ASC Framing Decision List (ASC FDL)

Advanced Data Management Subcommittee

THE AMERICAN SOCIETY OF CINEMATOGRAPHERS

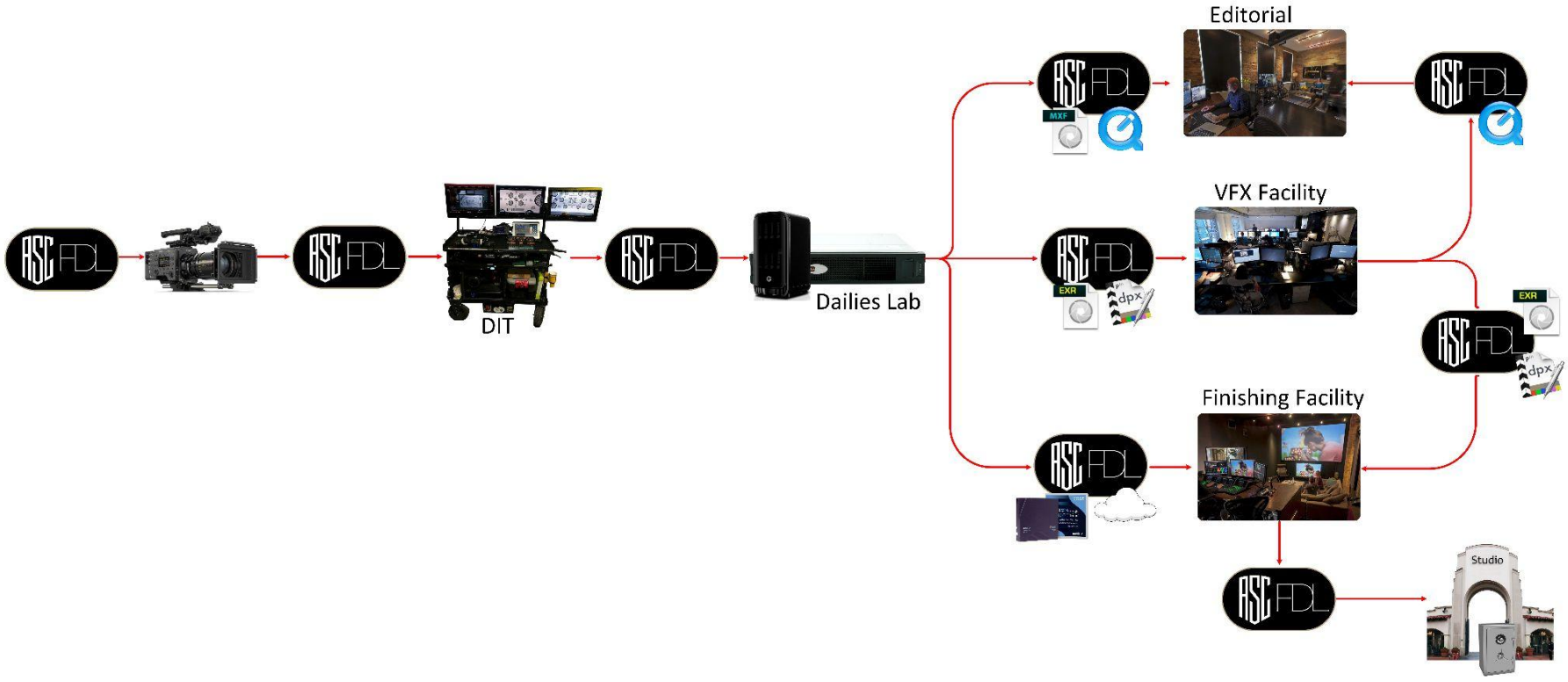
User Guide

v1.0

Table of Contents

1. Introduction

ASC FDL's are a set of instructions for how to view content in any application. The ASC FDL provides a mechanism to document framing decisions through all phases of a production's life cycle, from pre-visualization through post-production. The FDL can exist in the form of a sidecar JSON file, or embedded into another data structure like camera original files. Any time an application is rendering media to another department or person, an accompanying set of ASC FDL data should be created and delivered. Therefore any downstream applications are able to apply the intended frame.



This guide provides users an overview of the FDL JSON structure, as well as how to utilize ASC FDL's within their workflow. The specification for the ASC FDL does not dictate exactly how the FDL needs to be implemented in each software/hardware. Our goal was to clearly define the formatting of the FDL and the specific attributes within. Therefore, like a CDL, we chose to leave it up to filmmakers and implementers for how people creatively make use of it. Every production will continue to have their own unique workflows, utilizing FDLs in their own way, from on-set through post. Allowing users this flexibility was something we felt important to ensure the FDL could be used from simple framing to more complex use cases.

For a more technical breakdown of the ASC FDL please see our specification document.

2. Introduction to the FDL Structure

Step 1: Choose your Framing Intent

Every production is going to have a primary frame - the area that viewers will see in the finished product. Are you framing for 1.78:1? Maybe 2:1? Maybe multiple aspect ratios? Regardless of which camera is being used, understanding what you'll be framing for is step 1.

We call this aspect ratio the '**Framing Intent**'. A box, floating in air, unbounded by physical constraints.

Step 2: Choose your Framing Intent Protection (Optional)

Even without knowing which cameras are going to be utilized in production, you may already know how much padding/protection you want to add on your frame.

We call this the **‘Framing Intent Protection’.** A safety area around your intended frame, that could be used for stabilization, reframing, and more in post production.

You now know as an example, that you’re going to frame for 16x9, with a 10% pad.

Step 3: Select your ‘Canvas’

It’s time to choose a camera. Once a camera has been selected you’re going to decide which format to shoot in. The resolution you record in camera can be referred to as a **‘Canvas’**.

