ARRI ALEXA 35
ARRIRAW HDE Transcoder 1.6.4

Quick Guide

Date: 5 Nov 2025

1 About

This quick guide is an introduction to the ARRIRAW HDE Transcoder—a software encoder that converts ALEXA 35, 35 Xtreme or ALEXA 265 MXF/ARRIRAW files (SMPTE RDD 54/55) to MXF/HDE files.

2 ARRIRAW and High Density Encoding

2.1 ARRIRAW Recording File Types

ALEXA XT, SXT, LF and 65 cameras are recording ARRIRAW data as a sequence of individual '.ari' files. Each file containing a complete set of static (per-take) and dynamic (per-frame) camera and lens metadata.

ALEXA Mini, AMIRA and ALEXA Mini LF package everything into one MXF file per take. This removes metadata redundancies and significantly improves the efficiency when the original camera files need to be moved or archived.

With the introduction of the ALEXA 35 and REVEAL Color Science, we enhanced the image processing chain and fully aligned it with the metadata handling in MXF files. This led to a restructured, metadata-rich MXF container. Mapping of the ARRIRAW bitstream and carriage of ARRI system metadata in MXF files are defined in SMPTE RDD 54 and 55. The new MXF format is used by the ALEXA 35, 35 Xtreme and the ALEXA 265.

2.2 CODEX High Density Encoding (HDE)

Working with uncompressed ARRIRAW data presents major challenges in terms of storage and data transfer. Recognising these challenges, CODEX has developed High Density Encoding*—a lossless, variable bitrate compression, specifically optimized for Bayer pattern (ARRIRAW) images. HDE reduces the size of ARRIRAW files by over 40% without compromising image quality or metadata. Comparing the processed output of an HDE file and its original ARRIRAW file will result in a a bit-exact match.

CODEX originally encoded HDE files into frame-based '.arx' sequences for all cameras up to ALEXA Mini LF. With the introduction of REVEAL Color Science, they also adopted MXF output, which is now available for ALEXA Mini, AMIRA, ALEXA Mini LF, ALEXA 35, ALEXA 35 Xtreme and the ALEXA 265.

The ARRI image processing chain treats ARRIRAW and HDE files in the same way. Since most ARRIRAW-compatible software can also decode HDE files, users enjoy a seamless experience with significantly reduced storage requirements and transfer times. There is also no noticeable impact on real-time performance, which is why HDE files are regularly used as the camera master.

2.3 How to HDE

HDE files cannot be recorded on the camera itself; you first record ARRIRAW camera originals and then convert the files. **The standard HDE workflow relies on CODEX Device Manager,** which runs a virtual file system as macOS background application.

When original camera recording media containing ARRIRAW data is mounted, the software automatically presents a virtual volume with HDE files. These files can be offloaded using a compatible copy tool (see the CODEX Help Centre), which makes the process as simple as offloading footage from recording media.

If this workflow cannot be used, ARRIRAW HDE Transcoder offers an alternative. This may happen when:

- Camera originals were offloaded as ARRIRAW files instead of HDE
- No Mac was available when the media had to be cleared
- Clips were trimmed or repaired in the command-line ARRI Reference Tool, which can't export to HDE

NOTE: ARRIRAW HDE Transcoder only supports "new style" MXF/ARRIRAW files (RDD 54/55) from ALEXA 35, ALEXA 35 Xtreme, and ALEXA 265.

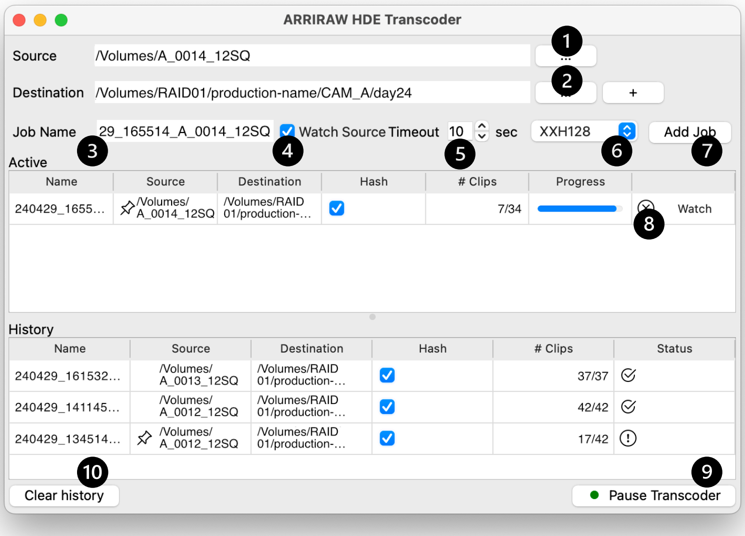
* In recognition of their achievement, the Academy of Motion Picture Arts and Sciences honored James Eggleton and Delwyn Holroyd with a Technical Achievement Award in 2024 for their invention of High Density Encoding.

