

MFormats SDK

Release notes

2.12.0.14909 VERSION
August 2026

This document lists MFormats SDK changes. Each release-note item includes a concise description of the affected workflow, the observable symptom, and the result.

23 ADDED	54 FIXED	3 UPDATED	0 CHANGED
--------------------	--------------------	---------------------	---------------------

DOCUMENT STRUCTURE

What is inside

Common Changes - 20 release-note items
File Recording and Network Streaming - 18 release-note items
Input/Output Devices - 13 release-note items
Decoding and Seeking - 7 release-note items
Plugins - 19 release-note items
Samples and Documentation - 3 release-note items

RELEASE OVERVIEW

MFormats SDK - Version 2.12.0.14909

Added capabilities are listed first in each section. Select any item to jump to its detailed explanation.

Common Changes

ADDED	New GPU shader transitions based on gl-transitions.
ADDED	LTC reading from a selected embedded audio channel.
ADDED	xMedia Converter statistics.
ADDED	Outside, styled borders for MFPrint text overlays.
ADDED	Runtime enumeration of GPU transitions and their default parameters.
ADDED	Configurable aspect ratio for Unreal Engine xMedia output.
FIXED	Crash when 24-bit audio conversion is used together with audio channel remapping.
FIXED	MFCConvert could duplicate audio samples when switching between video sources.
FIXED	Duplicated frames in rotate + maintain_ar + gpu_pipeline scenarios.
FIXED	object::vfilter and object::afilter processing in gpu_pipeline mode.
FIXED	MFPrint outline rendering on small ARGB frames.
FIXED	Audio distortion during normalization for specific source files.
FIXED	Frame cloning changing nRowBytes in gpu_pipeline mode.
FIXED	external_process behavior consistency between code and registry configuration.
FIXED	pixelize video filter processing for ARGB formats.
FIXED	MFReaderSink audio normalization for mp:// sources.
FIXED	Reusing writer format configuration after AttributesRemoveAll.
FIXED	Delayed MFXMediaConverter video-filter initialization for live sources.
FIXED	Unreal Engine xMedia video-callback pacing.
FIXED	Vertical orientation during ARGB32 frame conversion.

File Recording and Network Streaming

ADDED	DVB-compatible MPEG-TS streaming over RTP.
ADDED	WebVTT subtitle support for HLS streams.
ADDED	DVB subtitle generation and insertion into DVB streams.
ADDED	Teletext support for EBU Teletext / ETSI EN 300 706 workflows.
ADDED	Selective seamless output switching for multi-output writers.
ADDED	Subtitle encoder and rasterizer formatting options with automatic canvas sizing.
ADDED	Multilingual DVB Teletext encoding, including Cyrillic character sets.
ADDED	SCTE-35 detection from HLS playlist tags.
FIXED	MFWriter indexed audio PID settings for additional audio streams.
FIXED	Duplicated AC-3 descriptor in DVB PMT output.
FIXED	MXF pre-charge metadata loss during transwrapping.
FIXED	SRT streams with subtitle::codec=packets opening in VLC.
FIXED	Audio packet passthrough when source stream indices contain gaps.
FIXED	Timecode metadata in MP4 recordings produced by the NVIDIA encoder.
FIXED	setsar/setdar filter metadata in MOV and ProRes output.
FIXED	DVB subtitle packet timing after skipping initial source frames.
FIXED	DVB receiver interruption at overlapping SRT cue times.
FIXED	messageSize in single-operation SCTE-104 ANC packets.

Input/Output Devices

ADDED	ASIO device support as an LTC synchronization source.
ADDED	AJA LTC timecode input source option.
FIXED	DeckLink key/fill processing at frame rates above 30 fps.
FIXED	DeckLink custom ANC packet reception.
FIXED	Bluefish Kronos K8 audio loss in key/fill output modes.
FIXED	Long shutdown time for Blackmagic/DeckLink renderers.
FIXED	SDI auto-detect reinitialization when a 59.94p signal appears after no-signal startup.

FIXED	BMD no-signal frame timing for live capture pipelines.
FIXED	MFLive internal audio device switching without full reinitialization.
FIXED	NDI color metadata propagation when the source changes.
FIXED	HDMI DeckLink Mini Recorder 4K initialization crash in gpu_pipeline mode.
FIXED	MFLive event channel IDs for hardware inputs.
FIXED	SCTE-104 detection when a Blackmagic VANC line contains multiple ANC packets.

Decoding and Seeking

ADDED	video.auto_rotation property for automatic rotation-flag handling in MFReader.
ADDED	MXF SMPTE RP 188 and SMPTE 385M timecode support.
FIXED	MFReader reconnection to UDP streams in specific end-of-stream recovery cases.
FIXED	start_timecode metadata reading for GXF files.
FIXED	Frame-rate flags returned by timecode conversion.
FIXED	Intermittent or missing audio while receiving live UDP streams.
FIXED	Audio codec metadata when the selected decoder has a different name.

