8KDVD Standard Specification v1.0: The Unified Technical White Paper

The Future of Ultra-High Definition Physical & Digital Media

1. Introduction

The **8KDVD** format is a high-fidelity video distribution standard designed to bridge the gap between legacy physical media and modern 8K computing power. It utilizes a dual-path navigation architecture: an **HTML5 Fallback Path** for universal browser compatibility and a **Structured XML Path** for dedicated 8KDVD hardware and software players.

2. Physical & Logical Disc Structure

An 8KDVD compliant volume **shall** utilize the UDF 2.60 file system. The root directory must contain the following mandatory structure to ensure cross-platform "Auto-Run" capability:

2.1 Directory Map

- **ROOT/**: Contains Universal_web_launcher.html and Launch_8KDVD_Windows.bat.
- **8KDVD_TS/**: The primary data bin.
    - **STREAM/**: Stores the encrypted or raw payload files (.EVO8, .EVO4, etc.).
    - **ADV_OBJ/**: Stores the index.xml for standalone players and UI assets.
- **CERTIFICATE/**: Stores the Certificate.html file (The "Master Key" for playback).
- **LICENCEINFO/**: Stores the LICENCEINFO.xml (Optional restriction manifest).

3. Video & Audio Specifications

To ensure smooth playback across the standard, all payloads must conform to the **VP9 Video / Opus Audio** stack.

3.1 Payload Identifiers

The 8KDVD standard uses resolution-specific suffixes to assist the player in auto-selecting the stream based on hardware capabilities:

Extension	Target Resolution	Codec	Recommended Bitrate
.EVO8	8K (7680×4320)	VP9	26.9 Mbps
.EVO4	4K (3840×2160)	VP9	14.9 Mbps
.EVOH	1080p (1920×1080)	VP9	6.0 Mbps
.3D4	4K 3D Anaglyph	VP9	14.9 Mbps

3.2 Container Mandate

While files are authored as .MP4 via FFmpeg, they **shall** be renamed to the .EVO suffixes. The internal container structure remains a standard MPEG-4 Part 14 stream, ensuring that 8KDVD-aware decoders can demux the VP9/Opus streams efficiently.

4. Proprietary Licensing & Security (The "C-Logic")

8KDVD implements a multi-stage verification process designed to protect content while remaining accessible to indie creators.

4.1 The Certificate Check

Before any media buffer is opened, the player **must** verify CERTIFICATE/Certificate.html.

- If the file is absent: Playback is **Hard-Locked**.
- If Licence: Yes is present: **Level 2 Verification** is triggered.

4.2 The "Dummy File" & Folder Lock (Level 2)

When Level 2 is active, the player performs a "Triple-Check":

1. **The Folder:** The /LICENCEINFO/ directory must be present.
2. **The Manifest:** LICENCEINFO/LICENCEINFO.xml must exist. (Note: This file is intentionally non-schema compliant to deter standard XML scrapers).
3. **The Dummy Suffix:** A **0 KB** file must exist in /8KDVD_TS/STREAM/. The filename must match the main feature + a capital **"C"** (e.g., PAYLOAD_01.EVO8C).

5. Navigation APIs

5.1 Standalone Path (XML)

Standalone players utilize the _8KDVD.Player API. This allows the index.xml to trigger deep-level hardware calls that are faster than standard HTML5 video tags.

5.2 Browser Path (HTML5)

The Universal_web_launcher.html provides a fallback for devices without a dedicated 8KDVD player (e.g., standard PCs running Chrome/Edge).

6. Hardware Performance Tiers (Playback Capabilities)

To ensure a consistent user experience, 8KDVD players **shall** identify the hardware's decoding capabilities and map them to the following Performance Tiers. If a disc is inserted and the hardware cannot meet the requirements of the primary .EVO8 stream, the player **should** automatically offer or switch to the highest supported resolution.

Tier	Designation	Minimum Decoder Requirement	Target Stream
Tier 1	**Ultra (8K)**	8K VP9 Hardware Acceleration (e.g., NVIDIA RTX 30+ / Apple M2)	.EVO8
Tier 2	**High (4K)**	4K VP9 Hardware Acceleration (e.g., Intel Iris Xe / RTX 20+)	.EVO4

Tier	Designation	Minimum Decoder Requirement	Target Stream
Tier 3	**Standard (1080p)**	Quad-core CPU with Software Decoding or Legacy GPU	.EVOH

Adaptive Logic: In the event of significant frame drops (>15% over 5 seconds), the 8KDVD player API (_8KDVD.Player.AutoScale) is authorized to switch to the next lower .EVO payload without halting playback.

7. Extended Audio & Subtitle Architecture

While the primary audio is embedded within the .EVO payload, 8KDVD supports "External Stream Injection" for multi-language support and director commentaries.

7.1 Audio Stream Naming

Secondary audio tracks shall be stored in the /8KDVD_TS/STREAM/ folder.

