

VFX Image Sequence Naming

Addresses Plates, Comps, and other frame-based image sequences

ETC Adaptive Production VFX Working Group

Horst Sarubin, Chair

Erik Weaver, Co-chair

Craig Seidel, Executive Editor, Principal MovieLabs Liaison

Published by

The Entertainment Technology Center at The University of Southern California

Executive Summary

Currently, the range of methodologies to ingest, store, and utilize digital assets generated during the production process lacks wide industry standardization. Without a consistent rule set in place for data and asset handling, managing and organizing this ever-growing flood of data and digital assets has become increasingly more challenging. If you add the constant adoption of new technologies in the production pipeline, the current model for handling these assets and metadata has become untenable.

With interoperability and exchange at the heart of these issues, the ETC Adaptive Production group decided that the initial goal for the workgroup would be to create naming specifications for image sequences used to review and deliver composited VFX shots (often called plates or “comps”). This naming specification is meant for stakeholders in the VFX creation. Industry pain points that this naming specification tries to address are:

- **Productions** - The lack of clear interoperable specifications leads to operations having to pull resources to develop these bespoke pipelines, which are often far from ideal. This leads to slower shot turnover, higher miscommunication, and time spent solving problems as opposed to refining the creative aspects of the shots;
- **Vendors** – Vendors, who have spent years refining their production process, must spend precious time mapping each individual production’s bespoke VFX pipeline into their ecosystem. There is often a lack of consistency between productions from the same studio. Large VFX-heavy shows often require multiple vendors who need to share assets and data. Without a common spec, this process is rife with complications, leading to more time spent on solving ingest issues and locating missing data;
- **Studios** – Studios will always endeavor to get value for their investment, and the more time spent on bringing these VFX pipelines online, the less time and investment go into the final image. The studio is also increasingly interested in reusing the VFX asset for games, parks, rides, and marketing. Finding the appropriate assets and converting them into a useable format is becoming a problem.

The guiding principles for this specification are as follows:

- The data and image sequences should be delivered in a format, layout, and storage system that can be easily understood and ingested regardless of production-specific knowledge
- Standardized naming conventions and development of best practices for data handling should be enacted as the data and assets are generated
- These specifications should allow for easy parsing and ingestion while remaining agnostic to any particular pipeline or technology, where possible
- The specifications should be easy to use or implement, particularly when time and pressure are the greatest, such as the onset
- This specification should be used when assets are handed off between facilities working on a common project

Notes

We have been lucky to have most of the major VFX companies involved and most of the studios involved at one point or another. All of them were happy to give feedback while a lot of people were very excited about it because it feel like low hanging fruit. This is a living document, so if there are concerns with some aspects of it, we have left room for improvement.

Horst Sarubin
Working Group Chair
Universal Pictures

We thank the following for their contribution to this document.

ETC Member Organizations

Fox Corp
Iron Mountain Entertainment Services
Disney/Marvel Studios
Lucasfilm/ILM
Microsoft
Paramount Pictures
Technicolor
Universal Pictures
Vubiquity
Warner Bros. Entertainment

Industry Contributors

AMPAS
Adobe
Autodesk
Digital Domain
DreamWorks Animation
HBO
Legendary Entertainment
Luma Pictures
MovieLabs
Netflix
Sony Pictures Entertainment
techXmedia
Unity
VES
Weta
Ziva Dynamics

Individual Industry Contributors

Steven Anastasi
Buckley Collum
Michael Friend
Joshua Kolden
Joanne Kim
Harvey Landy
Florence Litchfield Castaneda
Arjun Ramamurthy
Steve Shapiro
Bret Shefter
Spencer Stephens
Mark Wendell

CONTENTS

1	Introduction	1
1.1	Overview.....	1
1.2	Document Organization	1
1.3	Document Notation and Conventions.....	2
1.3.1	General Notes.....	2
1.4	Normative References	2
1.5	Informative References	3
1.6	Status.....	3
1.7	Best Practices for Maximum Compatibility	3
2	VFX Image Sequence Metadata Objects	4
2.1	Version Information	4
2.2	Show ID	4
2.3	VFX Sequence	5
2.4	VFX Shot	6
2.5	Image Type.....	7
2.5.1	Type Names for Inputs to VFX (Plates)	7
2.5.2	Type Names for Other Inputs to and Outputs from VFX	7
2.6	Identifying Description.....	8
2.7	Vendor Code	9
2.8	Revision Code	9
2.9	Alternate	9
2.10	Multi-Camera Reference	10
2.11	Frame Number	10
3	File Name construction.....	11
3.1	General Syntax for File Name Encoding	11
3.2	Sequence of Metadata.....	12
3.2.1	Examples	12
3.3	Directory Naming.....	14



