7. Automating workflows using Render Automation

Render Automation is an add-on included in PixelConduit Complete. It can be used to control and automate many things within a project.

The core of Render Automation is the concept of “batch actions”. These are events that are performed automatically in sequence.

In this section of the User’s Guide, I’d like to show you how Render Automation’s batch actions work. I hope to also give you an idea of why batch actions are more powerful than an ordinary render queue because actions can interact with the project in many interesting ways.

Some of the things you can do with batch actions are:

- Converting video files to various movie and image sequence formats
- Rendering variations of a project by modifying individual node widgets
  (e.g. loading different background clips while leaving other parts of the project unchanged)
- Applying different effects to different video clips in the batch list
- Creating instant variations of an effect by saving effects directly into the batch list
- Easily rendering dual stereo 3D streams to single-stream previews, or vice versa
- Performing JavaScript commands within the batch list for completely customizable actions.

The elements of an action

A batch action in Conduit is essentially like a little robot that can do things within the application based on your instructions. (It’s fairly similar to an action in Photoshop, if you’re familiar with those.) Each action is built up from commands, which in Conduit are called *events*. The events within an action can do many things, but the most useful are:

Image/movie event – load a video or picture file into a source within the project

Conduit event – load an effect setup into the project

Slider event – modify a slider value (this is useful for easily rendering variations, e.g. with different opacities for some effect)

Color picker event – modify a color picker value (also useful for variations)

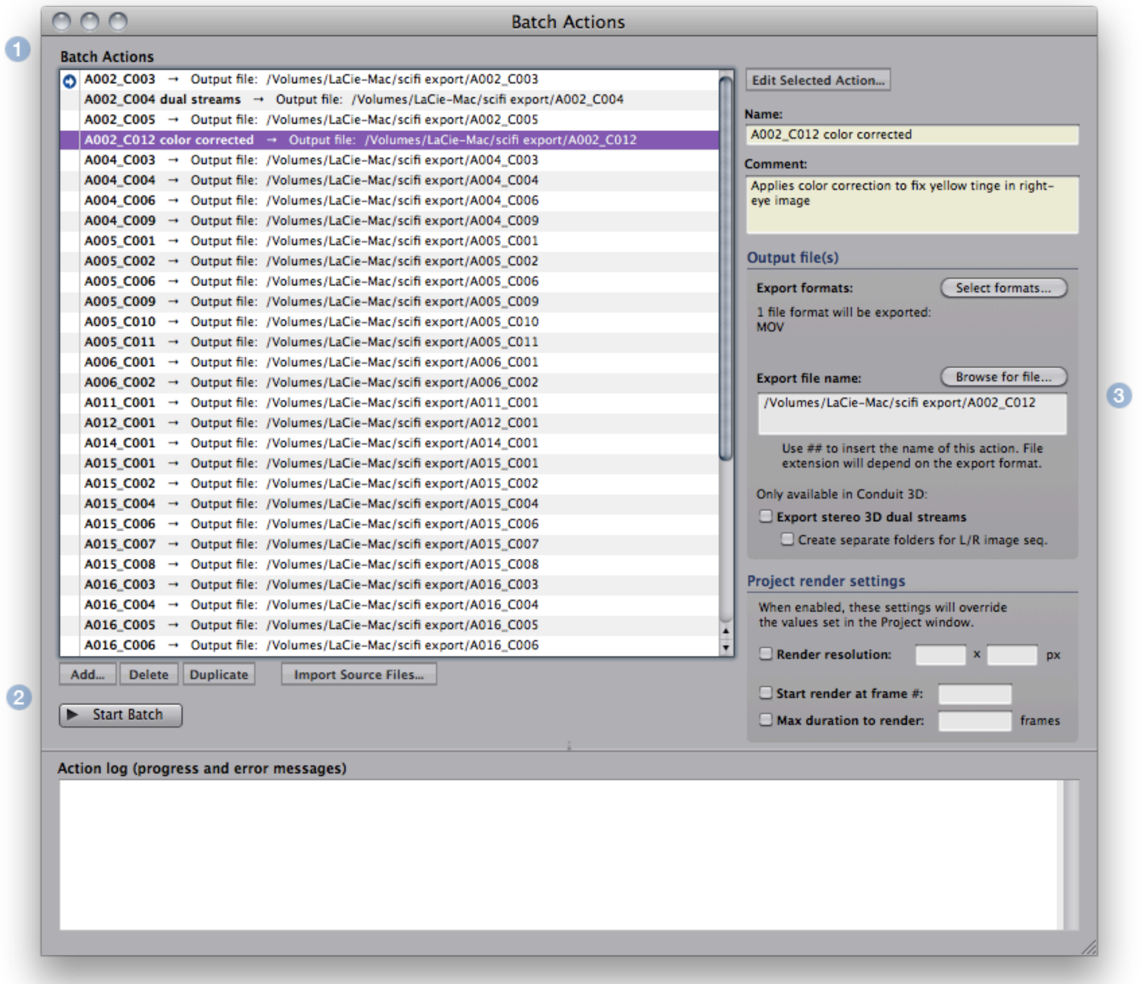
JavaScript event – perform custom commands on any scriptable node widget