- **Format:** Opus 48000 Hz.
- **Naming:** PAYLOAD_01_AUD[X].OPUS (where X is the track index).
- **Reference:** These tracks are mapped via the index.xml using the <AudioTrack> tag.

7.2 Subtitle Synchronization

The /subtitles/ folder supports standard .srt files.

- **Mandatory Encoding:** UTF-8.
- **Naming:** Language codes shall follow ISO 639-1 (e.g., en.srt, fr.srt).

8. The Boot Sequence & Intro Mandate

To establish the format's identity, all 8KDVD players **shall** execute the following sequence upon disc detection:

1. **Integrity Check:** Verify CERTIFICATE/Certificate.html (If missing, abort).
2. **Branding Sequence (The "Intro"):** If a file named INTRO.EVOH exists in the /8KDVD_TS/STREAM/ folder, the player **must** play this file before loading the index.xml or Universal_web_launcher.html. This ensures the 8KDVD branding or studio logos are displayed.
3. **License Verification:** If Licence: Yes is detected, perform the 0KB Dummy File check.
4. **Menu Load:** Transition to the interactive UI.

9. 8KDVD Standard Player API (Full Specification)

This section defines the mandatory API that any 8KDVD-compliant player (Hardware or Software) **shall** support. This ensures that discs authored today remain compatible with all future players1.

9.1 Core Playback & Navigation

- **playMovie(startTime)**: Initializes the video buffer and begins playback at a specified integer in seconds2.
- **selectQuality(quality)**: Swaps the active payload (EVO8, EVO4, EVOH, or 3D4) while maintaining the current playhead position3.
- **seekTo(timeInSeconds)**: Direct instruction to move the playback position4.
- **pause() / resume()**: Toggles the playback state without flushing the buffer.
- **stop()**: Terminates the video engine and returns the user to the mainMenu5.
- **step(frames)**: Advances or reverses the video by a single frame (standard for professional-grade players).

9.2 Legacy-Standard Hardware Calls

These calls bridge the gap between 8KDVD and traditional disc formats:

- **_8KDVD.Player.Eject()**: Instructs physical hardware to open the disc tray.
- **_8KDVD.Player.SetParentalLevel(level)**: Compares the disc's metadata against player-stored restrictions (Levels 1-8).
- **_8KDVD.Player.GetAspect()**: Returns the current display's aspect ratio to ensure 8K content doesn't pillar-box incorrectly.
- **_8KDVD.Player.PersistentStore.Save(key, value)**: Allows a disc to save "Bookmarks" or "High Scores" to the player's internal memory/SD card.
- **_8KDVD.Player.PersistentStore.Load(key)**: Recalls saved data across different viewing sessions.

9.3 Advanced Media & Stream Logic

- **toggleSubtitles()**: Toggles the overlay of UTF-8 .srt files from the /subtitles/ directory6.
- **changeLanguage()**: Switches between internal and external .OPUS audio tracks7.
- **setSecondaryVideo(stream, x, y, w, h)**: Enables "Bonus View" / Picture-in-Picture (PiP) functionality, a staple of the HD DVD/Blu-ray experience.
- **getBitrate()**: Returns the real-time bitrate (in bps) to help the UI display a "Technical Info" overlay for enthusiasts.

9.4 Automated Security & System Calls

- **_8KDVD.Hardware.SilentLicenseCheck()**: Background task that verifies Certificate.html and the **0KB** PAYLOAD_01.EVO8C dummy file8.
- **_8KDVD.Hardware.GetTier()**: Queries GPU/CPU for decoding support (Tier 1-3)9.
- **openCertificate()**: Displays the Certificate.html overlay for ownership and social links10.

9.5 API Call Table for Developers

Call Name	Legacy Equivalent	8KDVD Purpose
selectQuality	N/A (New to 8K)	Swaps EVO suffixes to match hardware capability11.
playTitle	PlayTitle (DVD)	Identical to playMovie but can target different files12.
setParental	GPRM (HD DVD)	Locks content based on player-level settings.
PersistentStore	BUDA (Blu-ray)	Saves user data to the player's local storage.
CheckLicense	AACS (Blu-ray)	Proprietary 0KB dummy file/Certificate handshake13.

10. File System & Disc Mount Mandate

To ensure universal recognition across Windows, Linux, and Android, the 8KDVD standard dictates a strict "Waterfall" boot priority. **File System**: The disc must be authored using **UDF 2.60** for high-capacity BDXL/HD DVD compatibility. The hardware or software player **shall** scan for launch files in the following order:

1. **Level 1 (Standalone):** The player first attempts to parse and execute `8KDVD_TS/ADV_OBJ/index.xml` using the native 8KDVD Runtime Environment.
2. **Level 2 (Media Center):** If Level 1 is unavailable, the system checks for the **8KDVD Kodi App** integration to handle playback within the Kodi ecosystem.
3. **Level 3 (Web Browser):** If no native or media-center app is detected, the system executes `Universal_web_launcher.html` located at the disc root.
4. **Level 4 (Legacy Windows):** As a final fallback, the `Launch_8KDVD_Windows.bat` is executed to trigger the local Windows browser environment.
   +1