Plugins

ADDED	SMPTE 436M CC, V-Chip, and AFD metadata support.
ADDED	cc.display_order option to choose CEA-608 processing order from packet-level or decoded-frame data.
ADDED	V-Chip/XDS insertion and parsing support.
ADDED	HTML5 MLCG editor.
FIXED	CG overlay color rendering when overlay.thread=0.
FIXED	CG overlay rendering for interlaced formats when overlay.thread=0.
FIXED	Extra spacing between adjacent CG text runs in tickers.
FIXED	CEF opening an external browser when HTML overlay is used from two applications at the same time.
FIXED	CG RSS ticker text width and spacing mismatch.

FIXED	Display of closed captions written to MXF files.
FIXED	Generated CEA-608 closed captions in files for VLC playback.
FIXED	Blurred CG text when <code>decoder.convert_planar=true</code> .
FIXED	MCCDisplay CEA-608 character ordering for specific files.
FIXED	CEF HTML plugin and WebCapture page loading on first start after OS boot.
FIXED	CEF audio mute and preview-control handling.
FIXED	CEF popup handling that could create an extra preview/window for new tabs.
UPDATED	CEF to version 147.0.11.
FIXED	Smooth scheduled appearance for CG items shown with <code>cg-props::show</code> .
FIXED	CG hang when hiding a ticker containing a non-RSS web page.

Samples and Documentation

ADDED	MLCG REST control sample for MFormats workflows.
UPDATED	MFormats MLShader samples with the new shader-library project and shader catalog.
UPDATED	Subtitle workflows split into dedicated Conversion and Generation samples.

TECHNICAL DETAILS

MFormats SDK release details

Descriptions below summarize the affected SDK workflow and result. Internal tracker references and raw issue text are intentionally omitted.

Common Changes

ADDED**New GPU shader transitions based on gl-transitions.**

APPLIES TO MFormats

Applications using GPU transitions had a limited built-in transition set and could not directly use the newer gl-transitions style effects from the SDK package.

RESULT

The SDK includes the new GPU shader transition set.

EXAMPLE - MFormats GPU transition call

```
factory.PropsSet("gpu_pipeline", "true");
double position = 0.5;
frameTo.MFTransition(
    frameFrom,
    out MFFrame transitionFrame,
    position,
    transitionName,
    "",
    "transition_instance");
```

ADDED**LTC reading from a selected embedded audio channel.**

APPLIES TO MFormats

Applications receiving LTC embedded in a multi-channel audio signal needed to choose the exact audio channel used as the LTC source. Without that selection, synchronization could read the wrong channel or fail to lock.

RESULT

LTC can be read from a selected embedded audio channel.

EXAMPLE - Select an embedded audio channel for LTC

```
live.PropsSet("ltc.enabled", "true");
live.PropsSet("ltc_audio_device", "<input device>");
live.PropsSet("ltc_audio_channel", "1");
live.ObjectStart(true);
// Read timecode after the LTC source locks.
```

ADDED**xMedia Converter statistics.**

APPLIES TO MFormats

Applications using xMedia Converter could not read the conversion statistics needed for monitoring, diagnostics, or UI status.

RESULT

xMedia Converter now exposes statistics through the SDK workflow.

ADDED**Outside, styled borders for MFPrint text overlays.**

APPLIES TO MFormats

MFPrint accepted inside borders but could not draw a negative-width border outside the text rectangle or configure its line style independently.

RESULT MFPrint now supports negative outside borders plus border-width, border-style, border-color, and rounded background corners.

EXAMPLE - Draw an MFPrint border outside the text rectangle

```
string textProps =  
    "face='Consolas' color='white' bg-color='black(255)' " +  
    "border-width='-8' border-style='dashed' border-color='red(255)' " +  
    "bg-corners='6'";  
frame.MFPrint("Status", 1.0, ref textRect,  
    eMFTextFlags.eMFT_WordBreaks, textProps);
```

ADDED

Runtime enumeration of GPU transitions and their default parameters.

APPLIES TO MFormats

Applications had to hardcode transition names and parameters to build transition pickers, which became error-prone as the GPU transition library expanded.

RESULT The SDK factory now exposes TransitionGetCount and TransitionGetByIndex.

EXAMPLE - Enumerate MFormats GPU transitions

```
MFFactoryClass factory = new MFFactoryClass();  
factory.TransitionGetCount(out int count);  
for (int index = 0; index < count; index++)  
{  
    factory.TransitionGetByIndex(index, out string name, out string defaultParams);  
    AddTransitionToUi(name, defaultParams);  
}
```

ADDED

Configurable aspect ratio for Unreal Engine xMedia output.

APPLIES TO MFormats

xMedia output created by the Unreal Engine integration always advertised a 4:3 aspect ratio, even when the rendered content used a different display shape.