This work is licensed under a [Creative Commons Attribution 3.0 Unported License](https://creativecommons.org/licenses/by/3.0/).

NOTE: No effort is being made by the Entertainment Technology Center (ETC) to in any way obligate any market participant to adhere to this document. Whether to adopt this document in whole or in part is left entirely to the individual discretion of individual market participants, using their own independent business judgment. Moreover, the ETC disclaims any warranty or representation as to the suitability of this document for any purpose, and any liability for any damages or other harm you may incur as a result of subscribing to this document.

REVISION HISTORY

Version	Date	Description
1.0	June 23, 2020	Original Version

1 INTRODUCTION

The purpose of this document is to codify a set of naming best practices for VFX image sequences with aim of interoperability between production, vendors, and studio.

By our definition, *VFX image sequences* are plates, comps, and any other ‘video’ that exists as a series of single image frames. This does not generally include encoded video such as you might find in a MOV file, although some of these are supported by this specification. This can also include non-picture data, such as masks or mattes, that are also delivered as a sequence of frames.

This document sets out to establish a systematic approach towards the naming, which will allow more automated ingestion, tracking, and tagging of the image files that comprise an image sequence used in the VFX ecosystem of media creation.

When we talk about plate file naming, what we’re really talking about is plate *identification*. How is the plate different from all other plates? It turns out that certain characteristics of the plate (i.e., metadata) are sufficient to uniquely identify them. This document identifies those essential metadata objects and defines how to encode them.

Note that there are many other plate metadata objects that are essential to the VFX process but are not necessary for purposes of identification. These are discussed elsewhere, such as in the [VES Transfer Specification](#). The VES Transfer Specification requires that plate naming be specified. We recommend this specification be referenced.

From the standpoint of this document, an image sequence is any sequential set of images. These can be delivered either as groups of individual images (e.g., DPX or OpenEXR) or as video (e.g., MOV).

This specification addresses file names as a human-readable form of identification. This is not to the exclusion of other identifiers. There are numerous IDs associated with these image sequences, such as C4ID which uniquely and reproducibly identifies the content of the file, and EIDR, which references the movie, season, series or episode of which the sequence is a part.

1.1 Overview

This document defines an approach for file naming of VFX image sequences, as defined in the introduction. This has two parts:

- Purpose and syntax for each metadata parameter associated including controlled vocabulary standard terms.
- Method for encoding these parameters into file names.

1.2 Document Organization

This document is organized as follows:

1. Introduction – Provides background, scope and conventions

2. VFX Image Sequence Metadata – Defines the metadata and encoding for individual elements
3. File Name Encoding – Defines how file names are constructed from metadata.

1.3 Document Notation and Conventions

Terms defined to have a specific meaning within this specification will be capitalized, (e.g. “Track”), and should be interpreted with their general meaning if not capitalized.

Strings are specified using the following conventions

- Plus sign (“+”) means concatenation
- Curly braces (“{”, “}”) contain optional information
- Angle brackets (“<”, “>”) contain a named parameter that is described elsewhere
- Quoted objects are included literally
- Parentheses (“(”, “)”) indicate grouping
- Vertical Bars (“|”) indicate a choice between what is on the left and what is on the right.

For example:

<example> = <prefix> + “-” + <suffix> {+ “@” + <more>}

- <prefix> is an alphabetic prefix
- <suffix> is a numeric suffix
- <more> an alphanumeric addition that defines strings like ‘abc-123’ and ‘x-23@stuff’

1.3.1 General Notes

All required elements and attributes must be included.

When enumerations are provided in the form ‘enumeration’, the quotation marks (‘’) shall not be included.

There is a general tradeoff between clarity and brevity. As filename encoding is required, brevity often wins—long file names are often impractical.