11. The "Dual-Menu" Resolution Logic

8KDVD manages navigation through two distinct methods to ensure playback on both high-end hardware and standard PCs:

- **The Standalone Path (`index.xml`):** This is the primary standard. It allows for deep integration with player APIs, such as `playTitle` and `seekTo`, and utilizes the `eightkdvdmenu` namespace for high-performance UI rendering.
- **The Web Path (`weblauncher.html`):** This fallback uses JavaScript to map the `selectedQuality` (8K, 4K, HD, 3D) to the corresponding `.EVO` suffix in the `STREAM` folder, providing an interactive experience in any Chromium-based browser.

12. HLS Integration & Segmented Playback

The `main.m3u8` file serves as the master manifest for high-end players that support adaptive bitrate switching:

- **Bandwidth Hierarchy:** The player reads defined bandwidths, such as 100Mbps for 8K (`.EV08`) and 20Mbps for 4K (`.EV04`), to match the stream to the hardware's decoding power.
- **Relative Pathing:** To maintain disc integrity, the playlist references streams via the `../STREAM/` relative directory.

13. Metadata & Metadata Linkage

The 8KDVD structure requires specific metadata directories to support professional playback features:

- **CLIPINF:** Stores metadata for individual video streams, including frame rates (typically 23.976 fps) and stream duration.
- **PLAYLIST:** Contains the `main.m3u8` for segmented or multi-resolution playback.
- **ADV_OBJ:** The central hub for UI assets, containing the `index.xml`, `weblauncher.html`, `logo.png`, and `background_8k.png`.
- **CERTIFICATE:** A mandatory folder containing `Certificate.html`, which holds the "Licence: Yes" trigger for the security handshake.

14. Chaptering & Time-Code Indexing

8KDVD supports two methods of chaptering to ensure creators have flexibility:

- **XML-Based (Preferred):** The `chapters.xml` defines specific time-stamped titles, such as "Chapter 1" at 00:07:00 and "Chapter 2" at 00:10:00.
- **Algorithmic (Fallback):** In the absence of an XML definition, the `weblauncher.html` logic mathematically generates a chapter link every 600 seconds (10 minutes) based on the total duration of the video.

15. Logo & Physical Branding Guidelines

To maintain a professional and unified format identity, all physical 8KDVD releases **shall** adhere to the following branding mandates:

15.1 Logo Dimensions and Placement

- **Front Cover:** The official 8KDVD logo must be placed in the top-right corner. It shall have a minimum width of **3.5 cm** to ensure legibility and brand presence.
- **Spine:** The 8KDVD logo must be featured on the spine of the case, oriented **sideways**. It shall have a minimum width (measured across the spine) of **2.0 cm**.
- **Disc Face:** The logo must be clearly visible on the disc artwork, typically centered or placed in the upper hemisphere.

15.2 Case Specifications (The "Red Box" Standard)

The physical housing of the disc is a primary identifier of the 8KDVD format.

- **Standard Case:** All releases **shall** utilize **Standard Solid Red Color Single DVD Cases**.
- **Dimensions:** Cases must maintain standard DVD height and depth to ensure they fit within existing consumer shelving systems.
- **Color Variations:** While solid red is the "Standard," creators may utilize **Yellow**, **Green**, or other colors to signify special content or editions.
- **Prohibition:** The use of **Black** cases is strongly discouraged and should be avoided to distinguish 8KDVD from legacy DVD formats.
- **Recommended Hardware:** The format recommends high-quality cases such as the **Mediaxpo Brand STANDARD Solid Red Single DVD Case** for retail consistency.

15.3 Copyright & URL Attribution

The back cover of every 8KDVD release must include a formal credit block:

- **Hierarchy:** The creator’s own copyright notice shall appear first.
- **Mandatory Link:** Immediately following the creator's notice, a formal credit to the format parent must be included: **"HDDVD-revived"** or **"8kdvd.hddvd-revived.com"**.
- **Typography:** This notice shall be set in a font size between **6 pt** and **10 pt**.

16. Compliance Certification

To ensure the "8KDVD Certified" badge is applied accurately, manufacturers must prove their players can navigate the file structure provided by the **8KDVD Maker** software.

16.1 The "Golden Disc" Test Suite

A hardware or software player is considered compliant only if it successfully passes the following logic tests:

1. **Handshake Test:** Automatically recognizes and parses the `Certificate.html` file.
2. **Security Trigger:** Blocks playback immediately if the **0 KB** `PAYLOAD_01.EVO8C` dummy file is missing or invalid.
3. **Boot Priority Check:** Correctly prioritizes `index.xml` (Standalone), then the Kodi app integration, then `Universal_web_launcher.html`.
4. **HLS Buffer Test:** Successfully parses `main.m3u8` and maintains a stable buffer for the **100 Mbps** 8K stream.

