 8 88) 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
»	сору	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
» »	filter foldermatch	(alias: where) Fliter sequences and files by given test folder(r) Sequence is in a folder matching #r#
» »	filter foldermatch frame	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in"
» » »	filter foldermatch frame fullhandles	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles
» » » »	filter foldermatch frame fullhandles goodtc	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode
» » » »	filter foldermatch frame fullhandles goodtc hasaudio	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode Sequence has audio
»» » » » »	filter foldermatch frame fullhandles goodtc hasaudio haskc	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode Sequence has audio Sequence has [some] keycode
» » » » » » » » » » » » » » » » » » »	filter foldermatch frame fullhandles goodtc hasaudio haskc hastc	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode Sequence has audio Sequence has [some] keycode Sequence has (some] timecode
»» »» »» »>>>>>>>>>>>>>>>>>>>>>>>>>>>>	filter foldermatch frame fullhandles goodtc hasaudio haskc hastc hastc2	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode Sequence has audio Sequence has [some] keycode Sequence has [some] timecode 2
>> >> >> >> >> >> >> >> >> >> >> >> >>	filter foldermatch frame fullhandles goodtc hasaudio hasaudio hastc hastc2 height	(alias: where) Filter sequences and files by given test folder(r) Sequence is in a folder matching #r# frame(in) Contains frame number "in" (alias: allhandles) Expand sequences to full handles Sequence has valid continuous timecode Sequence has audio Sequence has [some] keycode Sequence has [some] timecode 2 height(n) Sequence is exactly n pixels high
» » » » » » » » » » » » » » » » » » »	filter foldermatch frame fullhandles goodtc hasaudio haskc hastc2 hastc2 height inresdir	(alias: where)Filter sequences and files by given testfolder(r)Sequence is in a folder matching #r#frame(in)Contains frame number "in"(alias: allhandles)Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has [some] keycodeSequence has [some] timecode 2Sequence has [some] timecode 2height(n)Sequence is exactly n pixels highSequence resolution matches resolution directoryit's in
» » » » » » » » » » » » » » » » » » »	filter foldermatch frame fullhandles goodtc hasaudio hasaudio hastc2 hastc2 hastc2 hastc2 inresdir	(alias: where)Filter sequences and files by given testfolder(r)Sequence is in a folder matching #r#frame(in)Contains frame number "n"(alias: allhandles)Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has [some] keycodeSequence has [some] timecode 2Sequence has [some] timecode 2height(n)Sequence resolution matches resolution directory it's inSequence is in resolution directory but its resolution doesn't match
>> >> >> >> >> >> >> >> >> >> >> >> >> >> >> >> >> >>	filter foldermatch frame fullhandles goodtc hasaudio haskc hastc2 hastc2 height inresdir inwrongresdir isblg	(alias: where) Filter sequences and files by given testfolder(r) Sequence is in a folder matching #r#frame(in) Contains frame number "in"(alias: allhandles) Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has [some] keycodeSequence has [some] timecode 2height(n) Sequence is exactly n pixels highSequence resolution matches resolution directory it's inSequence is n resolution directory but its resolution doesn't matchSequence is a BLG
>> >>	filter foldermatch frame fullhandles goodtc hasaudio haskc hastc2 hastc2 height inresdir invrongresdir isblg iso	(alias: where)Filter sequences and files by given testfolder(r)Sequence is in a folder matching #r#frame(in)Contains frame number "in"(alias: allhandles)Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has [some] keycodeSequence has [some] timecode 2Sequence has [some] timecode 2height(n)Sequence resolution matches resolution directory its inSequence is in resolution directory but its resolution doesn't matchSequence is a BLGSequence has a given ISO setting
>> >>	filter foldermatch frame fullhandles goodtc basaudio haskc hastc2 hastc2 height inresdir invrongresdir isblg iso length	(alias: where)Filter sequences and files by given testfolder(r)Sequence is in a folder matching #r#(rame(in)Contains frame number "in"(alias: allhandles)Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has [some] keycodeSequence has [some] timecode 2Sequence has [some] timecode 2height(n)Sequence resolution matches resolution directory it's inSequence is a BLGSequence has a given ISO settinglength(n)Sequence is exactly n frames long