1.4 Normative References

[EXRFILE] Kainz, Florian, *OpenEXR File Layout*, March 13, 2013,
<https://www.openexr.com/documentation/openexrfilelayout.pdf>

- [ST268] SMPTE ST 268:2014, *File Format for Digital Moving-Picture Exchange (DPX)*, 2014, <https://ieeexplore.ieee.org/document/7292028>

1.5 Informative References

- [C4ID] Kolden, Joshua, *C4 Identification System*, Entertainment Technology Center, 2015, <https://www.etcentric.org/wp-content/uploads/2015/09/C4-ID-ETC-Whitepaper.pdf>
- [DCNC] Digital Cinema Naming Conventions, <https://isdcf.com/dcnc/>
- [EIDR] Entertainment Identifier Registry (EIDR), <http://eidr.org/technology/>
- [VES-TS] Visual Effects Society (VES) Transfer Specification, <https://vestransferspec.org/>

1.6 Status

This specification is ready for implementation. As requirements evolve, we anticipate that the identification of additional use cases will motivate changes. Implementers should anticipate future revisions. Reasonable measures will be taken to ensure changes are backwards compatible.

1.7 Best Practices for Maximum Compatibility

Metadata typically evolves with the addition of new elements, attributes and vocabularies. Existing applications should be capable of accepting metadata, even though there might be more data than expected.

Metadata specifications and schema updates are designed to support backwards compatibility. For example, elements and attributes can be added, but required elements are not removed; or more generally ordinality of elements and attributes can be widened but not narrowed. Values are not changed in either syntax or semantics. Therefore, we strongly encourage implementations to either be diligent in tracking to the latest version or follow the backwards compatibility rules provided here.

We will try to update metadata definitions such that following these rules work consistently over time. Sometimes, changes must be made that are not always backwards compatible, so we will do our best to note these.

2 VFX IMAGE SEQUENCE METADATA OBJECTS

This section defines individual metadata objects that are required for identification.

As noted above, there are many metadata objects that are essential to the VFX process but are not necessary for identification. These are addressed elsewhere.

2.1 Version Information

Version Information identifies the version of this specification used for encoding. Parsers may need this information to ensure they are parsing correctly.

This specification will likely evolve over time. Any given implementation will comply with one version of this specification. For purposes of validation and interpretation it is useful to know the version of the spec that was used when encoding.

<spec-version> = “sv” + <major-version> + “.” + <minor-version>

- <major-version> is two alpha-numeric digits (with leading zeros if necessary) corresponding with the version left of the decimal point
- <minor-version> is two alpha-numeric digits (with leading zeros if necessary) corresponding with the version left of the decimal point

For example, version 1.0 would be encoded ‘sv01.00’, version 1.2 would be ‘sv01.02’.

2.2 Show ID

The Show ID is an abbreviated code that uniquely identifies a feature, episode or other work. The term “show” is used broadly to cover what is often called “title” or “project”.

Show ID is both human and computer readable. Care must be taken to ensure the Show ID is unique across all titles.

Examples of Show ID include ‘un-ff25’, ‘sw-solo’, and ‘gotr-s08e007’.

It has forms for individual work (e.g., features or shorts) and for episodic series (aka ‘shows’).

Part of the Show ID is an optional <studio-or-franchise> where

- <studio-or-franchise> is an optional alphanumeric identifier for the studio or franchise. This is used to address situations where more than one studio uses the same <show> values. Recommendation is 2 characters. This should only be used where there is concern it might collide with other Show ID values (i.e., from other studios).

In the examples above, ‘un-ff25’, ‘sw-solo’, the <studio-or-franchise> is ‘un’ (Universal Pictures) and ‘sw’ (Star Wars), respectively.

Note that although the Show ID can have multiple parts it is still one field. Because it is one field, parts are separated by dashes (“-”) which are effectively white space inside the field.

2.2.1.1 Individual works

For features, Show ID is encoded as follows:

- `<showid> = {<studio-or-franchise> + “-” +} <show>`
 - `<show>` is a 4 alphanumeric character identifier of the features

For example, ‘ff25’ could be used for *Fast and Furious 25*; ‘solo’ for *Solo*, ‘rasp’ = Rasputin (the code name for Iron Man 3).