17. Multi-Stream Audio Implementation

This section details how the player handles supplementary audio tracks beyond the primary muxed stream.

- **Secondary Tracks:** Additional audio (commentaries, alternate languages) shall be stored as **Opus 48000 Hz** files in the `/8KDVD_TS/STREAM/` directory.
- **Naming Convention:** Tracks must follow the `PAYLOAD_01_AUD[X].OPUS` format for automated player recognition.
- **API Mapping:** These files must be indexed within the `index.xml` or `chapters.xml` to allow the `changeLanguage()` API call to switch between them seamlessly during playback.

8KDVD Creator’s Quick-Start Sheet

Follow this checklist to ensure your disc meets the official 8KDVD Standard.

1. Artwork & Branding (Visual Compliance)

- [ ] Front Cover: Official 8KDVD logo in the top-right corner (Min width: 3.5 cm).
- [ ] Spine: 8KDVD logo oriented sideways (Min width: 2.0 cm).
- [ ] Case: Housed in a Standard Solid Red DVD Case (or approved special edition colors).
- [ ] Back Cover Notice: Creator copyright followed by `8kdvd.hddvd-revived.com` (Font size: 6–10 pt).
- [ ] Disc Face: 8KDVD logo clearly visible on the printed artwork.

2. Video & Audio Encoding (The Payload)

- [ ] Video Codec: VP9 (Profile 2).
- [ ] Resolution Tiers: Mandatory `.EVO8` (8K), `.EVO4` (4K), and `.EVOH` (1080p).
- [ ] Audio Codec: Opus 48000 Hz (Stereo or Surround).
- [ ] Bitrate: Target 26.9 Mbps for 8K; 14.9 Mbps for 4K.

3. Logical Structure (The Master Folder)

Ensure the root of your disc contains exactly this hierarchy:

- [ ] Root: `Universal_web_launcher.html` and `Launch_8KDVD_Windows.bat`.
- [ ] 8KDVD_TS/STREAM: Contains your `.EVO8`, `.EVO4`, `.EVOH`, and `.3D4` files.
- [ ] 8KDVD_TS/ADV_OBJ: Contains `index.xml`, `weblauncher.html`, and `background_8k.png`.
- [ ] 8KDVD_TS/PLAYLIST: Contains the `main.m3u8` master manifest.
- [ ] CERTIFICATE: Contains `Certificate.html`.

4. Security Handshake (The "C-Logic")

- [ ] Certificate: Check if `Certificate.html` has `<p><strong>Licence:</strong> Yes</p>` if you wish to enable protection.
- [ ] Dummy File: If the license is "Yes," ensure a 0KB file named `PAYLOAD_01.EVO8C` is present in the `STREAM` folder.
- [ ] Verification: Test that the disc fails to play if the dummy file is removed.

5. Interactive Navigation

- [ ] Chapters: `chapters.xml` is present and timecodes match the video.

- [ ] Secondary Audio: Any external commentary tracks are named `PAYLOAD_01_AUD2.OPUS`.
- [ ] Kodi Support: Ensure the metadata is formatted for the 8KDVD Kodi app integration.

8KDVD Standard License & Format Agreement (v1.0)

1. INTRODUCTION & MISSION The 8KDVD standard, managed by **hddvd-revived.com** ("the Licensor"), is established to provide a high-fidelity, offline alternative for 8K video distribution. This Agreement governs the use of the proprietary 8KDVD disc structure, the 8KDVD Maker software, and associated branding. Our mission is to keep physical media alive by offering an accessible, low-barrier entry for creators while maintaining a unified format standard.

2. DEFINITION OF THE STANDARD The "8KDVD Standard" refers to the specific file directory structure, the HTML/Java-based navigation system, and the multi-platform launchers (Windows, macOS, Linux, Android) provided by the Licensor. Any disc or digital ISO utilizing this architecture must be identified as an **8KDVD**.

3. TIERED LICENSING MODEL To support the independent filmmaking community, licensing fees are scaled based on the production's Gross Production Budget:

- **3.1 Small-Scale & Independent Productions:** Any production with a total budget **under $50,000 USD** is granted a royalty-free license. No payment is required to use the 8KDVD logos, disc structure, or authoring tools.
- **3.2 Community-Funded Projects:** Projects funded via crowdfunding (e.g., Kickstarter, Indiegogo) are exempt from licensing fees regardless of the total raised, provided they adhere to branding guidelines.
- **3.3 Commercial Productions:** Productions with a budget of **$50,000 USD or more** are required to pay a licensing fee of **0.05%** of the gross production budget.
- **3.4 Liability Cap:** To remain the most affordable format in the industry, the maximum licensing fee is capped at **$50,000 USD** for high-budget studio productions.