	filter foldermatch frame fullhandles goodtc baskc haskc haskc hastc2 hastc2 inersdir inresdir invrongresdir isblg iso length longerthan	(alias: where) Filter sequences and files by given testfolder(r) Sequence is in a folder matching #r#frame(in) Contains frame number "in"(alias: alihandles) Expand sequences to full handlesSequence has valid continuous timecodeSequence has audioSequence has (some) keycodeSequence has (some) timecode 2Sequence has (some) timecode 2height(n) Sequence is exactly n pixels highSequence is in resolution directory but its resolution doesn't matchSequence is a BLGSequence is a given ISO settinglength(n) Sequence is exactly n frames longlongerthan(n) Sequence is longer than n frames

meta meta(he/y/suble) Metadata value 'key' is a string matching #value# Metadata value 'key' is a string matching #value# Metadata value 'key' is a string matching #value# Metadata value 'key' is a string matching #value# Modify a set of sequences by one or more operations movie movie Sequence is a movie movie Sequence is a movie namematch namewrthan namewrthan(n) Sequence is newer than n pixels wide norrowerthan newerthan(n) Sequence is newer than n days norrowerthan newerthan(n) Sequence is newer than n days norrowerthan newerthan(n) Sequence is older than n days norrowerthan Neturn number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days or (alias I) True if one or more of the supplied tests are true pius Add two sets of sequences and files together reverse Reverse a sort order reverse Reverse a sort order scene Return sequences and files used by one Baselight scene in given job or folder scene Sorts a set of sequences is horter than n frames sort (alies sorted) Sorts a set of sequences is horter than n frames show Display a textual summary of what follows sort (alies sorted) Sorts a set of sequences into order sort tope(fxi) Tape norme matches #x# sort tope(fxi) Tape norme matches			
metamatch metamatch/key/#value#) Metadata value 'key' is a string matching #value# modify (allas apply) Metadata value 'key' is a string matching #value# movie Sequence is a movie namematch nameworthan narrowerthan narrowerthan(n) Sequence is narrower than n pixels wide newerthan newerthan(n) Sequence is narrower than n days not (allas i) inver the following test not (allas i) inver the following test numframes Return number of frames in supplied sequence olderthan Sequence is older than n days or (allas i) True if one or more of the supplied tests are true or (allas i) True if one or more of the supplied tests are true plus Add two sets of sequences, rendering them using a BLG renderbig Copy a set of sequences and files together reverse Reverse a sort order scene Return sequences and files used by one Baselight scene scene Return sequences and files used by all Baselight scene show Display a textual summary of what follows show Display a textual summary of what follows show Display a textual summary of what follows squences in given jb or folder Sequence is scene show Display a textual summary of what follows <td< td=""><td>»</td><td>meta</td><td>meta("key",value) Metadata value "key" is value</td></td<>	»	meta	meta("key",value) Metadata value "key" is value
modify (allas: apply) Modify a set of sequences by one or more operations movie Sequence is a movie namematch namewerthan narrowerthan narrowerthan(n) Sequence is narrower than n pixels wide newerthan newerthan(n) Sequence is newer than n days noproxies Remove proxy sequences not (allas: 1) invert the following test numframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days or (allas: 1) True if one or more of the supplied tests are true or (allas: 1) True if one or more of the supplied tests are true plus Add two sets of sequences, rendering them using a BLG resolution resolution(wh) Sequence is exactly w x h pixels resolution resolution(wh) Sequences and files used by one Baselight scene scene Return sequences and files used by all Baselight scenes in given job or folder soft soft shorterthan storterthan shorterthan(n) Sequence is shorter than n frames given job or folder softs a set of sequences into order soft soft shorterthan(n) Sequence is norder soft soft soft a set of sequences into order soft tape ^(X) Tape name is X [*] tape tape	>	metamatch	metamatch("key",#value#) Metadata value "key" is a string matching #value#