Step5: Apply the Framing Intent to the Canvas

Now that a **Canvas** has been chosen, your **Framing Intent** can be placed into the **Canvas**. Once the **framing intent** has been placed within a **Canvas**, it will now have physical dimensions/coordinates,

This frame; now minted into a **Canvas**, is now called a **Framing Decision**.

Advanced Option: Add an ‘Effective Canvas’

Depending on the camera/lens combination you choose to shoot with, you may have a lens that does not cover the full sensor of your camera. Therefore you may want to constrain certain **Canvases** to the usable area.

This protection area on your **Canvas** is called the **Effective Canvas**.

The example scenario above was for a typical production. However **Canvases** are not restricted to only being utilized for cameras. A **Canvas** could technically be used for a display, or for any image.

In this example we have two **framing intents**

One is 1.78

One is 2:1

We have one **Canvas**

Which is from an ARRI camera and is 4448:3096

Resulting in 2 **Framing Decisions**

A 1.78 Framing Decision

And a 2:1 Framing Decision


```
{
  "uuid": "DEAF6B84-6B8C-46DB-8CE3-DA5DAB8C9817",
  "version": {"major": 0, "minor": 1},
  "content_creator": "Jane Doe",
  "framing_intents": [{
      "label": "Hero 1.78",
      "id": "29A901F1",
      "aspect_ratio": {"width": 16, "height": 9},
      "is_primary": true,
      "protection": 0.05
    },
    {
      "label": "Hero 2:1",
      "id": "0302684B",
      "aspect_ratio": {"width": 2, "height": 1},
      "is_primary": false,
      "protection": 0.05
    }
  ],
  "contexts": [{
    "label": "ArriLF",
    "content_creator": "Arri LF",
    "canvases": [{
      "label": "Open Gate ARRIRAW",
      "id": "ArriLF-20210902",
      "source_canvas_id": "ArriLF-20210902",
      "dimensions": {"width": 4448, "height": 3096},
      "effective_dimensions": {"width": 4448, "height": 3096},
      "effective_anchor_point": {"x": 0, "y": 0},
      "photosite_dimensions": {"width": 4448, "height": 3096},
      "physical_dimensions": {"width": 36.7, "height": 25.54},
      "anamorphic_squeeze": 1.0,
   "framing_decisions": [{
          "id": "ArriLF-20210902-29A901F1",
          "framing_intent_id": "29A901F1",
          "framing_intent_label": "Hero 1.78",
          "anchor_point": {"x": 111, "y": 260},
          "dimensions": {"width": 4226, "height": 2376},
          "protection_dimensions": {"width": 4448, "height": 2508},
          "protection_anchor_point": {"x": 0, "y": 360}
        },
        {
          "id": "ArriLF-20210902-0302684B",
          "framing_intent_id": "0302684B",
          "framing_intent_label": "Hero 2:1",
          "anchor_point": {"x": 111, "y": 492},
          "dimensions": {"width": 4224, "height": 2112},
          "protection_dimensions": {"width": 4448, "height": 2224},
          "protection_anchor_point": {"x": 0, "y": 294}
        }
      ]
    }]
  }]
}
```

A single FDL file can also contain multiple **Canvases**

Camera A:

Camera B:


```
{
  "uuid": "BCD142EB-3BAA-4EA8-ADD8-A46AE8DC4D97",
  "version": {"major": 0, "minor": 1},
  "content_creator": "ASC FDL Committee",
  "framing_intents": [{
      "label": "2:1 Framing",
      "id": "FDLSMP01",
      "aspect_ratio": {"width": 2, "height": 1},
      "is_primary": true,
      "protection": 0.1
    }
  ],
  "contexts": [{
    "label": "ArriLF",
    "content_creator": "ASC FDL Committee",
    "canvases": [{
      "label": "Open Gate RAW",
      "id": "ArriLF-20220310",
      "source_canvas_id": "ArriLF-20220310",
      "dimensions": {"width": 4448, "height": 3096},
      "effective_dimensions": {"width": 4448, "height": 3096},
      "effective_anchor_point": {"x": 0, "y": 0},
      "photosite_dimensions": {"width": 4448, "height": 3096},
      "physical_dimensions": {"width": 36.7, "height": 25.54},
      "anamorphic_squeeze": 1.0,
      "framing_decisions": [{
          "id": "ArriLF-20220310-FDLSMP01",
          "framing_intent_id": "FDLSMP01",
          "framing_intent_label": "2:1 Framing",
          "anchor_point": {"x": 222, "y": 547},
          "dimensions": {"width": 4004, "height": 2002},
          "protection_dimensions": {"width": 4448, "height": 2224},
          "protection_anchor_point": {"x": 0, "y": 436}
      "label": "SonyVenice-6K",
      "id": "SonyVenice-6K-20220310",
      "source_canvas_id": "SonyVenice-6K-20220310",
      "dimensions": {"width": 6048, "height": 4032},
      "effective_dimensions": {"width": 5442, "height": 3628},
      "effective_anchor_point": {"x": 304, "y": 202},
      "photosite_dimensions": {"width": 5442, "height": 3628},
      "physical_dimensions": {"width": 36.7, "height": 25.54},
      "anamorphic_squeeze": 2.0,
      "framing_decisions": [{
          "id": "onyVenice-6K-20220310-FDLSMP01",
          "framing_intent_id": "FDLSMP01",
          "framing_intent_label": "2:1 Framing",
          "anchor_point": {"x": 576, "y": 792},
          "dimensions": {"width": 4896, "height": 2448},
          "protection_dimensions": {"width": 5442, "height": 2721},
          "protection_anchor_point": {"x": 303, "y": 655}
        }
      ]
    }]
  }]
}
```