RESULT Unreal Engine xMedia output can now specify its aspect ratio.

FIXED

Crash when 24-bit audio conversion is used together with audio channel remapping.

APPLIES TO MFormats

24-bit audio conversion combined with explicit audio channel remapping could access the audio buffer incorrectly and crash.

RESULT The 24-bit audio conversion path now handles channel mapping safely.

FIXED

MFFormat could duplicate audio samples when switching between video sources.

APPLIES TO MFormats

MFFormat used only for video format conversion could occasionally change the audio sample count, for example doubling 1920 samples to 3840 while the audio format itself stayed unchanged. This could distort audio after source switching.

RESULT MFFormat preserves the audio sample count when only video format conversion is requested.

EXAMPLE - Check MFFormat preserves audio sample count

```
frame.MFAVPropsGet(out M_AV_PROPS avProps, out int samplesBefore);
avProps.vidProps.fccType = eMFCC.eMFCC_YV12;
frame.MFConvert(avProps, out MFFrame convertedFrame, out int frameRest, "", "");
convertedFrame.MFAVPropsGet(out M_AV_PROPS convertedProps, out int samplesAfter);
if (samplesAfter != samplesBefore)
    throw new InvalidOperationException("MFConvert changed audio sample count.");
```

FIXED**Duplicated frames in rotate + maintain_ar + gpu_pipeline scenarios.**

APPLIES TO MFormats

Combining rotation, maintain_ar, and gpu_pipeline could produce duplicated output frames in this processing path.

RESULT

The rotate + maintain_ar GPU path now preserves correct frame delivery without duplicates.

FIXED**object::vfilter and object::afilter processing in gpu_pipeline mode.**

APPLIES TO MFormats

Object-level video and audio filters could be skipped or processed incorrectly when gpu_pipeline mode was enabled.

RESULT

object::vfilter and object::afilter are now applied correctly in gpu_pipeline mode.

FIXED**MFPrint outline rendering on small ARGB frames.**

APPLIES TO MFormats

MFPrint text outlines could render incorrectly on small ARGB frames.

RESULT

MFPrint outline rendering is now correct on small ARGB frames.

FIXED**Audio distortion during normalization for specific source files.**

APPLIES TO MFormats

Audio gain normalization could create a loud distorted transient for some sources, especially after an initial silent part of the file.

RESULT

Audio normalization handles this source pattern without producing the distorted transient.

FIXED**Frame cloning changing nRowBytes in gpu_pipeline mode.**

APPLIES TO MFormats

Cloning a frame in gpu_pipeline mode could change nRowBytes and expose an unexpected row stride to downstream processing.

RESULT

Frame cloning now preserves the expected nRowBytes value in gpu_pipeline mode.

FIXED**external_process behavior consistency between code and registry configuration.**

APPLIES TO MFormats

external_process could behave differently depending on whether it was configured from code or through default/registry settings.

RESULT external_process behavior is now consistent across configuration paths.

FIXED**pixelize video filter processing for ARGB formats.**

APPLIES TO MFormats

The pixelize video filter could produce incorrect output when processing ARGB frames.

RESULT The pixelize filter now processes ARGB formats correctly.

FIXED**MReaderSink audio normalization for mp:// sources.**

APPLIES TO MFormats

Audio normalization enabled on MReaderSink was skipped when its input was an mp:// MReaderSink source, so measured and audible levels differed from direct-file playback.

RESULT MReaderSink now applies the normal audio-processing path, including LUFS and gain normalization.

FIXED**Reusing writer format configuration after AttributesRemoveAll.**

APPLIES TO MFormats

After removing every attribute from a writer format configuration, selecting the same format again could fail until the writer object was recreated.

RESULT AttributesRemoveAll restores the writer configuration defaults, allowing repeated ConfigSet calls without recreating the writer.

FIXED**Delayed MFXMediaConverter video-filter initialization for live sources.**

APPLIES TO MFormats

A video filter configured on MFXMediaConverter could take about ten seconds to affect webcam or other live-source frames even though reader sources were filtered immediately.

RESULT Live-source video filters are initialized before frame processing begins.

FIXED**Unreal Engine xMedia video-callback pacing.**

APPLIES TO MFormats

Video callbacks delivered to Unreal Engine could arrive in bursts with long and short intervals even though the average frame rate remained correct, forcing applications to add extra buffering.

RESULT xMedia callback delivery is paced more consistently for Unreal Engine sources.

FIXED**Vertical orientation during ARGB32 frame conversion.**

APPLIES TO MFormats

Explicit conversion to ARGB32 could vertically flip the decoded picture because the requested RGB row orientation was not preserved.

RESULT ARGB32 conversion and cloning now preserve the requested top-down or bottom-up orientation.

File Recording and Network Streaming

ADDED

DVB-compatible MPEG-TS streaming over RTP.