A batch action also has *output settings*. These determine whether the action renders some output to disk. Settings include the destination file, render resolution, export formats (e.g. QuickTime and image sequence), whether to render dual-stream 3D, and so on.

Note that you can create an action that doesn't render output. This can be very useful if you want to have a change happen during the batch – for example, loading a different effect setup. Similarly, you can create an action that doesn't have any events. Such an empty action simply renders out the project in its current state.

The Batch Actions window

Everything about batch actions happens in this window. You can find it in the Tools menu.



Batch Actions

```
A002_C003 → Output file: /Volumes/LaCie-Mac/scifi export/A002_C003
A002_C004 dual streams → Output file: /Volumes/LaCie-Mac/scifi export/A002_C004
A002_C005 → Output file: /Volumes/LaCie-Mac/scifi export/A002_C005
A002_C012 color corrected → Output file: /Volumes/LaCie-Mac/scifi export/A002_C012
A004_C003 → Output file: /Volumes/LaCie-Mac/scifi export/A004_C003
A004_C004 → Output file: /Volumes/LaCie-Mac/scifi export/A004_C004
A004_C006 → Output file: /Volumes/LaCie-Mac/scifi export/A004_C006
A004_C009 → Output file: /Volumes/LaCie-Mac/scifi export/A004_C009
A005_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C001
A005_C002 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C002
A005_C006 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C006
A005_C009 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C009
A005_C010 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C010
A005_C011 → Output file: /Volumes/LaCie-Mac/scifi export/A005_C011
A006_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A006_C001
A006_C002 → Output file: /Volumes/LaCie-Mac/scifi export/A006_C002
A011_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A011_C001
A012_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A012_C001
A014_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A014_C001
A015_C001 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C001
A015_C002 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C002
A015_C004 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C004
A015_C006 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C006
A015_C007 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C007
A015_C008 → Output file: /Volumes/LaCie-Mac/scifi export/A015_C008
A016_C003 → Output file: /Volumes/LaCie-Mac/scifi export/A016_C003
A016_C004 → Output file: /Volumes/LaCie-Mac/scifi export/A016_C004
A016_C005 → Output file: /Volumes/LaCie-Mac/scifi export/A016_C005
A016_C006 → Output file: /Volumes/LaCie-Mac/scifi export/A016_C006
```

Name: A002_C012 color corrected

Comment: Applies color correction to fix yellow tinge in right-eye image

Output file(s)

Export formats:

1 file format will be exported:
MOV

Export file name: /Volumes/LaCie-Mac/scifi export/A002_C012

Use ## to insert the name of this action. File extension will depend on the export format.

Only available in Conduit 3D:

☐ Export stereo 3D dual streams

☐ Create separate folders for L/R image seq.

Project render settings

When enabled, these settings will override the values set in the Project window.

☐ Render resolution: ____ x ____ px

☐ Start render at frame #: ____

☐ Max duration to render: ____ frames

Action log (progress and error messages)

The elements of this window are:

1) Batch List – this is where all the actions are.
Double-click an item in the list to set the batch cursor, that is, the first action to be rendered. The cursor is indicated with a blue arrow to the left of the item name. (By default, the cursor is at the start of the list, so all actions get executed.)