If `<studio or franchise>` were included, it might be ‘sw-rasp’ or ‘un-ff25’; representing Star Wars (‘sw’) franchise or NBCU (‘un’) studio respectively.

2.2.1.2 Episodic

For an episode, the Show ID is a combination of the series and episode. It is encoded as follows:

- `<showid> = {<studio-or-franchise> + “-” +} <production sequence>`
- `<production-sequence>` is an alphanumeric production sequence identification.
 - Generally, `<production-sequence> = <series> + “-” + “s” + <season> + “e” <episode>`
 - `<series>` is a 4-character alphanumeric identifier of the series
 - `<season>` is 2 numeric digits identifying the season, with leading zeros.
 - `<episode>` is 3 numeric digits identifying the episode, with leading zeros.

An example of `<series>` is ‘gotr’, which identifies *Game of Thrones*. Note that ‘got’ is not valid because it is only three characters.

`<series>` is followed by a dash, then season and episode. For example, ‘gotr-s08e007’ or ‘hb-gotr-s08e007’ would refer to Game of Thrones Season 8 Episode 7.

Note that episode is a production sequence that might not correspond with airing sequence.

2.3 VFX Sequence

The VFX Sequence¹ is an abbreviation of the sequence name which is generally an easily remembered description of a sequence in the film.

An example of the VFX Sequence is ‘sb’ for the stone ball sequence in *Raiders of the Lost Ark*.

¹ VFX Sequence is distinct from VFX Image Sequence. The first refers to the sequence itself. The latter refers to a sequential images.

That sequence is typically organized by the unity of the VFX work. VFX environments, large scale effects, and particularly difficult VFX tasks all inform how to organize the sequence. VFX Sequence is typically designated by the production VFX staff.

VFX Sequence is encoded as follows:

- `<vfx-sequence>` is 2-4 alphanumeric characters.

VFX Sequence that apply to a well-defined portion of the work is encoded as one of the following:

- “show” – complete show
- `<portion>` + `<portion-number>` as follows:
 - `<portion>` is one of the following:
 - “act” – references an act
 - “reel” – references a reel
 - “part” – references a part of the show
 - `<portion-number>` should be the same number of digits for all show-related deliveries. For example, if there are 5 reels, 1 digit is acceptable. If there are 20 parts, two digits should be used. It should be zero-padded to ensure the correct number of digits.

One must avoid using `<vfx-sequence>` values starting with the vocabulary listed above (“show”, “act”, “reel” or “part”) if it is not one of those designators. Sequences with “act” are called *act designators*.

For example, the iconic rolling stone ball sequence from *Raiders of the Ark* had VFX, might be encoded as ‘sb’ (i.e., “Stone Ball”). If the work was only meant for act one it would be encoded ‘act1’.

2.4 VFX Shot

VFX Shot, also known as VFX Shot Number or VFX Shot ID, uniquely identifies the VFX Shot within the VFX Sequence (defined in Section 2.3). It must be unique within that sequence.

- `<vfx-shot>` is a 4-digit number. It organizes these shots sequentially. “0000” is reserved (see below).

For example, given the “Stone Ball” VFX Sequence, one might use ‘sb_0100’.

The value “0000” is used when the concept of ‘shot’ does not apply. It might apply to something like a model, or an effect that spans shots. For example, ‘sb_0000’ applies to the entire Stone Ball sequence. All of Act 1 would be encoded as ‘act1_0000’. The entire work would be encoded as ‘show_0000’.

2.5 Image Type

Image Type indicates the purpose of the image sequence. For example, it could signify if the image sequence is a main plate, a base plate, a composite, look dev, etc.

The lists below are not comprehensive. For the types given, the controlled vocabulary should be used. For types not anticipated by this specification, use of a custom term is acceptable.

Image Type is formatted as follows

<type-name> + {<instance-number>}

- <type-name> values are listed in the following subsections.
- <instance-number> is an optional 2-digit, zero-padded number differentiating image sequence.

An image sequence with a different number is a distinct plate (i.e., not a different version). For example, EL01 might be fire, and EL02 might be smoke.