A **Canvas** is also not limited to cameras. It can be the dimensions of any display, or image. As an example:

Here is a **Canvas** for VFX Plates:

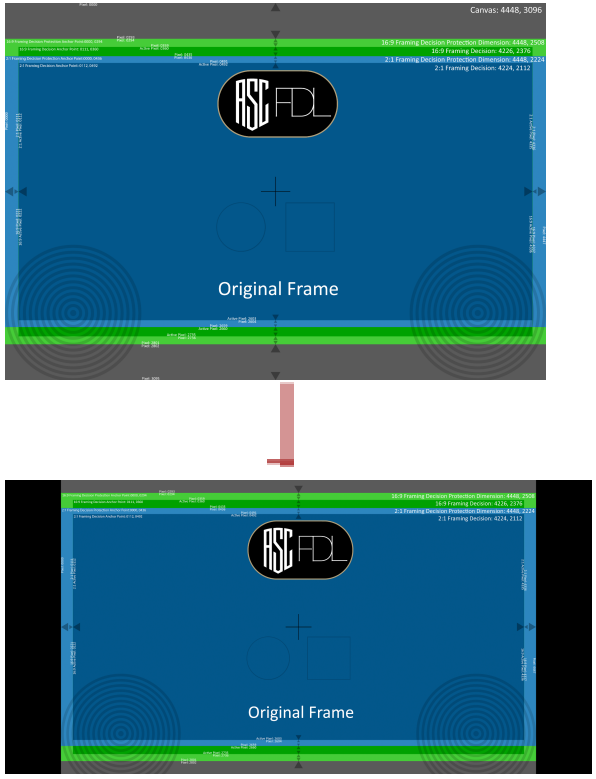
```
{
  "uuid": "BCD142EB-3BAA-4EA8-ADD8-A46AE8DC4D97",
  "version": {"major": 0, "minor": 1},
  "content_creator": "ASC FDL Committee",
  "framing_intents": [{
      "label": "2:1 Framing",
      "id": "FDLSMP01",
      "aspect_ratio": {"width": 2, "height": 1},
      "is_primary": true,
      "protection": 0.1
    }
  ],
  "contexts": [{
    "label": "ArriLF",
    "content_creator": "ASC FDL Committee",
    "canvases": [{
      "label": "Open Gate RAW",
      "id": "ArriLF-20220310",
      "source_canvas_id": "ArriLF-20220310",
      "dimensions": {"width": 4448, "height": 3096},
      "effective_dimensions": {"width": 4448, "height": 3096},
      "effective_anchor_point": {"x": 0, "y": 0},
      "photosite_dimensions": {"width": 4448, "height": 3096},
      "physical_dimensions": {"width": 36.7, "height": 25.54},
      "anamorphic_squeeze": 1.0,
      "framing_decisions": [{
          "id": "ArriLF-20220310-FDLSMP01",
          "framing_intent_id": "FDLSMP01",
          "framing_intent_label": "2:1 Framing",
          "anchor_point": {"x": 222, "y": 547},
          "dimensions": {"width": 4004, "height": 2002},
          "protection_dimensions": {"width": 4448, "height": 2224},
          "protection_anchor_point": {"x": 0, "y": 436}
      "label": "VFX-Plates",
      "id": "VFX-Plates-20220310",
      "source_canvas_id": "VFX-Plates-20220310",
      "dimensions": {"width": 6048, "height": 4032},
      "effective_dimensions": {"width": 5442, "height": 3628},
      "effective_anchor_point": {"x": 304, "y": 202},
      "photosite_dimensions": {"width": 5442, "height": 3628},
      "physical_dimensions": {"width": 36.7, "height": 25.54},
      "anamorphic_squeeze": 2.0,
      "framing_decisions": [{
          "id": "VFX-Plates-20220310-FDLSMP01",
          "framing_intent_id": "FDLSMP01",
          "framing_intent_label": "2:1 Framing",
          "anchor_point": {"x": 576, "y": 792},
          "dimensions": {"width": 4896, "height": 2448},
          "protection_dimensions": {"width": 5442, "height": 2721},
          "protection_anchor_point": {"x": 303, "y": 655}
      }
```

3. Introduction to Canvas Templates

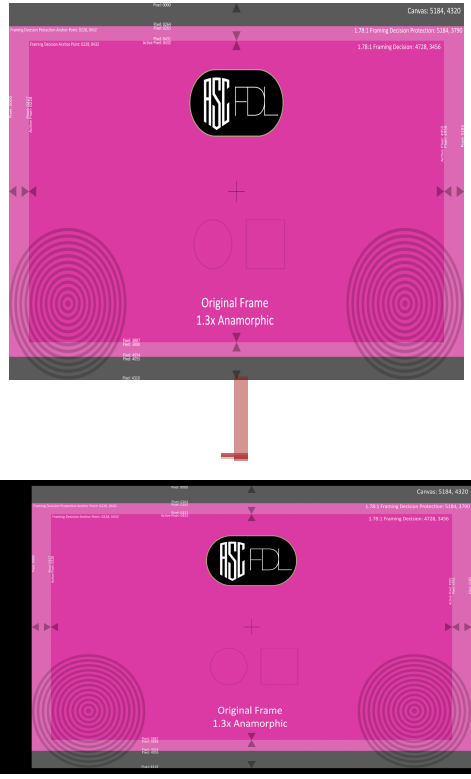
Example Scenario:

Production was shot with many different camera formats, but visual effects wants all plates delivered normalized to one common frame**.** Therefore they can apply one framing preset to ALL plates they receive.