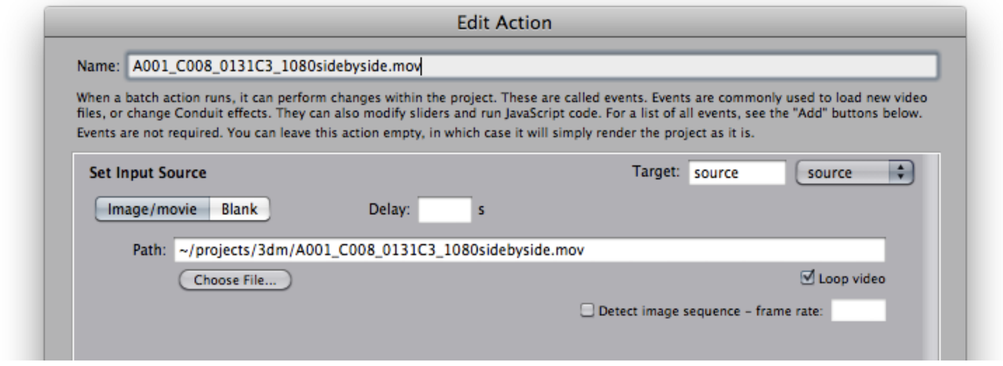
2) Commands.
Add, delete or duplicate actions in the list; start the batch render; import files.

3) Settings for the selected action.
Settings include a name and description for the action; its output files; and project render settings which can be used to override render resolution and in/out times.

Importing source files

This is probably the most common use case for batch automation in Conduit: you have a bunch of videos and want to render them out to a different format (or multiple formats), perhaps with some effects applied. To accomplish this, we need to create one action for each source file.

You could do this manually by clicking on the **Add** button, then adding an image source event to the new action and specifying a file as the input:


Edit Action

Name: A001_C008_0131C3_1080sidebyside.mov

When a batch action runs, it can perform changes within the project. These are called events. Events are commonly used to load new video files, or change Conduit effects. They can also modify sliders and run JavaScript code. For a list of all events, see the "Add" buttons below. Events are not required. You can leave this action empty, in which case it will simply render the project as it is.

Set Input Source — Target: source, source

Image/movie | Blank — Delay: s

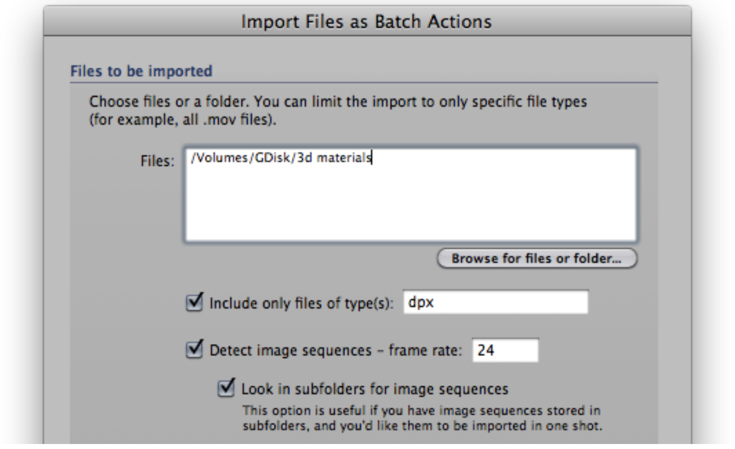
Path: ~/projects/3dm/A001_C008_0131C3_1080sidebyside.mov

Choose File...

☑ Loop video

☐ Detect image sequence – frame rate:

There is an easier way, however. Click on the **Import Multiple Files** button, and the following dialog opens:


Import Files as Batch Actions

Files to be imported

Choose files or a folder. You can limit the import to only specific file types (for example, all .mov files).

Files: /Volumes/GDisk/3d materials

Browse for files or folder...

☑ Include only files of type(s): dpx

☑ Detect image sequences – frame rate: 24

☑ Look in subfolders for image sequences
This option is useful if you have image sequences stored in subfolders, and you'd like them to be imported in one shot.