3 ARRIRAW HDE Transcoder

The ARRIRAW HDE Transcoder was created for situations where CODEX Device Manager could not be used. While it does support HDE encoding directly from the recording media, it was not intended as a replacement for the standard workflow with dedicated data wrangling tools and therefore lacks most advanced features these tools would offer. The ARRIRAW HDE Transcoder can:

- Read MXF/ARRIRAW data from any source, including directly from camera media
- Apply HDE using the same CODEX engine (ensuring identical results)
- Write MXF/HDE files to several destinations
- Generate an initial hash list/checksums (MHL or ASC MHL) for downstream verification
- Run with a graphical interface on macOS and Windows, or as a command-line tool on macOS, Windows, and CentOS

3.1 Graphical User Interface


ARRIRAW HDE Transcoder

Source: /Volumes/A_0014_12SQ

Destination: /Volumes/RAID01/production-name/CAM_A/day24

Job Name: 29_165514_A_0014_12SQ ✓ Watch Source Timeout 10 sec XXH128

Active

Name	Source	Destination	Hash	# Clips	Progress	
240429_1655…	/Volumes/ A_0014_12SQ	/Volumes/RAID 01/production-…	✓	7/34		Watch

History

Name	Source	Destination	Hash	# Clips	Status
240429_161532…	/Volumes/ A_0013_12SQ	/Volumes/RAID 01/production-…	✓	37/37	✓
240429_141145…	/Volumes/ A_0012_12SQ	/Volumes/RAID 01/production-…	✓	42/42	✓
240429_134514…	/Volumes/ A_0012_12SQ	/Volumes/RAID 01/production-…	✓	17/42	!

Setting up a Job

(1) Select a `Source` folder which may contain clips or a folder containing more than one reel of clips. All other files (clips in other codecs, ale, bin, and non-camera data) will be copied to the destination.

(2) Select a `Destination` folder where the output will be written. The software will mirror the structure under the source directory. Add/remove destinations with + or –.

(3) The `Job Name` is automatically created based on the source folder name and can be changed.

(4) The `Watch Source` option tells the software to encode all files in a folder and then keep monitoring for new files. A *watch job* is identified with a pin in the `Active` job list.

(5) Use the `Timeout` setting in case the application produces a `not enough data` error message. This may be the case the source cannot be accessed for a longer period, or the transfer data rate drops significantly.

(6) Use the dropdown menu next to the Timeout box to select from the available hash types and media hash list types (MHL or ASCMHL).

(7) Use `Add Job` to start transcoding or add a job to the queue.

(8) The `X` icon in the `Active` queue to abort a job or stop an active `Watch Source` job.

(9) Use `Pause Transcoder` to temporarily pause an active job. The encoder will finish the current clip and then wait until you click the button again to `Resume`.

(10) `Clear History` will clear the jobs shown in the job `History` list.

To access the ARRIRAW HDE Transcoder log files, use `File > Show Log Files…` from the menu bar.

3.2 Command Line Version

ARRIRAW HDE Transcoder is also available as a command line tool for macOS, Windows, and Linux.