innovie Sequence is a movie inamematch namematches #x# inarrowerthan narrowerthan(n) Sequence is narrower than n pixels wide newerthan(n) Sequence is narrower than n days newerthan(n) inarrowerthan newerthan(n) Sequence is newer than n days newerthan(n) inarrowerthan newerthan(n) inarrowerthan newerthan(n) inarrowerthan(n) Sequence is older than n days inarrowerthan olderthan(n) inarrowerthan(n) Sequence is older than n days inarrowerthan olderthan(n) inarrowerthan(n) Sequence is older than n days inarrowerthan(n) Sequence is older than n days inarrowerthan(n) Sequence is exactly with a sequences in gloen provide tests are true inarrowerthan(n) Sequence is exactly with pixels inarrowerthan Return sequences and files used by one Baselight scene inarrowerthan shorterthan(n) Sequence is shorter than n frames inarrowerthan Secrees in gloen job of folder inarrowerthan(n) Sequences and files used by all Baselight scenes in gloen job of folder inarrowerthan shorterthan(n) Sequence is shorter than n frames inarrowerthan(n) Sequence is equences into order inarowerthan(n)	•	modify	(alias: apply) Modify a set of sequences by one or more operations
namematch namematches #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 (allas i) Invert the following test numframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days or (allas i) True if one or more of the supplied tests are true or (allas i) True if one or more of the supplied tests are true plus Add two sets of sequences in diffes together resolution Sequence is exactly with pixels resolution Return sequences and files used by one Baselight scene scene Return sequences from the supplied sestelight scene scenes Return only the sequences from the supplied sequences soft softerthan shorterthan shorterthan(n) Sequence is shorter than n frames show Display a textual summary of what follows soft (alias sorted) sort (alias sorted) sorts a set of sequences into order sort (alias sorted) sort (alias sorted) </td <td>,</td> <td>movie</td> <td>Sequence is a movie</td>	,	movie	Sequence is a movie
narrowerthan narrowerthan(n) Sequence is narrower than n pixels wide newerthan newerthan(n) Sequence is newer than n days not (alias: i) invert the following test numframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days or (alias: i) True if one or more of the supplied tests are true or (alias: i) True if one or more of the supplied tests are true or (alias: i) True if one or more of the supplied tests are true renderbig 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 scenes in given job or folder scenes Return only the sequences into order shorterthan shorterthan(n) Sequence is shorter than n frames squences sect of sequences into order sort (alias sorted) sort (alias sorted) sort tape name is x' tape name is x' Sequence (as far as we know) contains timecode tc <	•	namematch	name(#x#) Name matches #x#
newerthan newerthan(n) sequence is newer than n days noproxies Remove proxy sequences not (alias: i) Invert the following test numframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days or (alias: i) or (alias: i) True if one or more of the supplied tests are true plus Add two sets of sequences, rendering them using a BLG renderbig Copy a set of sequences, rendering them using a BLG reverse Reverse a sort order reverse Reverse a sort order scene Return sequences and files used by one Baselight scenes in given job or folder scenes Return only the sequences from the supplied sequence and files shorterthan shorterthan() Sorts a set of sequences into order sort (alias: sorted) sort (alias: sorted) sorts a set of sequences into order squence (as far as we know) contains timecode tc tape tcpf(Xr) tape name is X' tape tcall tape tcall(tcf.tc2) sequence (as far as we know) contains all of the t	•	narrowerthan	narrowerthan(n) Sequence is narrower than n pixels wide
noproxies Remove proxy sequences invert the following test invert test <td>•</td> <td>newerthan</td> <td>newerthan(n) Sequence is newer than n days</td>	•	newerthan	newerthan(n) Sequence is newer than n days