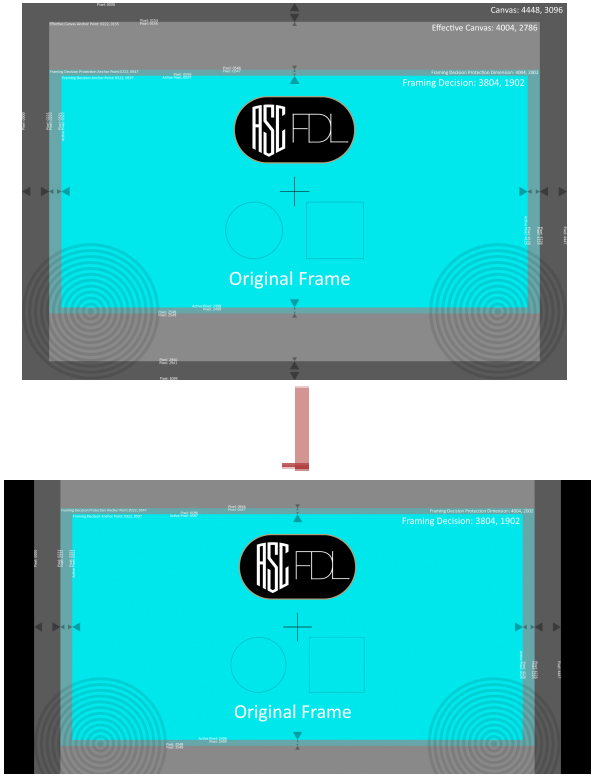
Camera A:



Camera B:



Camera C:



The FDL as it's been presented thus far can provide instructions for any application to know the intended area for viewing. However it cannot provide instructions for how to take the intended area and remap it into a new **Canvas**. With an additional set of instructions added to your FDL, this is possible through what we call a **Canvas Template**.

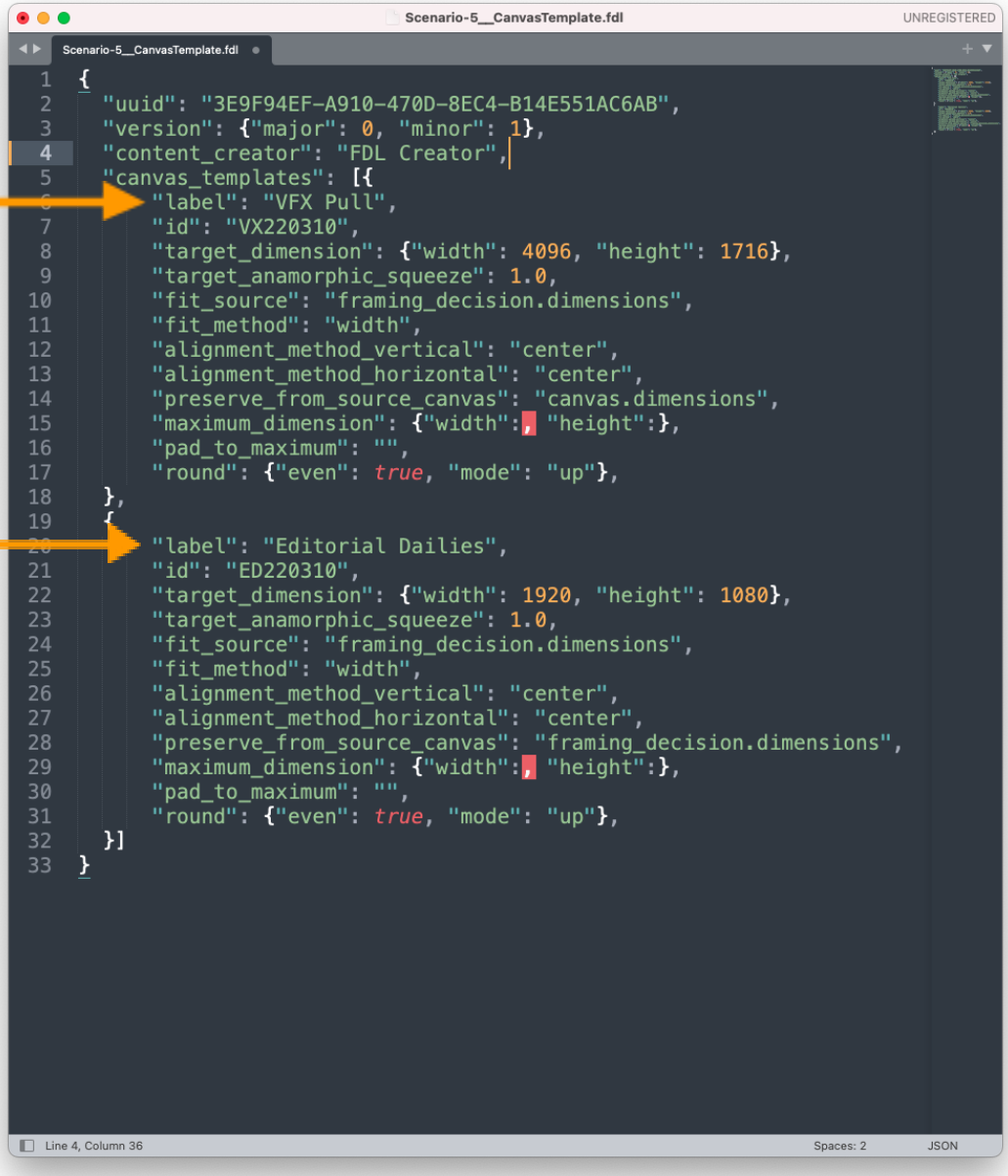
A **Canvas Template** can be appended to a standard FDL with source data

A **Canvas Template** can also be the only data within an FDL.