APPLIES TO MFormats

Applications that needed DVB-compatible MPEG-TS delivery over RTP had no direct SDK-supported output path for that transport combination.

RESULT DVB-compatible MPEG-TS streaming over RTP is now supported.

ADDED

WebVTT subtitle support for HLS streams.

APPLIES TO MFormats

Applications that generate HLS output can now include a WebVTT subtitle track. This is useful when an external subtitle file, such as ASS or SRT, must be passed through the reader and written as .vtt files next to the HLS playlist.

RESULT HLS output with WebVTT subtitles is supported.

EXAMPLE - MFormats HLS/WebVTT frame-loop setup

```
string readerConfig =
    "external_process=false subtitle_track=0 " +
    "experimental.out_subtitle_packets=2 " +
    $"external_subtitle='{subtitlePath}';";
string writerConfig =
    "format=hls' mux_buffers=0 hls_time=2 hls_list_size=0 " +
    "hls_playlist_type='vod' subtitle::codec='webvtt' " +
    "subtitle::metadata::language='eng' video::codec='libopenh264' " +
    "video::maxrate='5M' audio::codec='aac';";
reader.ReaderOpen(sourcePath, readerConfig);
writer.WriterSet(outputPlaylist, 0, writerConfig);
reader.SourceFrameGet(-1, out MFFrame frame, "");
writer.ReceiverFramePut(frame, -1, "");
Marshal.ReleaseComObject(frame);
// After writing, read generated *.vtt files and check for expected cue text.
```

ADDED

DVB subtitle generation and insertion into DVB streams.

APPLIES TO MFormats

Developers implementing DVB subtitle insertion in MFormats did not have a complete SDK sample showing the workflow.

RESULT A MFormats DVB subtitle insertion sample was added.

EXAMPLE - External text subtitles to DVB output

```
string readerConfig =
    "external_process=false subtitle_track=0 " +
    "experimental.out_subtitle_packets=2 " +
    $"external_subtitle='{subtitlePath}';";
string writerConfig =
    "format=dvb' protocol='udp://' video::codec='n264' audio::codec='aac' " +
    "subtitle::codec='dvbtxt' subtitle::stream_id=400 " +
    "subtitle::metadata::language='eng' " +
    "subtitle.1::codec='dvbsub' subtitle.1::stream_id=401 " +
    "subtitle.1::metadata::language='eng';";
```

ADDED**Teletext support for EBU Teletext / ETSI EN 300 706 workflows.**

APPLIES TO MFormats

Applications that need EBU Teletext / ETSI EN 300 706 subtitle or data workflows previously had no built-in SDK support for that Teletext output path.

RESULT

Teletext support was added for EBU Teletext / ETSI EN 300 706 workflows.

ADDED**Selective seamless output switching for multi-output writers.**

APPLIES TO MFormats

Changing a writer destination to segment a local recording could also reconnect a parallel network output in a two-output writer configuration.

RESULT

The `seamless_switches` option selects only the output indices that should move to the new destination, allowing other streams to continue uninterrupted.

EXAMPLE - Switch only the local MFormats recording output

```
string writerConfig =
    "format='mp4' video::codec='libopenh264' audio::codec='aac' " +
    $"stream.1::format='mpegts' stream.1::path='{udpUrl}'";
writer.WriterSet(firstSegment, 1, writerConfig);
// Keep stream.1 running and switch only primary output index 0.
writer.WriterSet(secondSegment, 1,
    "seamless_switch='true' seamless_switches='0'");
```

ADDED**Subtitle encoder and rasterizer formatting options with automatic canvas sizing.**

APPLIES TO MFormats

Generated DVB subtitles and Teletext used a fixed internal option path, making it difficult to set fonts, colors, backgrounds, outlines, or a canvas matched to the output video.

RESULT

Subtitle writer options now reach the encoder/rasterizer, and video width and height are supplied automatically as the subtitle canvas.

EXAMPLE - Configure generated DVB subtitle appearance

```
string writerConfig =
    "format='dvb' protocol='udp://' video::codec='libopenh264' audio::codec='aac' " +
    "subtitle::codec='dvbsub' subtitle::metadata::language='eng' " +
    "subtitle::font='Arial' subtitle::font_size=28 subtitle::font_color='FFFFFF' " +
    "subtitle::background=true subtitle::background_color='000000' " +
    "subtitle::background_alpha=128 subtitle::outline=true " +
    "subtitle::outline_color='000000' subtitle::outline_thickness=2";
```

ADDED**Multilingual DVB Teletext encoding, including Cyrillic character sets.**

APPLIES TO MFormats

DVB Teletext output could replace or corrupt Cyrillic text and did not provide a way to select primary and secondary Teletext G0 character sets for mixed-language subtitle tracks.

RESULT

DVB Teletext encoding supports language-aware and explicitly configured Latin/Cyrillic character-set combinations.