interference invert the following test inumframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days inumframes Keep only proxy sequences inumframes Keep only proxy sequences inumframes Add two sets of sequence and files together inumframes Add two sets of sequences, rendering them using a BLG inumframes Reverse a sort order inumframes Return sequences and files used by one Baselight scene inumframes Return sequences and files used by all Baselight scenes inumframes Return only the sequences from the supplied sequences in given job of folder inumframes Sorter a sort order inumframes Sorter a set of sequences into order inumframes Sorter a set of sequences in the supplied sequence and sum the results inumframes Sorts a set of sequences into order inumframes Sorts a set of sequences into order intape tape(%) Tape name is % intape mark the results Tape name matches ### intape tcali("tcf","tc2") Sequence is far as we know) contains tinecode to Sequence is a given		noproxies	Remove proxy sequences
numframes Return number of frames in supplied sequence olderthan olderthan(n) Sequence is older than n days onlyproxies Keep only proxy sequences or (dias; I) True if one or more of the supplied tests are true plus Add two sets of sequence and files together renderbig Copy a set of sequences, rendering them using a BLG resolution resolution(w,h) Sequence is exactly w × h pixels reverse Reverse a sort order scene Return sequences and files used by one Baselight scenes in given job or folder squences Return only the sequences from the supplied sequences and files shorterthan shorterthan (h) Sequence is shorter than n frames show Display a textual summary of what follows sort (alias: sorted) Sorts a set of sequences into order swm Apply a numeric function to each sequence and sum the results tape tape(#X#) Tope name matches #X# tape tcal(tcf) Sequence (as far as we know) contains timecode tc tcall tcall(tcf)'tc2) Sequence (as far as we know) contains all of the timecodes in the range[tctt2] true Always true true Always true true Always true		not (alias: !)	Invert the following test
olderthanolderthan(n) Sequence is older than n daysonlyproxiesKeep only proxy sequencesor (allas:)True if one or more of the supplied tests are trueplusAdd two sets of sequence and files togetherrenderbigCopy a set of sequences, rendering them using a BLGresolutionresolution(w,h)Sequence is exactly w x h pixelsreverseReverse a sort ordersceneReturn sequences and files used by one Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapematchtape(##) Tape name is x'tapematchtape(##) Sequence (as far as we know) contains timecode to tocal(tcr!tc2) Sequence into and pisetupeSequence timecode has a given fpstupeSequence timecode has a given fpsunderReturn sequences and files under the given directory widerthan(n) Sequence is a given typeunderReturn sequences and files under the given directorywiderthanSequence is exactiy n pixels wide		numframes	Return number of frames in supplied sequence
or (alias:]) True if one or more of the supplied tests are true or (alias:]) True if one or more of the supplied tests are true plus Add two sets 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 scenes 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 (as far as we know) contains timecode tc tcall tcall(tct", tc2") Sequence (as far as we know) contains all of the timecodes in the range[tcttc2] ttppe Sequence (as far as we know) contains all of the timecodes in the range[tcttc2] ture Always true ture Always true under Return sequences and files under the given directory <td>•</td> <td>olderthan</td> <td>olderthan(n) Sequence is older than n days</td>	•	olderthan	olderthan(n) Sequence is older than n days
or (alias: II) True if one or more of the supplied tests are true plus Add two sets of sequence and files together renderbig 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 scene sequences Return only the sequences from the supplied sequences and files shorterthan shorterthan(h) Sequence is shorter than n frames show Display a textual summary of what follows sort (alias: sorted) sort (alias: sorted) sort (alias: sorted) sort (alias: sorted) sort (alias: for a set of sequences into order tape tape(x) Tape name is 'x' tape tape(x) Tape name matches #x# tc tc(tfc') Sequence (as far as we know) contains timecode to type Sequence is a given type true tupe(Tx) Sequence (as far as we know) contains all of the timecodes in the range[tc(tc2]) <t< td=""><td>•</td><td>onlyproxies</td><td>Keep only proxy sequences</td></t<>	•	onlyproxies	Keep only proxy sequences