You will notice there are no **Canvases**, or **Framing Decisions** in this FDL. Just two **Canvas Templates**:

One Canvas Template for VFX

And one for Editorial



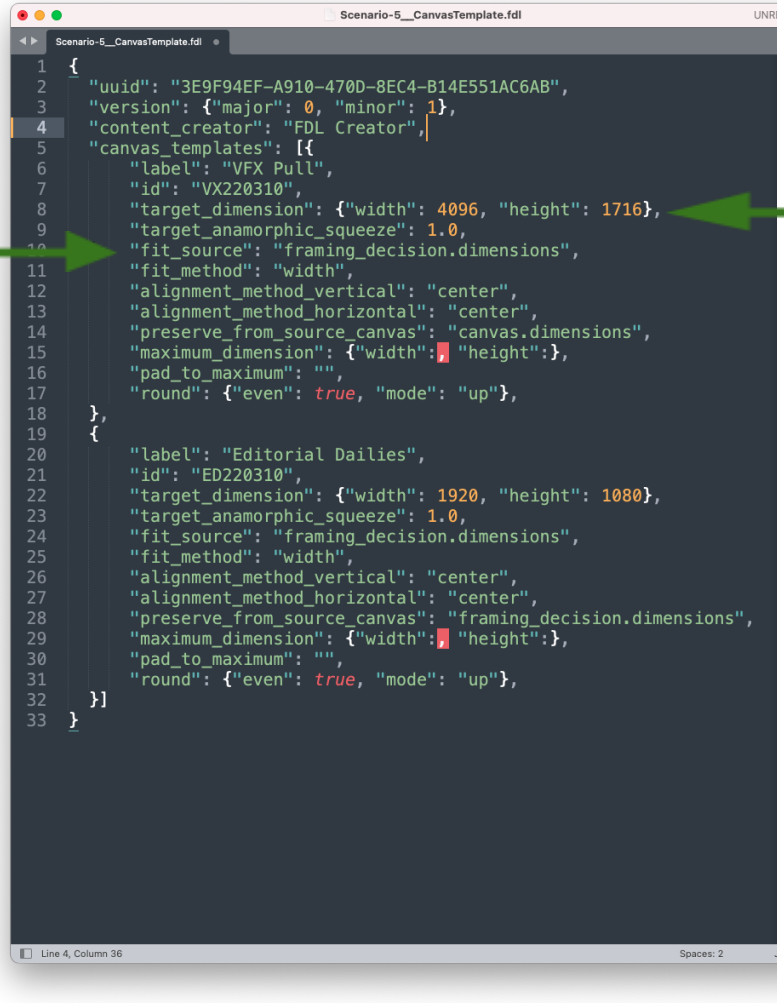
Why would you ever have an FDL with only Canvas Template data?

What if a VFX Supervisor was not aware of every camera format shot and didn't have access to the source FDL files, but they still want to provide instructions for the lab to use to normalize all VFX pulls.

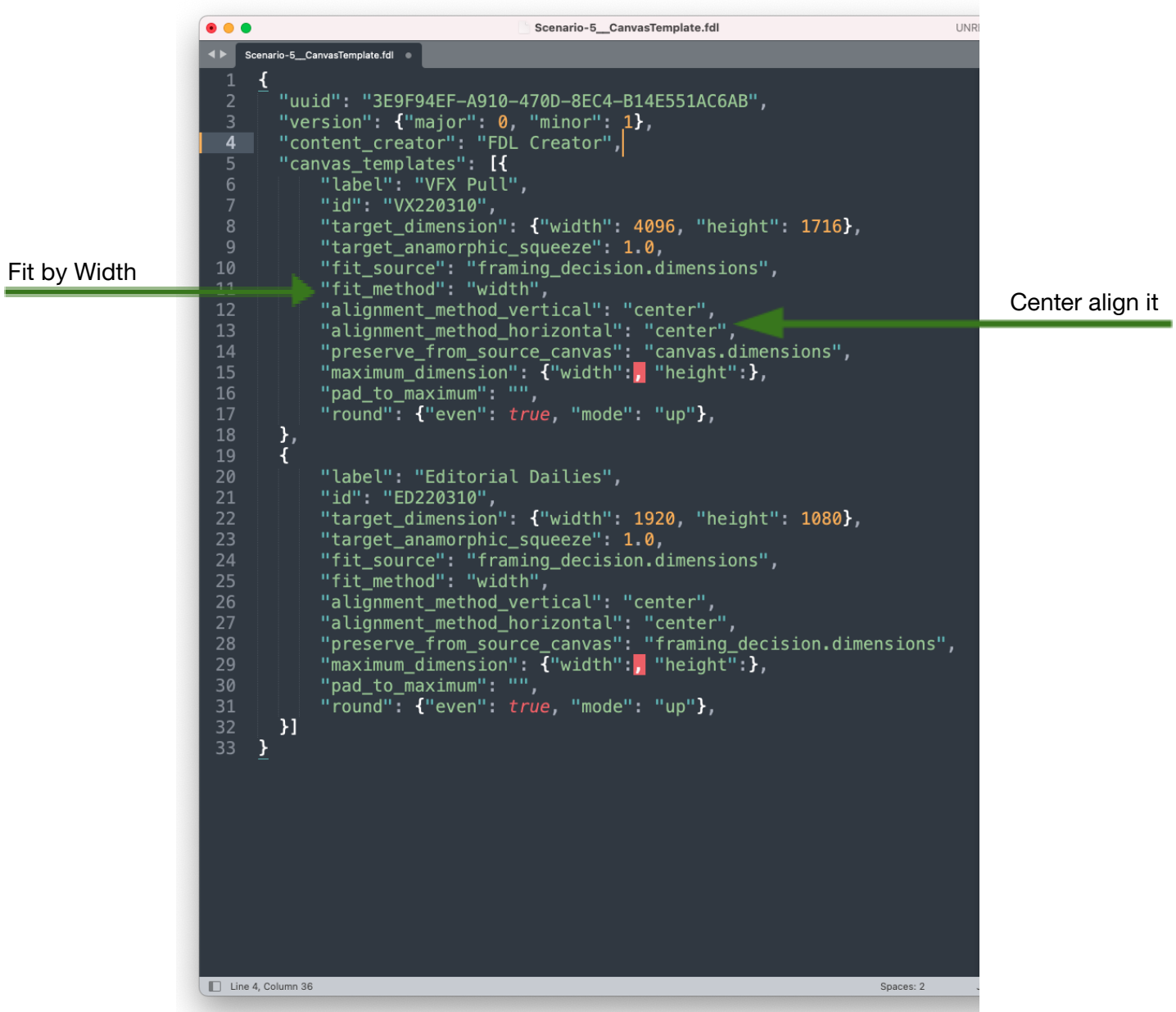
Here is an example **Canvas Template**, explaining the basic instructions for any application trying to use it.