4. MANDATORY BRANDING & IDENTITY To maintain the integrity of the 8KDVD ecosystem, all users must adhere to the following:

- **Naming Rights:** The product must be labeled as an **8KDVD**. Use of the name "8KDVD" is a trademark of hddvd-revived.com and is granted for use only with the proprietary structure.
- **Digital/Physical Attribution:** The official 8KDVD logo must be visible on the disc face and the rear of the packaging.
- **Mandatory URL:** Every 8KDVD release must clearly display the URL **8kdvd.hddvd-revived.com** on the physical box art (back cover) or within the primary digital menu.
- **Visual Packaging:** For physical releases, creators are strongly encouraged to use the **Red DVD-sized case** to ensure consumer recognition of the format.

5. HARDWARE & TECHNOLOGY INTEGRATION

- **5.1 Manufacturer Access:** Hardware developers are granted a **royalty-free window until December 31, 2030**, to integrate 8KDVD playback support into standalone players or drives.
- **5.2 Technical Specifications:** Enrollment as a hardware developer provides access to the full technical stack, provided the hardware maintains the 8KDVD branding.
- **5.3 Third-Party IP:** This license does *not* grant rights to “Blu-ray,” “HD DVD,” or “8K” generic logos owned by third-party conglomerates. Licensees are responsible for secondary licensing if they utilize those specific technologies.

6. RESTRICTIONS & TERMINATION The Licensor reserves the right to revoke the license if the 8KDVD name is used to distribute content that does not meet the 8K resolution standard (7680×4320) or if the branding is used to deceive consumers regarding the nature of the format.

DAY ONE STANDARD DATA

1. Introduction to 8KDVD Format

8KDVD is an advanced video format designed to support ultra-high-definition (UHD) content up to 8K resolution (7680x4320). It provides seamless integration with a modern web-based interface for user navigation and interaction, while simultaneously enabling deeper integration via XML-based menus for standalone devices. The format supports a range of resolutions and qualities, allowing content to scale between 1080p HD, 4K UHD, 8K, and 3D anaglyph versions.

This document outlines the technical structure of 8KDVD discs, their fallback HTML menus for compatibility, XML-based menus for enhanced standalone player features, and the technical codecs used for video and audio content creation.

---

2. Disc Structure

The 8KDVD structure contains several key directories and files that help manage the disc's content, metadata, and playback logic. The structure is as follows:

- **Root Directory**
  - `Universal_web_launcher.html`: Web-based fallback interface for HTML-capable players or browsers.
  - `Launch_8KDVD_Windows.bat`: Windows batch script to launch the disc's content.
- **8KDVD_TS**: Main directory for video content and metadata.
  - `ADV_OBJ`: Contains auxiliary objects such as XML menu definitions and media files like background images (`background_8k.png`).
  - `STREAM`: Video streams in different resolutions (`PAYLOAD_01.EVOH`, `PAYLOAD_01.EVO4`, `PAYLOAD_01.EVO8`, `PAYLOAD_01.3D4`).
  - `PLAYLIST`: Includes playlist files like `main.m3u8` for segmented playback.
  - `CLIPINF`: Metadata for video streams.
- **CERTIFICATE**: Authentication information for licensed players.
- **LICENSEINFO**: Licensing details.
- **subtitles**: Contains subtitle files such as `en.srt` for English subtitles.

The above structure supports both web-based and standalone playback

Disc Structure

```
¦   directory_tree.txt
¦   Universal_web_launcher.html
¦   Launch_8KDVD_Windows.bat
¦
+---8KDVD_TS
¦   +---ADV_OBJ
¦   ¦       background_8k.png
¦   ¦       index.xml
¦   ¦       logo.png
¦   ¦       weblauncher.html
¦   ¦
¦   +---CLIPINF
¦   +---PLAYLIST
¦   ¦       main.m3u8
¦   ¦
¦   +---STREAM
¦           PAYLOAD_01.EVO8
¦           PAYLOAD_01.EVO4
¦           PAYLOAD_01.EVOH
¦           PAYLOAD_01.EVOH
+---CERTIFICATE
+---LICENSEINFO
+---subtitles
        en.srt
```

---

3. Codecs and Video Stream Specifications

The video and audio streams on 8KDVD are encoded using the VP9 codec for video and Opus codec for audio, optimized for various resolutions:

- **1080p HD (PAYLOAD_01.EVOH)**:
    - **Video**: VP9 codec, 1920x1080 resolution, 23.976 fps.
    - **Audio**: Opus 48000 Hz stereo, 3072 kbps.
- **4K UHD (PAYLOAD_01.EVO4)**:
    - **Video**: VP9 codec, 3840x2160 resolution, 23.976 fps.
    - **Audio**: Opus 48000 Hz stereo, 3072 kbps.
- **8K UHD (PAYLOAD_01.EVO8)**:

    - **Video**: VP9 codec, 7680x4320 resolution, 23.976 fps.
    - **Audio**: Opus 48000 Hz stereo, 3072 kbps.
- **3D Anaglyph (PAYLOAD_01.3D4)**:
    - **Video**: VP9 codec, 3840x2160 resolution, 29.97 fps.
    - **Audio**: Opus 48000 Hz stereo, 3072 kbps