2.5.1 Type Names for Inputs to VFX (Plates)

In the context of this document, “Plates” are image sequences derived from images captured in photographic processes. This is not intended to be a precise definition, but differentiates plates from certain other image sequences (e.g., pre-vis).

The following values are consistent with VES Transfer Specification [VES-TS].

For Plates, <image-type> shall have one of the following values:

- “mp” – Main Plate when only one element is required to produce the final VFX work
- “bg” – Background plate for VFX work requiring multiple elements. Background plates layer identification. For example, ‘BG’ or ‘BG02’
- “fg” – Foreground plate for VFX work requiring multiple elements. Foreground plates include layer information. For example, ‘FG12’
- “el” – Element plate for VFX work requiring multiple elements. This include isolated elements of the shot, either partial composites in multi-vendor shots, or rotoscoped elements and CG elements delivered at productions request
- “cp” – Clean Plate to be used for cleanup work
- “rp” – Reference Plate such as lighting reference

Note: There is some redundancy between fg and el. Where ambiguous, choice is user-preference.

2.5.2 Type Names for Other Inputs to and Outputs from VFX

This section describes other (i.e., non “plate”) image sequences.

Note that many of these can be delivered as image sequences or as videos (e.g., QuickTime). This naming is intended to apply to the image sequences.

For image sequence other than plates, <image-type> shall have one of the following values:

- “anim” – animation version, sent for animation review can be a playblast, or if previously discussed, animation over green for editorial to comp
- “comp” – composited VFX shots
- “cnpt” – Concept
- “FINAL” – Special designation for final comp
- “layo” – Layout versions, generally sent for environment approval
- “lddv” – Look dev
- “matt” – Mattes and masks
- “prev” – Pre vis
- “prox” – Proxy (definition provided by user)
- “pstv” – Post vis
- “repo” – repositioning, sent for editorial approval
- “spfx” – effects pass, also can be a submitted as a playblast, or if previously discussed, animation over green for editorial to comp
- “test” – test shots, slap comps
- “time” – retime approvals, sent for editorial approval
- “vcam” – virtual camera
- “other-” + <custom-code> – Other custom category not addressed here.
<custom-code> is any code defined by parties exchanging data. For example, ‘other-magic’

2.6 Identifying Description

While parameters such as VFX Sequence, VFX Shot, and Image Type provide a formal mechanism for referencing an image sequence, there is often a need for additional non-standardized information to identify that image sequence.

Identifying Description, referred to as <identifying-description>, contains the information needed to provide context to the image sequence, such as what has been done in the latest pass. For example, Identifying Description might highlight the subject of the image sequence (e.g., character name), a specific effect performed on an image (e.g., facial hair removal), or non-show

generated reference material (e.g., reference footage showing car race). Any description that would be needed to fully understand the image sequence would be placed here.

Identifying Description shall contain only alphanumeric characters and dash (“-”). The intent is for words to be separate by the dash.

It is important that the optional description stays consistent throughout the production. If something is labeled ‘doms-beard’ that naming must remain constant (e.g., do not sometimes use ‘doms-beard’ and other times use ‘doms-facialhair’ or ‘torretos-beard’).

2.7 Vendor Code

<vendor-code> is two alphanumeric characters that uniquely identify a vendor. It is typically assigned by the production VFX department. It should be encoded for readability. For example, ‘dd’ could be used to identify Digital Domain or ‘im’ could be used to identify Industrial Light and Magic.

2.8 Revision Code

Revision Code represents the version control for the image sequence described. This represents a revision of the same version of the image sequence. That is, as an image sequence iterates, it gets a new revision code.

<revision-code> = “v” + <nnn>

- <nnn> is at least 3 numerals corresponding with the revision. There must always be the same number of numerals, so pad with zeros when necessary. For example, if using 3 digits, the revision 10 would be represented as ‘v010’. With 4 digits, it would be ‘v0010’.

Use the maximum number of digits to accommodate the worst-case number of digits potentially required for the production. For example, if values greater than 999 are possible, use 4 digits.

2.9 Alternate

There can be multiple versions or variations of a given image sequence that need to be differentiated. For example, 4K and 6K variants of the same plates might be provided. This ensures the different versions have different filenames.

If only one version exists, this should not be included. For example, if there is a 4K version and no 6K version this field is not needed.