EXAMPLE - Select DVB Teletext character sets

```
string writerConfig =  
    "format='mpegts' subtitle::codec='dvbtxt' " +  
    "subtitle::metadata::language='mul' " +  
    "subtitle::teletext_primary_charset='latin' " +  
    "subtitle::teletext_secondary_charset='cyrillic_russian_bulgarian' " +  
    "video::codec='libopenh264' audio::codec='aac';
```

ADDED**SCTE-35 detection from HLS playlist tags.**

APPLIES TO MFormats

HLS SCTE-35 markers carried in common playlist tags were not surfaced to the application, so it could not react to splice points described only by the m3u8 playlist.

RESULT

The SDK detects CUE, SCTE35, SPLICEPOINT, OATCLS, and SCTE35 DATERANGE forms and attaches the marker to the first video frame of the related segment.

EXAMPLE - Read an HLS SCTE-35 marker from a decoded frame

```
reader.ReaderOpen(hlsUrl, "");  
reader.SourceFrameGet(-1, out MFFrame frame, "");  
try  
{  
    frame.MFStrGet("SCTE-35", out string marker);  
    if (!string.IsNullOrEmpty(marker))  
        HandleScte35(marker);  
}  
finally  
{  
    Marshal.ReleaseComObject(frame);  
}
```

FIXED**MFWriter indexed audio PID settings for additional audio streams.**

APPLIES TO MFormats

MPEG-TS outputs with split audio channels could ignore audio.1::pid, audio.2::pid, and later indexed PID settings. Only audio::pid was used, while other audio streams received automatically assigned sequential PIDs.

RESULT

Indexed audio PID settings are applied to additional audio streams.

EXAMPLE - Explicit audio PIDs for split audio streams

```
string writerConfig =  
    "format='mpegts' protocol='srt://' split_channels='1' " +  
    "audio::pid=4112 audio.1::pid=4124 " +  
    "audio.2::pid=4128 audio.3::pid=4160";
```

FIXED**Duplicated AC-3 descriptor in DVB PMT output.**

APPLIES TO MFormats

DVB PMT generation for AC-3 audio could write two AC-3 descriptors for the same elementary stream. Some analyzers or receivers could then read duplicated or conflicting AC-3 metadata.

RESULT

DVB PMT output writes a single correct AC-3 descriptor for the stream.

FIXED**MXF pre-charge metadata loss during transwrapping.**

APPLIES TO MFormats

MXF transwrapping could lose pre-charge metadata, so the output file did not preserve the timing/metadata structure of the source around the beginning of the essence.

RESULT MXF transwrapping preserves pre-charge metadata.

EXAMPLE - MFormats MXF packet transwrap preserving metadata

```
string writerConfig =  
    "format='mxf' embed_cc='true' video::codec='packets' audio::codec='packets'";  
reader.ReaderOpen(sourceMxf, "");  
writer.WriterSet(outputMxf, 0, writerConfig);
```

FIXED

SRT streams with subtitle::codec=packets opening in VLC.

APPLIES TO MFormats

Applications that pass subtitle streams as packets could start the writer before the first subtitle packet arrived. In that case the stream could miss subtitle initialization metadata and fail to open correctly in receivers such as VLC.

RESULT Packet subtitle streams can be initialized before the first subtitle data packet arrives.

EXAMPLE - Packet subtitle stream initialization

```
writer.PropsSet("encoder::subtitle.init_without_data", "true");  
string writerConfig =  
    "format='mpegs' protocol='srt://' video::codec='n264' audio::codec='aac' " +  
    "subtitle::codec='packets' subtitle::metadata::language='eng'";
```

FIXED

Audio packet passthrough when source stream indices contain gaps.

APPLIES TO MFormats

Using audio::codec='packets' could omit later audio tracks when the source contained a missing stream index, changing the expected track order or channel layout in DVB output.

RESULT Available packet audio streams are now written consecutively while preserving their source order and channel count.

FIXED

Timecode metadata in MP4 recordings produced by the NVIDIA encoder.

APPLIES TO MFormats

MP4 recordings created with the NVIDIA encoder and start_timecode could omit timecode information for GOP frames, making dropped-frame analysis unreliable.

RESULT NVIDIA-encoded MP4 output now preserves timecode metadata across GOP frames.

FIXED

setsar/setdar filter metadata in MOV and ProRes output.

APPLIES TO MFormats

setsar/setdar filters changed the processed picture but MOV/ProRes files could still be written without the resulting sample/display aspect-ratio metadata.

RESULT Calculated SAR is now propagated to the codec parameters used by the MOV muxer.

FIXED

DVB subtitle packet timing after skipping initial source frames.

APPLIES TO MFormats

When a transwrap workflow skipped several seconds before starting its output, video began at the new position but DVB subtitle packets retained original timestamps and appeared late.

RESULT Attached subtitle packet timestamps now follow the adjusted frame timeline.