plus Add two sets of sequence and files together renderbig 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 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 (allas: 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 tcill tcall(tct?tc2) Sequence (as far as we know) contains all of the timecodes in the range[tcttc2] ttppe Sequence is a given type under Return sequences and files under the given directory widerthan widerthan(n) Sequence is wider than n pixels wide		or (alias:)	True if one or more of the supplied tests are true
renderbigCopy a set of sequences, rendering them using a BLGresolutionresolution(w,h) Sequence is exactly w x h pixelsreverseReverse a sort ordersceneReturn sequences and files used by one Baselight scenescenesReturn sequences and files used by all Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(*x) Tape name is 'x'tapematchtape(*tr) Sequence (as far as we know) contains timecode tctapittcall("tcf","tc2") Sequence (as far as we know) contains all of the timecodes in the range[tc1tc2]trueAlways truetuppeSequence is a given typeunderReturn sequences and files under the given directory widerthan(n) Sequence is wider than n pixels wide		plus	Add two sets of sequence and files together
resolutionresolution(w,h) Sequence is exactly w x h pixelsreverseReverse a sort ordersceneReturn sequences and files used by one Baselight scenescenesReturn sequences and files used by all Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(x) Tape name is x'tapematchtape(tc') Sequence (as far as we know) contains timecode tctcltc(tc') Sequence (as far as we know) contains all of the timecodes in the range[tcttc2]trueAlways truetupeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwiderthan(n) Sequence is wider than n pixels wide		renderblg	Copy a set of sequences, rendering them using a BLG
reverseReverse a sort ordersceneReturn sequences and files used by one Baselight scenescenesReturn sequences and files used by all Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(x) Tape name is 'x'tapematchtape(#x#) Tope name matches #x#tctc('tc') Sequence (as far as we know) contains timecode tctcalltcall('tc1,''tc2') Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]trueAlways truetupeSequence is a given typeunderReturn sequences and files under the given directory widerthan (n) Sequence is wider than n pixels wide		resolution	resolution(w,h) Sequence is exactly w x h pixels
sceneReturn sequences and files used by one Baselight scenescenesReturn sequences and files used by all Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(*) Tape name is *tapematchtape(#x#) Tape name matches #x#tctc(tc') Sequence (as far as we know) contains timecode tctclltcalltclltcall("tcf", "tc2") Sequence (as far as we know) contains all of the timecodes in the range[tct(tc2]trueAlways truetupeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwiderthan(n) Sequence is wider than n pixels wide		reverse	Reverse a sort order
scenesReturn sequences and files used by all Baselight scenes in given job or foldersequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(X) Tape name is X'tapematchtape(#X#) Tape name matches #X#tctc('tc') Sequence (as far as we know) contains timecode tctcalltcall('tc1,''tc2') Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]trueAlways truetypeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwidth(n) Sequence is exactly n pixels wide		scene	Return sequences and files used by one Baselight scene
sequencesReturn only the sequences from the supplied sequences and filesshorterthanshorterthan(n) Sequence is shorter than n framesshowDisplay a textual summary of what followssort(alia: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape(X) Tape name is X'tapematchtape(#X#) Tape name matches #X#tctc(*tc') Sequence (as far as we know) contains timecode tctcalltcall(*tcf,*tc2) Sequence (as far as we know) contains all of the timecodes in the range[tc1tc2]trueAlways truetypeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwidth(n) Sequence is exactly n pixels wide		scenes	Return sequences and files used by all Baselight scenes in given job or folder
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[tc1tc2] 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	•	sequences	Return only the sequences from the supplied sequences and files