Click Browse to select the files or folders that you want to import, or write paths in the field. (Multiple files or folders can be written separated by a semicolon, or you can pick them with the Browse file dialog.)

If you want to limit the import to specific files within a folder, for example QuickTime movies only, enter the file types in the corresponding entry field. You can include multiple types separated by comma, for example:

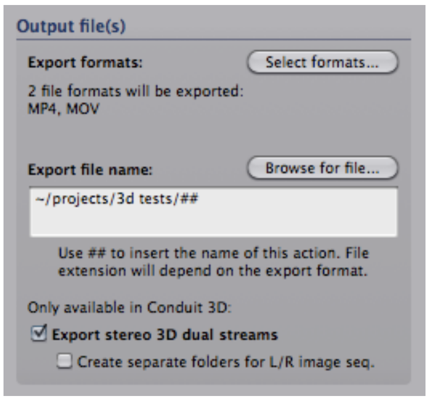
```
mov, mp4, avi
```

If you're using image sequences, there is an additional option for looking for files in subfolders. This is useful to deal with the common case where each image sequence is stored in a folder of its own – using this option, those folders will be imported like they were individual files.

Setting up exports

After using the multi-import, we have a bunch of actions that load video files. They don't yet have any outputs set, though. If you execute the batch now, it will load all the clips but won't actually write out anything. (What's the point here? It's that actions without outputs can be pretty useful. You could for example set up a background image in one action, then load a batch of foreground images and render out the composites. In this case you wouldn't want the first action that loads the background to render anything.)

To set the output files, select the action or multiple actions that you want to render out. Then edit the Output settings:


Output file(s)

Export formats:
2 file formats will be exported:
MP4, MOV

Export file name:
~/projects/3d tests/##

Use ## to insert the name of this action. File extension will depend on the export format.

Only available in Conduit 3D:
☑ Export stereo 3D dual streams
☐ Create separate folders for L/R image seq.

Click **Select formats** to choose the outputs. (Note: due to a limitation of how QuickTime exporting works in PixelConduit, you can't select multiple codecs for QuickTime output. All actions that use QuickTime output within the same batch will render using the same codec, which you'll be able to select when the batch is executed. This limitation doesn't apply to any other formats; they can all be independently set for different actions.)

A single action can export to multiple formats in one shot. This is convenient when you want to have both a full-resolution image sequence and a preview-quality movie, for example.

In the **Export file name** field, enter a path for the exported file, or choose it using the operating system's file dialog by clicking Browse.

It's often the case that the exported file should have the same name as the action. To accomplish this, simply enter the characters ## as the file name, as shown in the above screenshot. (You can use ## as part of the file name to make the file name more specific.)

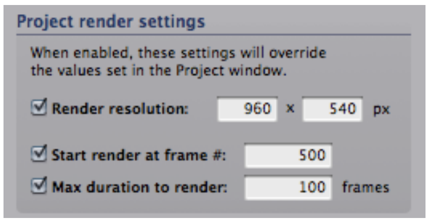
Another useful character that you can use in file names is ~ (tilde). It means the current user's home folder – if your user name on the computer is "John", the tilde would be expanded to */Users/john*.

Telling the project how and what to render

A project in PixelConduit has a number of settings that affect how the project gets rendered. These include the project resolution (which determines the size of images rendered by effects) and the In/Out times that you can set for the project when in timeline mode. (Specifically, In/Out times determine the specific frames within the project's timeline that actually get rendered. They are like a "loop region" in some other applications.)

Normally, you specify these in the Project window. For batch actions, these often need to be changed during the batch. For example, if an action is meant to only provide a quick preview, you might want it to render at a lesser resolution and only the first 10 seconds of the timeline.

To accomplish this kind of control, use the Project render settings:



Project render settings

When enabled, these settings will override the values set in the Project window.

☑ Render resolution: 960 x 540 px

☑ Start render at frame #: 500

☑ Max duration to render: 100 frames

These settings are hopefully self-explanatory. For easiest control, the render in/out times are given in frames. At what frame should the render start, and how many frames should be rendered?