FIXED**DVB receiver interruption at overlapping SRT cue times.**

APPLIES TO MFormats

Encoding SRT subtitles as DVB subtitles could interrupt receiver playback at a point where two source cues overlapped, although the writer still reported a running state.

RESULT The DVB subtitle encoder now handles overlapping source cue intervals without stopping the transport stream.

FIXED**messageSize in single-operation SCTE-104 ANC packets.**

APPLIES TO MFormats

Single-operation SCTE-104 packets counted the payload descriptor byte in messageSize, producing a value one byte larger than required by SCTE-104.

RESULT messageSize now describes only the SCTE-104 message after the payload descriptor.

Input/Output Devices

ADDED**ASIO device support as an LTC synchronization source.**

APPLIES TO MFormats

Applications that synchronize by LTC could not use an ASIO audio device as the LTC source. This mattered when the timecode reference was available only through an audio interface instead of a video device input.

RESULT ASIO devices can now be selected as LTC synchronization sources.

ADDED**AJA LTC timecode input source option.**

APPLIES TO MFormats

Applications using AJA hardware could not select the board LTC input as the timecode source directly from the SDK workflow.

RESULT AJA LTC input can now be selected as a timecode source.

FIXED**DeckLink key/fill processing at frame rates above 30 fps.**

APPLIES TO MFormats

DeckLink key/fill output could fail or drop performance for frame rates above 30 fps on affected key/fill hardware workflows.

RESULT DeckLink key/fill processing supports the affected high-frame-rate modes.

EXAMPLE - MFormats high-frame-rate key/fill output loop

```
live.DeviceSet("video", deckLinkInput, "format='1080p5994'");
renderer.DeviceSet("renderer", deckLinkOutput, "keying=internal");
while (running)
{
    live.SourceFrameGet(-1, out MFFrame frame, "");
    renderer.ReceiverFramePut(frame, -1, "");
    Marshal.ReleaseComObject(frame);
}
```

FIXED**DeckLink custom ANC packet reception.**

APPLIES TO MFormats

Custom ANC packets could be missed or received incorrectly in DeckLink capture workflows.

RESULT

DeckLink custom ANC packet reception is now handled correctly.

FIXED**Bluefish Kronos K8 audio loss in key/fill output modes.**

APPLIES TO MFormats

Bluefish key/fill output could lose audio while preparing fill/key video frames.

RESULT

Audio is preserved during Bluefish key/fill output processing.

FIXED**Long shutdown time for Blackmagic/DeckLink renderers.**

APPLIES TO MFormats

Blackmagic/DeckLink renderers could take an excessive amount of time to shut down.

RESULT

DeckLink renderer shutdown now completes without the previous long delay.

FIXED**SDI auto-detect reinitialization when a 59.94p signal appears after no-signal startup.**

APPLIES TO MFormats

If an SDI input was initialized with no signal and a 59.94p signal appeared later, the preview could show split, stacked, or otherwise incorrectly reassembled video until the device was reinitialized.

RESULT

SDI auto-detect reinitializes correctly when a 59.94p signal appears after no-signal startup.

FIXED**BMD no-signal frame timing for live capture pipelines.**

APPLIES TO MFormats

Live capture pipelines using Blackmagic/DeckLink inputs could miss proper frame timing while one input was in no-signal state. This could affect downstream live processing and renderer synchronization.

RESULT

No-signal frames now carry timing suitable for live capture pipelines.

FIXED**MFLive internal audio device switching without full reinitialization.**

APPLIES TO MFormats

Switching an MFLive internal audio device from From Video to No Audio and back could fail to restore audio until the video device was fully reinitialized.

RESULT MFLive can restore the internal audio source without full device reinitialization.

FIXED NDI color metadata propagation when the source changes.

APPLIES TO MFormats

NDI workflows could lose or propagate incorrect color metadata when the source changed.

RESULT NDI color metadata is now propagated correctly through source changes.

FIXED HDMI DeckLink Mini Recorder 4K initialization crash in gpu_pipeline mode.

APPLIES TO MFormats

Initializing an HDMI DeckLink Mini Recorder 4K input with gpu_pipeline enabled could crash the application.

RESULT DeckLink Mini Recorder 4K initialization is now stable in gpu_pipeline mode.

FIXED MFLive event channel IDs for hardware inputs.

APPLIES TO MFormats

MFLive OnEvent callbacks identified the channel as MFLive instead of the active device name, even though frame callbacks carried the real SDI device name.

RESULT Live format, signal, and SCTE-104 events now use the active device name as bsChannelID.

FIXED SCTE-104 detection when a Blackmagic VANC line contains multiple ANC packets.

APPLIES TO MFormats

Blackmagic input exposed no SCTE-104 frame property when the same VANC line also contained another ANC packet such as OP-47 subtitles.

RESULT The Blackmagic input path examines every ANC packet and recognizes SCTE-104 by DID/SDID regardless of neighboring packets.