---

4. Content Creation with FFmpeg

To encode video streams for 8KDVD, the following FFmpeg commands can be used:

- **8K Video Creation**:

```
ffmpeg -i HalfwaytoHeaven8K.mp4 -c:v libvpx-vp9 -crf 18 -b:v 26900k -b:a
384k PAYLOAD_01.MP4

Then file extension changed to EVO8
```

- **4K Video Scaling**:

```
ffmpeg -i PAYLOAD_01.EVO8 -c:v libvpx-vp9 -crf 23 -b:v 14900k -b:a 384k
-vf scale=3840:2160 PAYLOAD_01.MP4
Then file extension changed to EVO4
```

- **1080p HD Video Scaling**:

```
ffmpeg -i PAYLOAD_01.EVO8 -c:v libvpx-vp9 -crf 23 -b:v 6000k -b:a 256k -vf
scale=1920:1080 PAYLOAD_01.,P4

Then file extension changed to EVOH
```

These commands use the VP9 codec for video encoding and Opus for audio, adjusting bitrates and scaling as necessary for the different quality levels.

---

5. HTML-based Fallback Menu

The web-based fallback menu, accessible via `weblauncher.html`, provides a simple and interactive interface for users. This menu allows for the following functionalities:

- **Play Movie**: Starts playback of the selected video stream.
- **Chapter Selection**: Allows users to navigate between chapters based on time intervals.
- **Settings**: Offers users the ability to change the video quality (8K, 4K, HD, or 3D) or access other settings like subtitles.

This HTML menu is designed with responsive elements, ensuring compatibility with modern browsers across multiple device types(weblauncher).

---

6. Standalone Player XML Menu System

The standalone player utilizes a more sophisticated XML-based system defined in the `index.xml` file. This system is designed to fully leverage the capabilities of the standalone players:

- **Playback Control**: XML-driven control for switching between different quality streams (8K, 4K, HD, 3D) using custom player APIs such as `_8KDVD.Player` and `_8KDVD.Playlist`.
- **Dynamic Menu Creation**: The XML structure dynamically creates settings menus, chapter lists, and quality selection interfaces by interacting with the player's runtime environment.
- **Advanced Features**: Unlike the HTML menu, the XML version integrates tightly with the player's internal mechanisms for subtitle management, audio track selection, and enhanced playback controls.

For instance, selecting a video quality or chapter in the XML menu executes player-specific commands like `playTitle` or `seekTo`, ensuring seamless integration with the standalone 8KDVD players(index).

---

7. Comparison: XML Menu vs. HTML Fallback

Feature	XML Menu	HTML Fallback
Quality Selection	Full player integration, dynamic control	JavaScript-based, limited to simple cases
Chapter Navigation	Real-time, API-driven control	Pre-defined chapters using HTML/JS
Subtitle Management	Full control over subtitle files	Basic toggle functionality
Audio Tracks	Dynamic audio track selection	None (limited to default audio track)
Player Integration	Deep integration with standalone player	Browser-based with limited control
Extras	Customizable through player APIs	Limited to simple navigation

The XML menu system provides a more immersive and customizable experience tailored for advanced players, while the HTML fallback ensures basic functionality on less capable platforms.

---

8. Windows-Based Application for Full XML Integration

A Windows-based application could leverage the XML-based menu to provide a more comprehensive experience compared to the HTML fallback. Using tools like **ElectronJS** or **Node.js**, the application could:

- Load and parse the `index.xml` file for advanced menu control.

- Integrate with an embedded video player (such as VLC or an HTML5 player) to fully utilize the capabilities defined in the XML, including dynamic quality selection, chapter navigation, and subtitle management.
- Allow real-time interaction with player APIs via JavaScript, simulating the deeper controls of standalone devices while running on a Windows platform.

8KDVD represents a flexible, scalable video format suitable for both web-based playback and advanced standalone players. By combining an HTML fallback with a fully-featured XML menu system, the format offers both broad compatibility and the ability to deliver a premium, customizable user experience.

This white paper highlights the technical underpinnings of 8KDVD, providing a roadmap for developers to optimize their content creation, encoding, and playback workflows

When an 8KDVD is inserted, the very first check is whether the CERTIFICATE.html file is present. If that file is missing, the disc will not play at all. If the certificate file is present, the player then looks inside it for a licence line. The licence line is optional. If it is not there, the disc will play normally. If the line is present and set to “No,” the disc will also play normally. If the line is present and set to “Yes,” then the player performs the licence verification process.

```
  <div class="info">
    <p><strong>Year:</strong> 1949</p>
    <p><strong>Studio:</strong> US Army</p>
    <p><strong>Director:</strong> None</p>
    <p><strong>Producer:</strong> US ARMY</p>
    <p><strong>Budget:</strong> 10000</p>
    <p><strong>Licence:</strong> Yes</p>
  </div>
/strong> Yes</p>
```