- Find all of the source images & find all of their source FDL data. Then follow the instructions outlined in the **Canvas Template**:

Take the source files **Framing Decision**



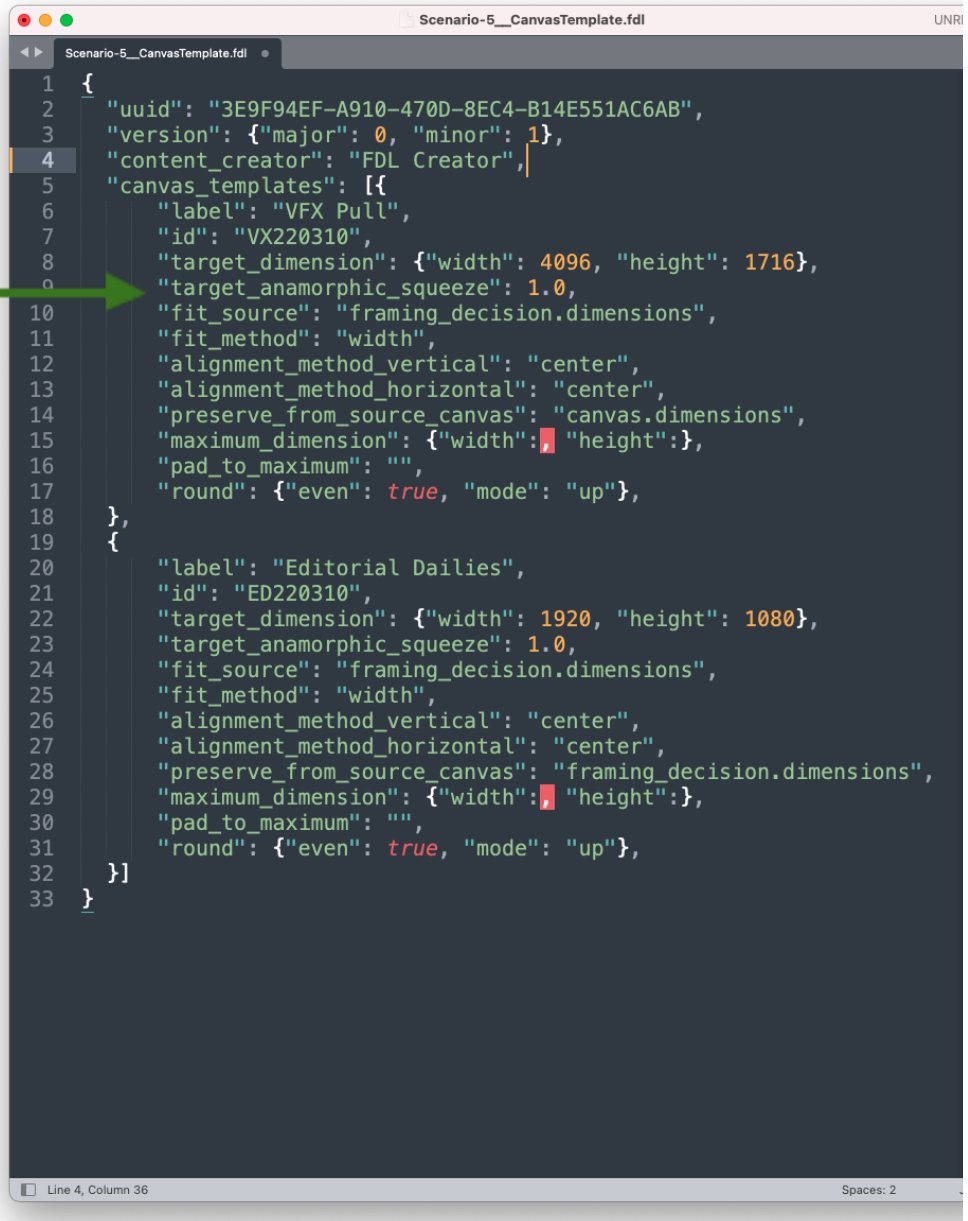
& fit it into a "Target Dimension" of 4096:1716




```
{
  "uuid": "3E9F94EF-A910-470D-8EC4-B14E551AC6AB",
  "version": {"major": 0, "minor": 1},
  "content_creator": "FDL Creator",
  "canvas_templates": [{
      "label": "VFX Pull",
      "id": "VX220310",
      "target_dimension": {"width": 4096, "height": 1716},
      "target_anamorphic_squeeze": 1.0,
      "fit_source": "framing_decision.dimensions",
      "fit_method": "width",
      "alignment_method_vertical": "center",
      "alignment_method_horizontal": "center",
      "preserve_from_source_canvas": "canvas.dimensions",
      "maximum_dimension": {"width":, "height":},
      "pad_to_maximum": "",
      "round": {"even": true, "mode": "up"},
  },
  {
      "label": "Editorial Dailies",
      "id": "ED220310",
      "target_dimension": {"width": 1920, "height": 1080},
      "target_anamorphic_squeeze": 1.0,
      "fit_source": "framing_decision.dimensions",
      "fit_method": "width",
      "alignment_method_vertical": "center",
      "alignment_method_horizontal": "center",
      "preserve_from_source_canvas": "framing_decision.dimensions",
      "maximum_dimension": {"width":, "height":},
      "pad_to_maximum": "",
      "round": {"even": true, "mode": "up"},
  }]
}
```