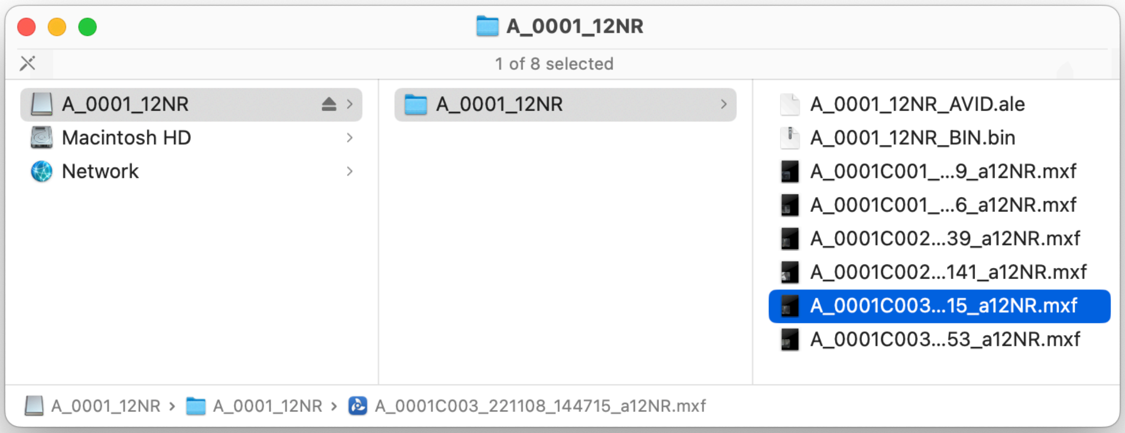
```
arrirawhde [options] -i <input.mxf | input_folder | job.json> -o <output.mxf | output_folder>

options:
 -a activate watch folder
 -c <value>-<type> sets the media hash list and checksum type
     m1 for mhl v1, m2 for ascmhl v2. md5, sha1, sha256, c4, xxh64, xxh3, xxh128
     Example: -c m2-xxh64
 -r <path/file> create a json report with specified file name at the specified path
 -t <value > demux timeout in seconds
 -l legacy output file name. Appends _hde instead of using the hde identifier _h####.mxf.
 -v verbose mode
 -eula print end user license agreement
```

In addition to the functions of the GUI version, the CMD version allows you to specify a single clip or a job file for batch processing. Appending another *-o output.mxf* or *-o output_folder* to the command will write the output to more than one destination.

Command Line Examples

Following are some examples for use of the command line tool. We'll use a Compact Drive named *A_0001_12NR* and two shuttle disks called *TD1* and *TD2*, which are mounted at */Volumes/A_0001_12NR/*, */Volumes/TD1/* and */Volumes/TD2/.*

The camera assigns the reel name to the drive and a folder at the root level, which contains the files.


A_0001_12NR
1 of 8 selected
A_0001_12NR
Macintosh HD
Network
A_0001_12NR
A_0001_12NR_AVID.ale
A_0001_12NR_BIN.bin
A_0001C001_...9_a12NR.mxf
A_0001C001_...6_a12NR.mxf
A_0001C002...39_a12NR.mxf
A_0001C002...141_a12NR.mxf
A_0001C003...15_a12NR.mxf
A_0001C003...53_a12NR.mxf
A_0001_12NR › A_0001_12NR › A_0001C003_221108_144715_a12NR.mxf

Note: In the examples, we use the path notation for macOS or Linux. For Windows, you'd need to specify a path like *X:\A_0001_12NR\A_0001_12NR\A_0001C001_240504_104800_a12NR.mxf*.

Encoding a Single Clip (File)

```
arrirawhde -i /Volumes/A_0001_12NR/A_0001_12NR/A_0001C001_240504_104800_a12NR.mxf -o /Volumes/TD1/
```

This will convert clip *A_0001C001_240504_104800_a12NR.mxf* in folder *A_0001_12NR* on drive *A_0001_12NR* and output an HDE clip named *A_0001C001_240504_104800_h12NR.mxf* on the root level of shuttle disk *TD1*. The ALE file also will be copied to the destination.

Encoding a Reel (Folder)

```
arrirawhde -v -c m2-xxh64 -i /Volumes/A_0001_12NR/ -o /Volumes/TD1/ -o /Volumes/TD2/
```

This will read all contents on drive *A_0001_12NR* and mirror the folder structure on destination drives *TD1* and *TD2*. ARRIRAW files will be encoded. Other files will be copied. Verbose mode will show a line of information for every clip that is processed. The software will also create an *ascmhl* folder including a new chain file (mhl history) and an *ascmhl* including a listing of the files and their *xxHash64* values for all transferred files.