ishowDisplay a textual summary of what followssort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape('x') Tape name is 'x'tapematchtape(#x#) Tape name matches #x#tctc('tc') Sequence (as far as we know) contains timecode tctcalltcall('tc1','tc2') Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]trueAlways truetypeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwiderthan(n) Sequence is wider than n pixels wide		shorterthan	shorterthan(n) Sequence is shorter than n frames
sort(alias: sorted) Sorts a set of sequences into ordersumApply a numeric function to each sequence and sum the resultstapetape('X') Tape name is 'X'tapematchtape(#x#) Tape name matches #x#tctc('tc') Sequence (as far as we know) contains timecode tctcalltcall('tcf,'tc2') Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]trueAlways truetrueAlways trueunderReturn sequences and files under the given directorywiderthanwiderthan(n) Sequence is wider than n pixels widewidth(n) Sequence is exactly n pixels wide		show	Display a textual summary of what follows
sumApply a numeric function to each sequence and sum the resultstapetape(*x) Tape name is "x"tapematchtape(#x#) Tape name matches #x#tctc("tc") Sequence (as far as we know) contains timecode tctcalltc("tc","tc2") Sequence (as far as we know) contains all of the timecodes in the range[tc1(tc2])tcfpsSequence timecode has a given fpstrueAlways truetypeSequence is a given typeunderReturn sequences and files under the given directorywiderthanwiderthan(n) Sequence is wider than n pixels widewidth(n) Sequence is exactly n pixels wide		sort	(alias: sorted) Sorts a set of sequences into order
tapetape(X) Tape name is X'tapematchtape(#X#) Tape name matches #X#tctco("tc") Sequence (as far as we know) contains timecode tctcalltcall("tc","tc2") 		sum	Apply a numeric function to each sequence and sum the results
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		tape	tape('x') Tape name is 'x'
tc tc('tc') Sequence (as far as we know) contains timecode tc tcall tcall('tc','tc2') Sequence (as far as we know) contains all of the timecodes in the range[tc',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	•	tapematch	tape(#x#) Tape name matches #x#
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		tc	tc("tc") Sequence (as far as we know) contains timecode tc
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		tcall	tcall("tc1","tc2") Sequence (as far as we know) contains all of the timecodes in the range[tc1,tc2]
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		tcfps	Sequence timecode has a given fps
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		true	Always true
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		type	Sequence is a given type
widerthan widerthan(n) Sequence is wider than n pixels wide width width(n) Sequence is exactly n pixels wide		under	Return sequences and files under the given directory
width width(n) Sequence is exactly n pixels wide	>	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.

FLUX> dumpxml under "/vol/images/jobc" where olderthan 10
<pre></pre> // version= 1.0 encoding= UIF-8 // // // // // // // // // // // // //
<response></response>
<rootdirectories></rootdirectories>
<rootdir path="/vol/images/Dior"></rootdir>
<dir name="."></dir>
<pre><dir name="raw_qts"></dir></pre>
<movie <="" fpshdr="25" ftype="QuickTime Movie" mtime="1338213430" name="A002R29K_0000000.mov" td=""></movie>
<frames></frames>
<fr num="0" tapename1="A002R29K" tc1="01:03:48:05" x="1920" y="1080"></fr>
<fr num="15" tapename1="A002R29K" tc1="01:03:48:20" x="1920" y="1080"></fr>

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.

Head Office & EMEA	Australia	Germany	Japan	Singapore	www.filmlight.ltd.uk
London, UK	Sydney	Berlin	Tokyo	Singapore	
t: +44.20.7292.0400	t: +61.2.8746.0602	t: +49.151.2345.5668	t: +81.3.6801.6280	t: +65.9670.3283	
info@filmlight.ltd.uk	China	India	Mexico	Thailand	
	Beijing	Mumbai	Mexico City	Bangkok	
	t: +86.139.1073.7940	t: +91.9819.426.677	t: +52(1)55.5165.2132	t: +66.891.259.009	
				USA	

Los Angeles t: +1.323.785.1<u>630</u>

FilmLight

Northlight, Baselight, Truelight, Daylight, Prelight, FLIP, FLUX, Blackboard and Slate are trademarks of FilmLight Ltd.

Other products that are referred to in this document may be either trademarks and/or registered trademarks of the respective owners. The publisher and the author make no claim to these trademarks.

© FilmLight 2017