Make everything spherical

This means if your source was 1.0, it will remain 1.0. If your source was 1.3 anamorphic, it would be altered to 1.0. etc.



Therefore you now have a new **Canvas** that is 4096:1716 and the **framing decision** from all source files will be fit into this box. However this means for all of the source files that go through this treatment, you're cutting off everything outside of the framing decision.

Rather than just completely cut off all of area outside of the **framing decision**, maybe the user wants to allow the resulting **Canvas** to expand outward including a specific area from the source files, to be used as padding. The user can choose to "**Preserve**" the:

- Framing Decision Protection
- Effective Canvas
- Canvas

As an example:

Preserving the **Framing Decision Protection** would result in:

Therefore the **Framing Decision** has still been fit into a **target dimension** of 4096:1716, but the **Framing Decision Protection** area now makes the new **Canvas** larger than 4096:1716.

If the user chose to: Preserve the **Canvas Protection**, it would result in another expansion outward:

And lastly if the user chose to: Preserve the **Canvas**, it would result in the **Canvas** expanding outward for the entire **Canvas** to be preserved:

But just like the other example above, the inner **Framing Decision** would still remain as the defined **target dimension** of 4096:1716.

If the user chose to preserve the **Canvas** on all sources, that means the resulting **Canvases** will likely come out in all different resolutions. But again, the common inner **framing decision** would be consistent. Considering you may not know just how large any of this padding may be, you may want to put a restriction on just how large any resulting **Canvas** is allowed to be

The **Maximum Dimension** field will cut off any area outside of a defined dimension:

In this case considering this field has been left blank, there is no maximum dimension and nothing will be cut off.

Pad to Maximum:

This is another option available in the **Canvas Template**. If you have filled in a **maximum dimension**, as noted in the previous page, it will not allow any new **Canvas** to be larger than that defined resolution. However if you also set the **Pad to Maximum** to True, it will force all resulting **Canvases** to be the dimension stated in the **Maximum Dimension** field.

This could result in some resulting **Canvases** being pillar and or letterboxed if the source files do not fill the **maximum dimension** target.

4. On Set Utilization

As a DP or DIT, you will want to know what features from the FDL your camera supports. How are you creating framelines for monitoring on-set, and how are those being conveyed downstream?

We anticipate various levels of support as the ASC FDL is adopted in cameras and software across the industry. Therefore two main questions that should be asked initially are:

- Does the camera support importing an FDL? Therefore can we set frame lines up using an FDL.
- Does the camera inject ASC FDL data into the header of its recorded files?

Similar to any other workflow conversation at the top of your job, these are things to discuss. Ie; how are framelines getting set up in camera and how are you conveying them to people in post.

Getting frame lines set up in camera:

- Option 1: Create an ASC FDL with a third party application
- Option 2: Generate frame lines in-camera using traditional methods

Option 1: Creating an ASC FDL within a third party application:

If your camera supports importing an FDL, you can use a third party application to generate your FDL. This same FDL file can be provided to post production if the camera does not support passing FDL data into the header of its files.

Option 2: Generate frame lines in-camera

If the camera does not support importing FDL data into the camera, but your camera is capable of putting ASC FDL data into the header of its files, you could simply add frame lines into your camera using traditional methods. If your camera does not support injecting FDL data into the headers of its files, FDL's could still be generated within a third party application to pass along to post.

Something to also ensure you understand is the flexibility within the FDL itself. Considering FDLs can contain multiple **Canvases** and **Framing Decisions**, you could choose to put all **Canvases** & **Framing Decisions** into one concatenated FDL file. The other option is to make several FDL's, organized however you see fitting to your production. As an example, maybe you want to make a different FDL for each camera, but within that FDL has each of the various **Canvases** used in that camera. Or maybe you want every different shooting mode even from the same camera to be their own FDL's. This is all open to how you want to manage your production.

Our prediction is that like CDL's, a production will begin with a small set of FDL's and as production continues, needs will arise to create additional files. Therefore many individual FDL files may be passed between set and the dailies lab, but once post production begins we see these being managed slightly differently. It will be more common to concatenate FDL's, or FDL's will be passed around within OTIO files, or FDL's will be baked into the header of files passed between departments.

5. Dailies Utilization

Similar to some cameras having the ability to have CDL data baked into the header of files, FDL data could be as well. Alternatively, FDLs can be shared as independent .fdl files. Therefore as a dailies lab you will need to have a workflow conversation with the team on set to talk through how you will share FDL data between set and the lab.