Encoding Data Specified in a Job File

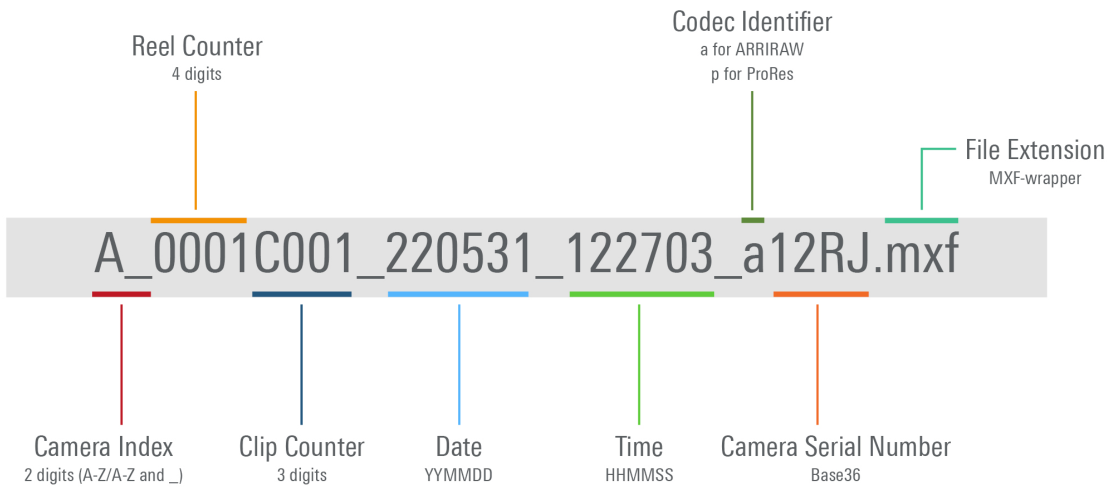
```
arrirawhde -v -i /Users/Demo/day001-transcode.json
```

This will convert the data listed in the job file named *day001-transcode.json*. For this example, we'll assume that all camera reels shot on day 1 have been offloaded to an RAID mounted at */Volumes/OSRAID/* into folders *day001/A-CAM*, *B-CAM* etc. Here's what a job file could look like:

```
{
    "name": "The Example Job – Day1 A-CAM all and B-CAM select",
    "checksum": "m2-xxh64",
    "report": "/Volumes/TD1/reports/day001-report.json",
    "items":
    [
        {
            "source": "/Volumes/OSRAID/day001/A-CAM/",
            "target_dir": "/Volumes/TD1/hde/day001/A-CAM/"
        },
        {
            "source": "/Volumes/OSRAID/day001/B-CAM/B_0001_12SQ/B_0001C003_240408_140120_a12SQ.mxf",
            "target_dir": "/Volumes/TD1/hde/day001/B-CAM/B_0001_12SQ/"
        }
    ]
}
```

4 Workflow Recommendations

MXF files may contain any codec or file format—uncompressed ARRIRAW data, ARRICORE or Apple ProRes data. Without a distinct file extension, you typically would need look into a file with software. To allow a visual differentiation, we use one digit of the file name as a "codec identifier":



H for HDE

When an ARRIRAW clip is converted to HDE, the codec identifier in the file name will change to '**h**'.
The clip metadata fields showing "name" and "file name" are also updated, as are the entries in the .ALE file.

If the original camera file name was changed, the software cannot apply the standard nomenclature. In that case, it will append *_hde* to the file name. You can also force the use of this suffix in the *File > Output File Name* menu. Use of this option, however, is not advisable as most post tools are aligned with the regular codec identifier.

Linking back

The HDE process essentially creates a new camera master. Therefore, it is important that any proxies and other deliverables will link back to the HDE as master and not to the ARRIRAW files. In other words, proxies must be created AFTER the HDE process, so they will link to the correct file (see H for HDE).

Some productions want to keep a copy of the original ARRIRAW data. If you must, then keep a copy of the ARRIRAW until the HDE version passed QC. After that, there really is no need to keep the ARRIRAW data. Especially for the archive, HDE can be accessed the same way but with less tapes/space/cost and with almost double speed due to the reduced file size.

Checksums

The HDE process reads the ARRIRAW image bitstream from an MXF/ARRIRAW file, applies its compression, a merges the encoded image bitstream with the remaining file contents into an MXF/HDE file. Since the ARRIRAW bitstream and the HDE bitstream is different data, the two files cannot be compared by their checksums (and the same applies to the ALE file).

Enabling checksum creation in the ARRIRAW HDE Transcoder means that the software creates the MXF/HDE file and then computes the hash for that file at the desination and stores it in a fresh mhl or ascmhl chain file.