The primary (best) version should not include this field. For example, if there is a 4K version and the 6K version and the 6K version is the one intended for use, the 6K version will not have this field and the 4K will. If they are equivalent, just choose one to be primary. Note that this is intended to support workflows where alternate versions show up later.

All alternate versions should have an <alternate> tag as follows:

- <alternate> = “a” + free text

For example, “a4k” for 4K or “ardeep” for deep bit depth.

2.10 Multi-Camera Reference

Two or more cameras might be synchronized in a shot; for example, a stereo pair or a camera array. Camera reference provides for the association of images to cameras.

Camera reference refers to the camera that captured the frame when more than one camera is simultaneously filming. This should not be used when only one camera is used.

<camera-reference> = ‘cam’+ {“l” | “r” | “c”| <camera-label>} where

- “l” represents left eye of stereo pair
- “r” represents right eye of stereo pair
- “c” for center camera
- <n> represents at least 1 alphanumeric characters identifying the camera. For example, ‘cama’ and ‘camb’, or ‘cam11’ and ‘cam12’.

When using digits, it is recommended that it always be the same number of numerals with zero padding as necessary. For example, if using 2 digits, the camera 10 would be represented as ‘cam10’. With 4 digits, it would be ‘cam0010’.

2.11 Frame Number

Frame number is the number of the frame in the image sequence.

<frame-number> = <nnnn>

- <nnnn> is at least 4 numeric digits with leading zeros associated with the frame number. There must always be the same number of numerals, so pad with zeros when necessary. For example, if using 4 digits, the revision 1001 would be represented as ‘1001’. With 5 digits, it would be ‘01001’.

Frame number is a positive integer.

By convention, frame number frequently starts with 1001.

<frame-number> shall be present when the file contains a single frame (e.g., DPX or OpenEXR file).

<frame-number> shall not be present if the file contains all the frames (e.g., a MOV file)

If more than four digits are used, then that number of digits must be used across the show. For example, if a show intends to use four digits, pad numbers less than 999 with leading zeros (e.g., ‘1102’ and another ‘2153’). Use the maximum number of digits to accommodate the worst-case number of revisions in the production.

3 FILE NAME CONSTRUCTION

Many workflows depend on file naming to find relevant assets. To accommodate those workflows, this section describes the method for encoding metadata in the filename.

Consistent encoding makes it easier to read and sort filenames.

Note that practice² has shown that automated parsing of filenames to extract metadata is risky. We strongly recommend only using filename encoding when the workflow requires humans to find assets using these names. To facilitate migration away from file name encoding, we also recommend always encoding metadata in the file using one of the other methods.

3.1 General Syntax for File Name Encoding

The rules are as follows

- Only allowed characters are alphanumeric, 0-9, “-” and “_”. Note that filesystems generally support UTF-8 alphanumeric characters. When filesystems that do not support full UTF-8 are used (e.g., FAT), limit text to ASCII.
 - Period (“.”) may be used prior to the Frame Number. When using this convention, omit ‘f’ from the Frame Number
 - Note that white space is not allowed
- Metadata parameters are separated by an underscore (“_”).
- Underscores (“_”) are not allowed within a metadata parameter.
- Typically, dashes (“-”) are used to replace whitespace. For example, ‘this-is-an-example’.
- If a parameter is missing, underscores must still be present as if it were there. For example, given `parameter1_parameter2_parameter3`, if `parameter2` was missing it would be encoded as `parameter1__parameter3`
- Case
 - Use of all lower case is preferable.
 - Case shall be used consistently. This ensures that file name sorting works consistently. For example, if ‘dd’ is used, ‘DD’, ‘Dd’ and ‘dD’ should not be used.
 - Case shall not be used to distinguish terms (e.g., ‘dd’ = ‘DD’).
- Filenames more than 255 characters should be avoided. Note that exFAT filename limit is 256 and Linux is 255.

² An example of filename metadata encoding in this industry is Digital Cinema Naming Conventions [DCNC]. Based on experience, DCNC also prohibits automated parsing of file names to extract metadata. They are also transitioning to metadata embedded in the file (CPL metadata).

File type is considered part of the file name.

- <file-extension> is the conventional extension for the file type (e.g., 'exr' for an OpenEXR file).