During licence verification, the player checks for the presence of the LICENCEINFO folder. If the folder does not exist, playback is blocked. If the folder does exist, the player looks for a LICENCEINFO.xml file inside it. That file defines the restrictions, such as which DVD region codes apply and whether DRM is locked. Alongside that, a dummy file must exist in the 8KDVD_TS folder. This dummy file must be zero kilobytes in size and must have the same filename as the main 8K video file except with a “c” added to the end of the extension.

```
## 8KDVD Licensing and Copy Protection System

### Certificate-Based Licensing

8KDVD implements a unique certificate-based licensing system that controls disc
playback based on the presence and content of specific files.

#### License Verification Process
```

Step 1: Certificate File Check
- **Required File**: `CERTIFICATE/Certificate.html`
- **Critical**: If this file is missing, the disc will NOT play at all
- **Purpose**: Contains ownership information and optional license line

Step 2: License Line Verification
The player examines the Certificate.html file for a license line in the info div:
- **No License Line**: Disc plays normally without restrictions
- **License Line = "No"**: Disc plays normally without restrictions
- **License Line = "Yes"**: Triggers full license verification process
- **Location**: Inside `<div class="info">` section as `<p><strong>Licence:</strong> Yes</p>`
- **Case Sensitivity**: Not case sensitive ("Yes", "yes", "YES" all work)

Step 3: License Verification (If Required)
When license line is set to "Yes", the player checks for:

1. **LICENCEINFO Folder**: Must exist at disc root
2. **LICENCEINFO.xml File**: Must exist inside LICENCEINFO folder
3. **Dummy File**: Must exist in 8KDVD_TS folder
   - **Name**: Same as main 8K video file + "C" at end of extension (CAPS)
   - **Size**: Must be exactly 0 kilobytes (be suspicious if above 5KB)
   - **Example**: If main file is `PAYLOAD_01.EVO8`, dummy must be `PAYLOAD_01.EVO8C`
   - **Quantity**: Only ONE dummy file per disc (always for PAYLOAD_01)

Implementation Requirements

Certificate Parser:
```javascript
function checkCertificate() {
  const certPath = 'CERTIFICATE/Certificate.html';

  if (!fileExists(certPath)) {
    throw new Error('Certificate file missing - playback blocked');
  }

  const certContent = readFile(certPath);
  const licenseLine = extractLicenseLine(certContent);

  if (licenseLine === null || licenseLine === 'No') {
    return { requiresVerification: false };
  }

  if (licenseLine === 'Yes') {
    return { requiresVerification: true };
  }
}

function verifyLicense() {
  // Check LICENCEINFO folder
```

```
  if (!folderExists('LICENCEINFO')) {
    throw new Error('LICENCEINFO folder missing - playback blocked');
  }

  // Check LICENCEINFO.xml
  if (!fileExists('LICENCEINFO/LICENCEINFO.xml')) {
    throw new Error('LICENCEINFO.xml missing - playback blocked');
  }

  // Check dummy file
  const dummyFile = findMainVideoFile() + 'c';
  if (!fileExists(`8KDVD_TS/${dummyFile}`) || getFileSize(dummyFile) !== 0) {
    throw new Error('Dummy file missing or invalid - playback blocked');
  }

  return parseLicenseRestrictions('LICENCEINFO/LICENCEINFO.xml');
}
```

License Restrictions Parser:
```javascript
function parseLicenseRestrictions(xmlPath) {
  const xmlContent = readFile(xmlPath);
  const restrictions = {
    regionCodes: [],
    drmEnabled: false,
    copyProtection: false
  };

  // Parse XML for restrictions
  // Implementation depends on LICENCEINFO.xml schema

  return restrictions;
}
```

License File Structure

Certificate.html Structure:
```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Certificate - [Title]</title>
</head>
<body>
  <h1>[Title] - Certificate</h1>

  <!-- License line goes in info div below -->
```

```
  <div class="statement">
    <h2>Ownership and Protection Statement</h2>
    <!-- Ownership and copyright information -->
  </div>

  <div class="info">
    <p><strong>Year:</strong> [Year]</p>
    <p><strong>Studio:</strong> [Studio]</p>
    <p><strong>Director:</strong> [Director]</p>
    <p><strong>Producer:</strong> [Producer]</p>
    <p><strong>Budget:</strong> [Budget]</p>
    <!-- Optional license line -->
    <!-- <p><strong>Licence:</strong> Yes</p> -->
  </div>

  <div class="links">
    <h2>Official Links</h2>
    <!-- Social media and official links -->
  </div>
</body>
</html>
```

LICENCEINFO.xml Structure:
```xml
Region="1"
DRM="Lock"
```
Notes:
- **No XML Schema**: Not real XML, just two simple lines
- **Region**: Same numbering as DVD regions (1-8)
- **DRM**: Can be "Lock" or "Unlock" (case sensitive)
- **DRM="Unlock"**: No DRM, dummy file can be missing
- **Both lines optional**: Unless license line says "Yes"

Security Implications

Blocking Conditions:
1. Certificate.html file missing
2. License line = "Yes" AND LICENCEINFO folder missing
3. License line = "Yes" AND LICENCEINFO.xml missing
4. License line = "Yes" AND dummy file missing or invalid size

Normal Playback Conditions:
1. Certificate.html present, no license line
2. Certificate.html present, license line = "No"
3. Certificate.html present, license line = "Yes", all verification files present

Implementation Priority

This licensing system must be implemented as the **first check** when an 8KDVD is inserted, before any other disc processing occurs. Failure to implement this correctly will result in complete playback failure for licensed content.