(To render out still images, you can of course specify a duration of 1 frame.)

When moving around in the PixelConduit project's timeline, you can easily find out what the current frame is: it's displayed in a yellow box in the top-left-hand corner of the Project window, just below the start of the timeline.

Executing the batch

Once your actions and their output files are set up, you can run the actions by clicking the **Start Batch** button.

As the batch is being executed, information about the batch status is printed in the Action Log text box at the bottom of the window.

Note that executing the batch always starts at the **batch cursor**. It's a little blue arrow displayed at the right-hand side of the actions list. To set where the cursor is, double-click an action in the list. (I.e. to execute just the last action, double-click it, then click Start Batch.)

Modifying effects within actions

Batch actions are not limited to loading video files and rendering out to different formats. They can also modify Conduit effects as well as set new values for sliders and color pickers. With these tools, you can easily create actions that change the project's output completely.

To make an action that changes a Conduit effect, click **Edit Selected Action**. This opens the Edit Action window.

As was previously mentioned, actions are built from events. We want to add a *conduit event* – an event that sets the contents of a Conduit Effect node widget within the project. Click **Add Conduit Event**.

Each event needs a target. If you only have one Conduit Effect node widget in your project, you don't need to worry about this – the target will be automatically set. But if you're using multiple effects, then you should pick the one you want to affect from the Target pop-up menu.

Next, we need to specify the effect that the event will apply. There are three ways to load an effect: either from the target node widget (i.e. whatever is the current effect); from a .conduit file saved on disk; or from the Conduit Editor window. These load options are available as buttons directly in the event list.

Once an effect is loaded into the event, it's saved permanently as part of the action. Modifications that you do in the Conduit Editor won't affect the action. To replace the event, open the Edit Action window again and click one of the load buttons.

If you want to get even more control, you can write JavaScript code to be executed within batch actions. Conduit has powerful scripting functionality, and this interface gives you complete control over scripts within the project.

For example, you could use a simple Canvas rendering script to render translucent timecodes that are composited on top of the image. With batch actions you could then change the values within that script, e.g. to modify the layout, or have specific texts printed out in different actions.

There are three scriptable node widgets: *Scripted Effect* (can be used to render hardware-accelerated graphics), *Scripted 2D Canvas* (easy way to render 2D graphics using the HTML5 Canvas interface), and *Script Widget* (can be used to write scripts that compute and output values instead of graphics). There are some example script projects provided as part of the templates available in the PixelConduit startup screen.

Working with stereo 3D content

PixelConduit Complete includes 3D Tools. This add-on also improves batch actions with capabilities that specifically address the challenges of working with stereo images.

Stereo 3D content is often stored in image sequences, one for each eye, in separate folders. PixelConduit + 3D Tools has built-in support for exporting in this format. In the output settings for an action, enable **Export stereo 3D dual streams** and **Create separate folders for L/R image sequences**.

Converting 3D images from two separate streams to single-stream stereo is a common operation. Joined images are usually in a side-by-side or top/bottom layout. The idea is that both eyes' images can be fit in one video frame, at the expense of losing half the resolution. Most 3D TVs can view images in this format, so it's quite convenient for editing and viewing.

These conversions can be really painful to do with most tools. In Conduit 3D, it's easy to convert a batch of dual-stream video to side-by-side format. Here are the specific steps:

In your project, create two *Image/Movie Source* node widgets, one for each stream. (You should name them something like *Left* and *Right* to make it easier to target the actions to them.) Then, create a *Stereo 3D Preview* node widget, and select side-by-side as the output type. In the Project view, connect the two inputs to the Preview node and its output to the display node. Now you're all set.

When importing your files, you'll need to do two passes using Import Source Files in the Batch Actions window: first import the files for the left eye and target them to the “Left” source node widget in the project, then import the right eye files. (Alternatively you can import the right eye files by editing the created actions to load both of the inputs within the same action.)

With the actions done, specify outputs for those actions that should render out. If you have separate actions to load left and right, you’ll want to interleave them so that “load left” is executed first, then “load right” renders out. (Just leave the export path empty for the “load left” actions, and they won’t render.)