3.2 Sequence of Metadata

Files are to be named as follows:

```
<showid> + “_” +  
<vfx-sequence> + “_” +  
<vfx-shot> + “_” +  
<image-type> + “_” +  
<vendor-code> + “_” +  
<revision-code>  
{ + “_” + <alternate>}  
{ + “_” + <camera-reference>}  
{ + “_” + <identifying-description>}  
{ + “_” + <spec-version>}  
{ + “.” + <frame-number>}  
+ “.” + <file-extension>
```

Note that although this specification is intended for VFX Image Sequences of individual frames, the convention can also be used for files that contain multiple frames (e.g., MOV files). Only in this case, is the frame number omitted.

3.2.1 Examples

The examples in this section illustrate common usage and variants.

Certain information is constant through most of the examples because variations are not that illustrative.

- All files use .exr extension. If other formats are used, the extension would be different
- Examples are for Fast and Furious 25. Show ID is 'ff25'. It could also be 'un-ff25'.
- The sequence 'ib'.
- Shot is 1, but by convention numbering starts at 100
- Vendor is 'dd'
- Everything is revision 1 (i.e., 'v001')
- For brevity, we're using three frame sequences (1000, 1001, and 1002).

Here is a set of background plates:

un-ff25_ib_0100_bg_dd_v001.1000.exr

un-ff25_ib_0100_bg_dd_v001.1001.exr

un-ff25_ib_0100_bg_dd_v001.1002.exr

Fire and smoke element plates:

un-ff25_ib_0100_el_dd_v001-fire.1000.exr

un-ff25_ib_0100_el_dd_v001-fire.1001.exr

un-ff25_ib_0100_el_dd_v001-fire.1002.exr

un-ff25_ib_0100_el_dd_v001-smoke.1000.exr

un-ff25_ib_0100_el_dd_v001-smoke.1001.exr

un-ff25_ib_0100_el_dd_v001-smoke.1002.exr

Examples of other image sequences (only one frame shown):

- Comp, revision 20

un-ff25_ib_0100_comp_dd_v020.1000.exr

- Concept, Dom's beard correction

un-ff25_ib_0000_cnpt_dd_v001_dom_beard.1000.exr

- Ice charger concept that applies to all of Act 3

un-ff25_act03_0000_cnpt_im_v020_ice_charger.1000.exr

- Generic animation for the sub explosion, revision 20.

un-ff25_ib_0000_anim_dd_v020_sub_explosion.1000.exr

- VFX ID for look dev and old age version of the character Dominic Toretto

- Used throughout the film.

un-ff25_show_0000_ldev_dom-old-age_dd_v001.1000.exr

- – Was only meant for act one

un-ff25_act1_0000_ldev_dom-old-age_dd_v001.1000.exr

- Where work was generic for a sequence in the film but not shot specific you would use the appropriate sequence code but still the generic number code. 'sc' is used for this sequence in the production named "Scooter Chase".

un-ff25_sc_0000_ldev_dom-old-age_dd_v001.0001.exr

The following example shows a plate, a related matte plate and an updated version of the matte plate. In the context, the original plate is often referred to as the 'parent'.

- The main plate, version 23

un-ff25_ib_0100_mp_dd_v023.1001.exr

- The matte plate (type = 'matt'). It is revision 1, but refers to revision 23 of the main plate. The reference to the main plate (i.e., parent) is in the description field,

'mp-v023'. Note the use of dash because the description is one field.

un-ff25_ib_0100_matt_dd_v001_mp-v023.1001.exr

- If the matte is revised, the revision would be updated to 'v002'

un-ff25_ib_0100_matt_dd_v002_mp-v023.1001.exr

3.3 Directory Naming

File naming is independent of directory path naming. That is, the file name remains the same, regardless of whether some elements are redundant with the directory path.

For example, consider this file: FF25_ib_0100_BG_dd_v020.1000

The file name remains the same in these two cases:

resource/ff25_ib_0100_bg_dd_v020.1000.exr

resource/ff25/ib_0100/bg/dd/v020/FF25_ib_0100_bg_dd_v020.1000.exr

The following is *incorrect*:

resource/ff25/ib_0100/bg/dd/v020/1000.exr