API Reference

Player APIs

Core Player Object
```javascript
// _8KDVD.Player - Main player control
_8KDVD.Player = {
  play: function() { /* Start playback */ },
  pause: function() { /* Pause playback */ },
  stop: function() { /* Stop playback */ },
  seek: function(time) { /* Seek to time position */ },
  setQuality: function(quality) { /* Set video quality */ }
};
```

Playlist Management
```javascript
// _8KDVD.Playlist - Playlist control
_8KDVD.Playlist = {
  load: function(playlist) { /* Load playlist */ },
  next: function() { /* Next item */ },
  previous: function() { /* Previous item */ },
  getCurrent: function() { /* Get current item */ }
};
```

Navigation Functions
```javascript
// playTitle() - Play specific content
function playTitle(titleId, startTime) {
  // Implementation
}

// seekTo() - Seek to specific time
function seekTo(timeInSeconds) {
  // Implementation
}
```

Menu System APIs

Quality Selection
```javascript
function selectQuality(quality) {
```

```
  const qualityMap = {
    '8K': 'EVO8',
    '4K': 'EVO4',
    'HD': 'EVOH',
    '3D': '3D4'
  };

  const streamType = qualityMap[quality];
  if (streamType) {
    loadVideoStream(streamType);
    updateQualityIndicator(quality);
  }
}
```

Chapter Navigation
```javascript
function selectChapter(chapterIndex) {
  const chapters = loadChapters();
  if (chapters[chapterIndex]) {
    const startTime = parseTime(chapters[chapterIndex].time);
    playMovie(startTime);
  }
}
```

Implementation Checklist

Prerequisites
- [ ] VP9 codec support (libvpx-vp9)
- [ ] Opus codec support (libopus)
- [ ] UDF 2.6 file system support
- [ ] HTML5/XML parsing capabilities
- [ ] HLS playlist support
- [ ] 8K video decoding hardware/software
- [ ] AACS 2.0 copy protection handling
- [ ] **Certificate-based licensing system** (CRITICAL - must be first check)

Core Features
- [ ] **Certificate file verification** (FIRST CHECK - blocks all playback if missing)
- [ ] License line parsing and verification
- [ ] LICENCEINFO folder and XML file validation
- [ ] Dummy file verification (0KB with 'c' extension)
- [ ] Automatic disc detection
- [ ] XML menu parsing with namespace support
- [ ] HTML fallback menu rendering
- [ ] Quality selection (8K, 4K, HD, 3D)
- [ ] Chapter navigation

- [ ] Subtitle support
- [ ] Settings management
- [ ] Video playback with seek support

Advanced Features
- [ ] HLS adaptive streaming
- [ ] Dynamic menu creation
- [ ] Player API integration
- [ ] Multi-language support
- [ ] Background image rendering
- [ ] Logo display
- [ ] Certificate access
- [ ] Social links integration

Testing Requirements
- [ ] 8K video playback performance
- [ ] Menu responsiveness
- [ ] Quality switching
- [ ] Chapter navigation accuracy
- [ ] Subtitle synchronization
- [ ] Cross-platform compatibility
- [ ] Error handling
- [ ] Memory usage optimization

Performance Targets
- **8K Playback**: 60+ FPS on high-end hardware
- **Menu Loading**: <2 seconds
- **Quality Switching**: <1 second
- **Chapter Seeking**: <500ms
- **Memory Usage**: <4GB for 8K content
- **CPU Usage**: <80% on quad-core systems

Conclusion

This technical implementation guide provides comprehensive specifications for developing 8KDVD support in a Windows media player. The format combines cutting-edge video technology (VP9/Opus) with web-standard menu systems for maximum compatibility and flexibility.

Key implementation priorities:
1. **VP9/Opus codec integration** for video/audio playback
2. **XML menu parsing** with proper namespace handling
3. **HTML fallback system** for broad compatibility
4. **HLS playlist support** for adaptive streaming
5. **8K video optimization** for performance

The modular architecture allows for incremental implementation, starting with basic playback and adding advanced features progressively.

If the licence line in the certificate file was set to “Yes” and either the folder is missing, the XML file is missing, or the dummy file is missing, playback is blocked. If all the required elements are in place, the defined restrictions are applied. If the certificate file is present but contains no licence line, or the line says “No,” then the disc will simply play without any restrictions.