With CODEX Device Manager, the software that is used to create the copy can create a source checksum from the virtual drive and validate that the copy on the backup disk is a match. This has the advantage that it should detect a data corruption immediately after the MXF/HDE file was written to the destination. The downside is that, depending on how the tools handles copy and checksum creation, you may have to go through a second read pass from the virtual drive.

Whichever option is used – each copy made down the line can use the hash values in the mhl/ascmhl to verify that the files were not corrupted or altered.

5 Known Issues

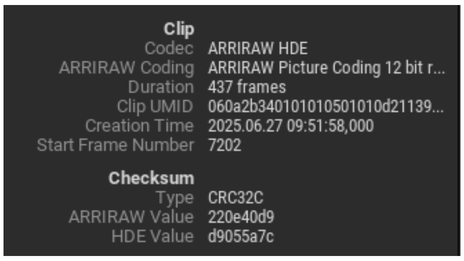
ASCMHL Chain Broken

If the ARRIRAW HDE Transcoder reads from a source that contains an ASC Media Hash List, none of the clip checksums nor the ALE file can be verified. As these files are changed by the transcoder (see section "Checksums" above, it has to create new original checksums.

MXF/HDE files created from the camera original have different checksums

In an MXF file, the Universal Unique Identifier (UUID) or Instance UID is a globally unique value that is generated, using mechanisms such as timestamps, hardware identifiers, or randomness. The UUID ensures that each MXF file or component can be reliably identified across systems, even if filenames or locations change. Generating an MXF/HDE file at different times therefore results in a different UUID, meaning the file also will have a different hash value (checksum).

If you set up more than one destination for the same encode job, the MXF/HDE is created once and written to different targets, resulting in the same UUID and checksum.



Clip	
Codec	ARRIRAW HDE
ARRIRAW Coding	ARRIRAW Picture Coding 12 bit r...
Duration	437 frames
Clip UMID	060a2b340101010501010d21139...
Creation Time	2025.06.27 09:51:58,000
Start Frame Number	7202
Checksum	
Type	CRC32C
ARRIRAW Value	220e40d9
HDE Value	d9055a7c

HDE CRC image checksum

MXF/HDE files include a secondary image checksum to allow for a validation of the compressed image bitstream without the need to invert the lossless encoding. The definition will be added to SMPTE RDD 55.

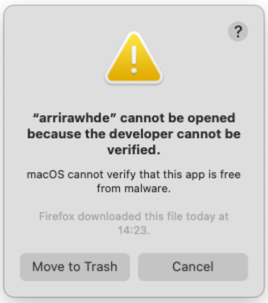
Legacy _hde filename is not linked in ALE File

The ALE file is always patched to reference clip names and file names using the _h codec identifier, even if the application is set to use the legacy _hde suffix. That's why we cannot recommend this option.

macOS and Installation of the GUI Version

If the installation of the software fails on a Mac, please check that Installer.app has Full Disk Access in System Prefs > Security & Privacy and launch the installer .PKG file from within the applications folder.

macOS and First Run of the Command Line Version



"arrirawhde" cannot be opened because the developer cannot be verified.

macOS cannot verify that this app is free from malware.

Firefox downloaded this file today at 14:23.

The first time you try to run the command line version on macOS, you will get a popup that the software cannot be opened.

Click 'Cancel', then open the System Preferences, go to 'Privacy and Security', scroll down to 'Security' where, you should that "arrirawhde" was blocked from use. Click 'Allow Anyway'. Next time you start the software, you have to OK that you really want to open it.

macOS and System Sleep

The encoding might pause if your Mac enters sleep mode. This may happen if the screen is locked or left unattended for an extended period. There are apps that allow you to lock the screen and keep the computer awake at all time, but there's nothing we can do about the system settings.

macOS and Spaces

When using 'Spaces' on the Mac, moving away from the Space running the arrirawhde process will push the process over to the slower efficiency cores. To avoid this behaviour, don't switch Spaces or right click on the ARRIRAW HDE Transcoder dock icon and assign the app to all Spaces.

Processing order

Especially for jobs using the 'Watch Source' option, the files may not be processed in the expected order.

Temporary Files and Watch Source

The ARRIRAW HDE Transcoder lacks the function to ignore and exclude certain files/filetypes. This is an issue when you run a 'Watch Source' job and the tool that drops files into the source folder uses temporary files (e.g. .pfncopy as used by Pomfort). If the transcoder happens to come across such a temporary file, it's going to detect it as foreign file and copy it to the destination. While it's doing that, the temporary file may not be changed by the copy tool. We're working on resolving this in an upcoming version.