Applying FDL's

When applying an ASC FDL to any shot, the intention is that the **Canvas Dimensions** from the FDL should match the resolution of the file. If you attempt to apply an ASC FDL to a shot that has a **Canvas Dimension** that doesn't match the resolution of the media you're applying it to, the software should prompt a warning. This warning should tell you as a user that you're attempting to apply an FDL that does not match the source imagery resolution. This should drastically reduce the opportunity for human error when applying FDL's to shots manually. Software manufacturers may also choose to use this same logic of matching the **Canvas Dimension** with the files resolution, to automate applying FDL's to shots in timelines.

Once you apply an FDL to your source imagery, the framing should automatically be adjusted visually for you by the application. Therefore, if there was a 10% punch in, the frame would have been zoomed in to reflect that. If there were multiple **Framing Decisions** in one **Canvas**, as an example a 2.39 and a 1.78 frame were monitored on set simultaneously and both are in the FDL, whichever one was labeled as the **'Primary'** in the FDL would be applied. Some software could even allow advanced settings to allow each of the various framing intents to be automatically applied to different renders you may be doing. As an example; Use the 1.78 for Editorial, and use the 2.39 for PIX.

How any software/hardware handles applying FDL's, or switching between aspect ratios after an FDL has been applied, or showing the framing values in app, is not something the specification details. Similar to the CDL, how this is handled will be up to the software and hardware manufacturers implementing the specification.

When it comes time to render, any time a new file is generated, the intention is for the application to create a new FDL for the newly rendered files. Therefore the application you're rendering from will need to be aware of where the **Framing Decision** is within the newly rendered **Canvas**, so it can provide instructions for the downstream application to view the files correctly. As an example, maybe in dailies you rendered with the reframe burnt in for editorial, but you did not burn in the matte. Therefore FDL data would be provided that tells the NLE (Non Linear Editing application) that a matte needs to be applied to view the intended frame. Alternatively if you had a 6K source file and rendered a scaled down UHD version of it, not cropping anything; An FDL should be generated that can tell any downstream application how to view this file properly.

6. Editorial Utilization

Two of the main FDL considerations for offline editorial teams are:

- Applying FDL's: Do I need to apply FDL's to view my content correctly in my editing application?
- Preserving FDL's: Ensuring FDL's are passed along downstream for conform/online editing.

Applying FDL's:

Within the ASC FDL specification there is a section that breaks down how to carry forward an ID that represents the specific FDL/**Framing Decision** that was used for any given shot, that can be put into an ALE, Avid Bin, XML, etc. This could be done by the dailies lab and therefore if your NLE supports FDL's it could apply the appropriate framing based off of this data, assuming it has access to the FDL files.

Preserving FDL's:

With the ASC FDL being a modular set of values that could be embedded in other files, our goal is to allow timeline based files to be able to share ASC FDL's per event within a timeline. With the JSON format being fairly sophisticated compared to the limitations in something like an EDL, or ALE, we're looking to newer file types like OTIO as an example to be able to carry these values to conform/online editing.

7. VFX Utilization

VFX Pulls

Framing Decisions should have already been made up stream for how the cinematographer intended the image to be viewed. But how is VFX expecting to receive their plates? Maybe they want:

- The full frame with an FDL to know how to scale, crop and reframe on their end
- Or all plates scaled down to 4K, with FDL's as instructions for viewing
- Or are all frames normalized

Our specification has given software/hardware manufacturers all of the tools they should need to offer any of the workflows noted above and more. But how should you as a user go about this process?

Option 1:

Follow whatever traditional method you have done in the past for reframing your shots in the application you're doing your pulls. As long as the application has access to the source FDL files, it should be able to understand the framing adjustment you made and where the **Framing Decision** is now positioned in your newly generated **Canvas** (plates) and it would generate an FDL per shot.

Option 2:

Create a **Canvas Template**. As noted above in the **Canvas Template** section, you can create a set of instructions that will allow the person performing the pulls to normalize all frames being rendered for VFX vendors. The application rendering the plates is going to need access to all of the source FDL data, as well as the **Canvas Template** instructions. Whether you decide to concatenate all of these into one clean concise FDL, or keep them separate is up to you.

Final VFX Shots

When VFX work has completed and files are ready to go to finishing, this is another area for a workflow decision.

- Are these files going to finishing with framing adjustments baked in?
- Or are you rendering a larger area than the **Framing Decision** and the Online Editor will need to apply a framing adjustment to see the intended frame?

Either are made possible by the FDL, as long as the FDL data is passed along from the visual effects team. These values could either be baked into the header of the final EXR frames for finishing as an example, or they could be passed along as independent files.

8. Picture Finishing Utilization

After consolidating all media and ensuring everything is online, Online Editors often have to spend time ensuring the framing of all master files in their timeline lines up with the reference quicktime from offline editorial. Within this final sequence you will have media from camera files, VFX shots and many others.

For Camera Files:
Our goal is to allow timeline based files to be able to share ASC FDL's per event. With the JSON format being fairly sophisticated compared to the limitations in something like an EDL, we're looking to newer file types like OTIO as an example, to be able to carry these values to conform/online editing. Therefore if the offline editorial team were to share an OTIO file, the FDL information for how to not only re-create the framing done by the dailies lab could be shared, but also concatenating the framing adjustments made in the offline editing application.

Alternatively the names of which FDL's were used to process which file for editorial in the dailies lab, could also be passed along for the Online Editor. At which point the Online Editing application could apply the FDL, and then use traditional methods like an AAF, or XML to apply the additional adjustments made in the offline edit.

For VFX Shots:

Reframing data will ideally be embedded in the EXRs you receive as final VFX shots. If not, sidecar FDLs will have ideally been provided by the VFX facility so you'll know how to properly reframe their shots.

** For additional technical information, please see our ASC FDL specification **