Decoding and Seeking

ADDED video.auto_rotation property for automatic rotation-flag handling in MFReader.

APPLIES TO MFormats

MFReader required application-side logic to apply source rotation flags automatically during decoding.

RESULT video.auto_rotation can now apply rotation flags automatically in MFReader.

ADDED**MXF SMPTE RP 188 and SMPTE 385M timecode support.**

APPLIES TO MFormats

MXF files carrying SMPTE RP 188 or SMPTE 385M timecode metadata could expose incomplete timecode information to SDK applications.

RESULT The SDK reads SMPTE RP 188 and SMPTE 385M timecode from MXF files.**EXAMPLE - MFormats reading MXF timecode sources**

```
reader.ReaderOpen(sourceMxf, "mxf.force_ffmpeg='true'");
reader.SourceFrameGet(-1, out MFFrame frame, "");
frame.MFTimeGet(out M_TIME time);
// Inspect time.tcFrame and the available timecode-source metadata.
```

FIXED**MFReader reconnection to UDP streams in specific end-of-stream recovery cases.**

APPLIES TO MFormats

MFReader could fail to recover after a UDP stream was stopped and restarted several times. A typical symptom was stopped video with continuously growing audio buffers.

RESULT MFReader reconnection is more reliable for repeated UDP end-of-stream recovery cases.**FIXED****start_timecode metadata reading for GXF files.**

APPLIES TO MFormats

Files or device inputs carrying timecode metadata could expose missing, discontinuous, or unavailable timecode in the SDK workflow.

RESULT Timecode extraction or propagation was corrected for this case.**FIXED****Frame-rate flags returned by timecode conversion.**

APPLIES TO MFormats

Converting a high-frame-rate timecode to a standard rate could retain progressive-rate flags from the source and report the wrong target rate flag.

RESULT Timecode conversion now clears incompatible source flags and reports the target frame rate correctly.**FIXED****Intermittent or missing audio while receiving live UDP streams.**

APPLIES TO MFormats

MFReader could decode video from a live UDP stream while audio remained absent for long periods or appeared only in short bursts when audio packets arrived unevenly and lagged behind video.

RESULT Network audio buffering and first-packet detection now tolerate irregular packet gaps without dropping the audio timeline.**FIXED****Audio codec metadata when the selected decoder has a different name.**

APPLIES TO MFormats

MFReader could report the selected FFmpeg decoder name instead of the actual stream codec, for example identifying an MP2 transport-stream track as MP3.

RESULT info codec fields now come from the input stream's codec parameters.

Plugins

ADDED

SMPTE 436M CC, V-Chip, and AFD metadata support.

APPLIES TO MFormats

SMPTE 436M workflows needed SDK support for closed-caption, V-Chip, and AFD metadata carried with the media.

RESULT SMPTE 436M CC, V-Chip, and AFD metadata are now supported.

ADDED

cc.display_order option to choose CEA-608 processing order from packet-level or decoded-frame data.

APPLIES TO MFormats

CEA-608 workflows needed a way to choose whether caption processing order follows packet-level data or decoded-frame data.

RESULT cc.display_order now lets applications select the CEA-608 processing order.

ADDED

V-Chip/XDS insertion and parsing support.

APPLIES TO MFormats

Applications could not generate and parse V-Chip/XDS metadata directly through the SDK workflow.

RESULT V-Chip/XDS insertion and parsing are now supported.

ADDED

HTML5 MLCG editor.

APPLIES TO MFormats

Applications that expose CG/graphics editing workflows did not have the new HTML5-based MLCG editor available as part of the SDK package.

RESULT The HTML5 MLCG editor is now included.

FIXED

CG overlay color rendering when overlay.thread=0.

APPLIES TO MFormats

CG overlays rendered with overlay.thread=0 could produce incorrect colors compared with the same CG content rendered through the normal overlay path.

RESULT CG overlay colors are preserved when overlay.thread=0 is used.

FIXED

CG overlay rendering for interlaced formats when overlay.thread=0.

APPLIES TO MFormats

CG items could disappear when `overlay.thread=0` was used together with interlaced video conversion, even though the same CG content was visible for progressive formats.

RESULT CG overlays render for interlaced formats when `overlay.thread=0` is used.

FIXED**Extra spacing between adjacent CG text runs in tickers.**

APPLIES TO MFormats

Adjacent CG text runs could get an extra inserted space. This broke punctuation and words where only part of the word used a different style.

RESULT Adjacent CG text runs keep their authored spacing.

FIXED**CEF opening an external browser when HTML overlay is used from two applications at the same time.**

APPLIES TO MFormats

Running HTML overlay from two applications at the same time could open an external browser window instead of keeping the page inside the SDK-controlled embedded browser path.

RESULT HTML overlay stays in the embedded browser path when multiple applications use it.