6 Change Notes

Version 1.6.4

- Metadata that is not included in an MXF/ARRIRAW that was shot with an SUP before 5.0.0 no longer triggers a warning.
- Bug fixed: application crash with ALEXA 35 SUP 1.2.0 footage

Version 1.6.3

- Bug fixed: wrong essence coding UL uses for the "RAW 12bit Reversed" layout.
- Bug fixed: the "secondary checksum" value inserted outside of the Frame Capture Set.
- Added support for new RDD-54/55 subdescriptors and PTP timestamp metadata.
- ASCMHL and MHL output now show correct tool verison.

Version 1.6.2

- Improved macOS GUI exception handling.
- Bug fixed: console app does require -o argument although destination is specified into the json input.
- Bug fixed: a hash specified in json input for console app is accounted only if some hash mode was specified in command line.
- Bug fixed: directory transcoding mode specified in json input for console app does not work.

Version 1.6.1

- Changed the mhl file name to <numbering>_<source-folder-name>_<YYMMDD_hhmmss>.mhl
- Legacy support: Reduced the hash type selection for MHL v1 to prevent verification errors resulting from unsupported hash types.
- Legacy support: Renamed the MHL hash type ID for “xxh64” to “xxhash64be” so MHL v1 tools will no longer fail to verify.
- Bugs fixed: "illegal instruction" and “segmentation fault” under Linux.
- Improvement: The CentOs Linux version now should run on RHEL and Rocky Linux as well.
- Improvement: The job table now shows an info for multi-destination jobs.
- Other minor fixes and improvements.

Version 1.6

- Bug fixed: wrong bytes 13,14,15 calculation for material and source UMIDs.
- Improved GUI layout under MacOs.
- Improved error handling if a destination does not have enough disk space.
- Added a warning if a multi-destination setup points to the same location more than once.

Version 1.5

- Multiple destinations feature added.
- Bug fixed: the warning dialog does not keep a resized state.
- Disable a filename suffix changing, ALE patching, clip metadta patching.
- Bug fixed: truncated output for SHA1 hash.
- Bug fixed: incorrect C4 calculation for big files.
- Skip files with name started with '.'
- C4 hashing now uses always for mhl chain items.

Version 1.4

- Output files now include an HDE CRC image checksum (to be added to the RDD)
- The application now copies and hashes all files from the source path.
- The _h encoding identifier is now carried forward into the clip name metadata and the ALE file.
- New hashers added: SHA1, SHA256, C4, XXH3, XXH128.
- Added maintenance for ascmhl_chain.xml file.
- Added ascmhl file sequencing.
- Handling for Ctrl-C and other signals has been added to console application.
- json report for command line watchfolder mode now updated after each file.
- Work in progress: Validation, if the source path contains an ascmhl hashlist.
- The behavior of the processed clip counter has been changed.

Version 1.3 beta

- The "clear log" button caption changed to "clear history".
- Keep history unless it is cleared by the user implemented.
- Spaces into the ìhashlistî tag of ascmhl have removed.
- Menu: File > Show Log File(s) replacement.
- Output file name correction to use _h encoding identifier by default instead of _hde suffix.
- Imprint tool's build number into the mhl/ascmhl.
- Mhl v1 support added.
- Checksums for .ale and .bin files calculated now.
- Preserve Clip (UU)ID in Material Package UID.
- Bug fixed: "ascmhl conformity - leading zeros of xxhash64 checksums missing".

7 References

Media Hash List: https://mediahashlist.org/

ASC Media Hash List: https://theasc.com/asc/asc-media-hash-list and https://github.com/ascmitc/mhl

ARRIRAW demosaicing, camera system metadata, the representation in the MXF container and High Density Image Encoding are available publicly as SMPTE Registered Disclosure Documents.

- RDD 31, Deferred Demosaicing of an ARRIRAW Image File to a Wide-Gamut Logarithmic Encoding
- RDD 54, Material Exchange Format — Mapping ARRIRAW Bitstreams into the MXF Generic Container
- RDD 55, Material Exchange Format, Carriage of ARRI Camera System Metadata
- RDD 51, High Density Image Encoding for ARRIRAW Files

To access SMPTE RDDs, please go here https://www.smpte.org/standards/document-index/rdd

For questions and feedback on the ARRIRAW HDE Transcoder, please send us a message to digitalworkflow@arri.de!