FIXED**CG RSS ticker text width and spacing mismatch.**

APPLIES TO MFormats

CG RSS ticker text could use different width or spacing metrics from a normal text ticker, so visually equivalent ticker content did not line up.

RESULT RSS ticker text metrics match normal CG ticker text more closely.

FIXED**Display of closed captions written to MXF files.**

APPLIES TO MFormats

Closed captions embedded into MXF output could fail to display after the file was written, making the recording look as if the caption payload had been lost.

RESULT Closed captions written into MXF output are preserved for display.

EXAMPLE - MFormats MXF output with embedded captions

```
string writerConfig =
    "format='mxf' embed_cc='true' anc_capture='false' " +
    "video::codec='mpeg2video' audio::codec='pcm_s16le' audio::ar='48000';"
writer.WriterSet(outputMxf, 0, writerConfig);
```

FIXED**Generated CEA-608 closed captions in files for VLC playback.**

APPLIES TO MFormats

Files written with generated CEA-608 captions could contain caption data that VLC did not display, even though the application attempted to embed captions during encoding.

RESULT Generated CEA-608 captions written by the SDK are compatible with VLC playback for this workflow.

EXAMPLE - MFormats embedding generated CEA-608 captions

```
ccDisplay.PropsSet("cc_type", "608");
string writerConfig =
    "format='mp4' embed_cc='true' scc_capture='false' " +
    "video::codec='libopenh264' video::b='5M' audio::codec='aac'";
writer.WriterSet(outputFile, 0, writerConfig);
```

FIXED**Blurred CG text when decoder.convert_planar=true.**

APPLIES TO MFormats

CG text could become blurred for certain text colors when decoder.convert_planar=true was enabled, even without scaling artifacts.

RESULT

CG text remains sharp with decoder.convert_planar=true.

FIXED**MCCDisplay CEA-608 character ordering for specific files.**

APPLIES TO MFormats

CEA-608 captions decoded through MCCDisplay could show characters in the wrong order for specific files, while reference playback displayed the text correctly.

RESULT

MCCDisplay preserves the expected CEA-608 character order for these files.

FIXED**CEF HTML plugin and WebCapture page loading on first start after OS boot.**

APPLIES TO MFormats

After a fresh OS boot, HTML plugin or WebCapture pages could fail to load on the first SDK start and require another attempt.

RESULT

HTML plugin and WebCapture pages load correctly on first start after boot.

FIXED**CEF audio mute and preview-control handling.**

APPLIES TO MFormats

CEF-backed preview/control workflows could handle mute state or preview control inconsistently, causing embedded-page audio or control state to differ from the application expectation.

RESULT

CEF mute and preview-control handling is consistent with SDK control state.

FIXED**CEF popup handling that could create an extra preview/window for new tabs.**

APPLIES TO MFormats

CEF popup or new-tab requests could create an extra preview/window instead of being handled inside the intended SDK browser workflow.

RESULT

CEF popup handling no longer creates the unwanted extra preview/window.

UPDATED**CEF to version 147.0.11.**

APPLIES TO MFormats

HTML overlay, CG/HTML, and WebCapture workflows rely on the embedded Chromium/CEF runtime, so browser compatibility and rendering behavior depend on that bundled runtime.

RESULT The embedded CEF runtime now uses 147.0.11.

FIXED**Smooth scheduled appearance for CG items shown with cg-props::show.**

APPLIES TO MFormats

Scheduling cg-props::show=yes displayed a hidden CG item immediately, ignoring the configured introduction time, while scheduled hiding faded correctly.

RESULT Scheduled CG show operations now use the configured intro fade duration.

FIXED**CG hang when hiding a ticker containing a non-RSS web page.**

APPLIES TO MFormats

Adding a regular web-page URL as ticker text and then hiding the ticker could block CG and the host application.

RESULT Ticker URL handling now rejects non-RSS pages instead of adding an unsupported web item.

Samples and Documentation

ADDED**MLCG REST control sample for MFormats workflows.**

APPLIES TO MFormats

Developers integrating MLCG graphics with MFormats did not have a ready REST API sample showing connection, session selection, timeline commands, and variable updates.

RESULT A new MFormats sample demonstrates MLCG REST control in an SDK workflow.

UPDATED**MFormats MLShader samples with the new shader-library project and shader catalog.**

APPLIES TO MFormats

The older MFormats shader sample did not show the current shader-library workflow with a catalog, metadata, and reusable HLSL examples.

RESULT The MFormats shader sample set was refreshed to the shader-library workflow.

UPDATED**Subtitle workflows split into dedicated Conversion and Generation samples.**

APPLIES TO MFormats

The former Subtitle Insertion sample combined file conversion and live subtitle generation in one project, making the two integration paths difficult to follow and extend.

RESULT Each SDK now includes a Subtitle Conversion sample for external SRT/ASS files and a Subtitle Generation sample for live-source workflows.