

Copyright and License

© 2026 InnoVadens, LLC. All rights reserved.

This specification is licensed under the Creative Commons Attribution–NoDerivatives 4.0 International License (CC BY-ND 4.0).

You are free to copy and redistribute this document in any medium or format, provided that you:

- Attribute the work to InnoVadens, LLC, and
- Do not remix, transform, or build upon the material.

No modifications of this specification may be distributed. Implementations may reference this specification, but derivative specifications or altered versions may not be published under the ADAC name or imply ADAC endorsement.

A copy of the license is available at: <https://creativecommons.org/licenses/by-nd/4.0/>

Status of This Document

This document defines the ADAC Specification for version 0.9.0-draft.1.

Status: Draft This profile is functionally complete but has not yet undergone full review. It is released for evaluation, testing, and community feedback. Breaking changes may occur before the 1.0.0 stable release.

InnoVadens, LLC is the steward of this specification.

This specification may be referenced by third-party software, libraries, and services. However, no external party may publish modified versions of this specification or claim compatibility beyond what is explicitly defined in the normative text.

Normative requirements in this document use the keywords MUST, MUST NOT, SHOULD, SHOULD NOT, and MAY as defined in RFC 2119.

Trademark Notice

"ADAC" and the ADAC logo are trademarks of InnoVadens, LLC. Use of the ADAC name in derivative specifications, competing standards, or modified versions of this document is prohibited.

Software implementations may state that they "implement the ADAC Specification" provided such statements are accurate and not misleading.

Revision History

Version 0.9.0-draft.1 — Draft release for evaluation, testing, and community feedback.

ADAC-Production v0.9.0

Production Asset Management Profile Specification

Domain Profile Extension for Film, Television, and Digital Media Production

Editor & Steward

John Vaden
Genealogist and Technical Workflow Contributor
InnoVadens, LLC — Cedar Lake, Michigan, USA

Contributors

AI-assisted drafting and refinement
Community feedback (ongoing)

Status

This document is maintained by InnoVadens, LLC as the official steward of the ADAC Standard.

John Vaden has decades of experience in genealogical research and in contributing to standards, workflows, and data practices across analyst and software development teams. He serves as the editor and steward of the ADAC Standard. InnoVadens, LLC is the legal steward of ADAC.

Version: v0.9.0 | **Status:** Draft | **Date:** 2026

Lead Author: John Vaden

Organization: InnoVadens, LLC

Requires: ADAC Core \geq 1.0

© 2026 InnoVadens, LLC. All rights reserved.

Status: Draft This profile is functionally complete but has not yet undergone jurisdictional legal review. It is released for evaluation, testing, and community feedback. Breaking changes may occur before the 1.0.0 legal-stable release.

Abstract

This document specifies ADAC-Production 0.9.0-draft.1, a domain profile extension to the Archival Digital Asset Container (ADAC) Core format for use in professional film, television, commercial, and digital media production. The profile defines a canonical **profile metadata file** (`metadata/profiles/production.json`), five **linked entity** types in the `production:` namespace, **edit pipeline** operation extensions for production workflows, and **region annotation** conventions for image, temporal, and multi-track audio regions. The profile is purely additive and does not redefine any ADAC Core field. Conformant implementations **MUST** satisfy all normative requirements of both this specification and ADAC Core ≥ 1.0 .

§1 Introduction

§1.1 Purpose

The ADAC-Production profile enables film, television, commercial, and digital media production organizations to package raw footage, audio tracks, edit pipelines, production metadata, and department-handoff provenance within a single self-describing ADAC **container**. Production assets managed with this profile carry full context through every stage of the production lifecycle — from camera acquisition through editorial, color, sound, VFX, final assembly, and archival delivery — without dependence on external databases or proprietary project management systems.

The profile introduces a standardized vocabulary for production-domain concepts (clips, slates, takes, VFX plates, and production audio), a canonical department-routing mechanism within the container's **provenance log**, and structured edit pipeline operations that extend the ADAC Core non-destructive editing model to address the specific operational requirements of professional media production.

§1.2 Target Domain

The target domain of this profile is professional media production, including but not limited to: feature film studios, television broadcasters, commercial production companies, post-production facilities, streaming content producers, and digital media organizations. The profile addresses the full production asset lifecycle: acquisition, editorial, post-production, delivery, and archival.

The profile is designed for use in both large-scale studio productions and independent productions requiring a portable, self-contained, standards-based asset management solution that travels with the **master file** throughout the production pipeline.

§1.3 Relationship to ADAC Core

This profile is a purely additive extension to ADAC Core ≥ 1.0 . It MUST NOT redefine any ADAC Core field or data structure. All core infrastructure — including the **manifest**, **fixity** checksums, **provenance log**, master file storage conventions, and **derivative** management — is inherited from ADAC Core without modification.

The profile interacts with ADAC Core via the profile wrapper file located at `metadata/profiles/production.json` (see §4.1) and via linked entities embedded in region annotation files stored in the `regions/` directory (see §5 and §7). For core infrastructure definitions, implementations MUST consult ADAC Core ≥ 1.0 ; see ADAC Core §2 for conformance, ADAC Core §4 for the edit pipeline model, and ADAC Core §6 for provenance logging.

§1.4 Scope

This profile covers the following asset categories:

- Moving image assets: digital cinema camera originals, film scans, high-definition and ultra-high-definition video recordings
- Production audio assets: multi-track location recordings, wild sound, and production music
- Still frame assets originating from production: slate frames, camera reference frames, and production stills
- Edit decision list (EDL) files stored as ADAC edit pipeline files
- Production metadata: project identification, asset classification, department routing, and delivery specifications

The following are explicitly **out of scope** for this profile: distribution and delivery packaging standards (IMF, DCP, broadcast delivery specifications) and theatrical exhibition systems. Those are delivery formats, not archival containers. This profile provides the archival metadata structure for assets that are subsequently packaged into those delivery formats.

§1.5 Audience

This specification is addressed to the following audiences:

- **Production supervisors** who direct asset management workflows across production departments
- **Digital image technicians (DITs)** who create and manage ADAC containers on-set
- **Editorial staff** who consume and modify ADAC containers in post-production
- **Post-production coordinators** who manage department handoffs and delivery workflows
- **Archivists** who receive and preserve production assets at the conclusion of post-production
- **Software engineers** building production asset management tools that implement the ADAC-Production profile

§2 Conformance

§2.1 RFC 2119 Keywords

The key words **MUST**, **MUST NOT**, **REQUIRED**, **SHALL**, **SHALL NOT**, **SHOULD**, **SHOULD NOT**, **RECOMMENDED**, **MAY**, and **OPTIONAL** in this document are to be interpreted as described in [RFC2119] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

§2.2 Conformance Levels

This profile defines a single conformance level. A conformant ADAC-Production implementation **MUST** satisfy all **MUST** and **MUST NOT** requirements stated in this specification. A conformant ADAC-Production implementation **MUST** also satisfy all normative requirements of ADAC Core ≥ 1.0 .

There are no partial conformance sub-levels. A container either is or is not ADAC-Production conformant. A container that satisfies some but not all normative requirements of this specification **MUST NOT** be described as ADAC-Production conformant.

NOTE — Production assets handled by the non-destructive edit pipeline defined in ADAC Core §4.5 are fully compatible with this profile's §6 extensions. Existing ADAC Core containers carrying production assets may be upgraded to ADAC-Production conformance by adding the `metadata/profiles/production.json` file and updating the manifest accordingly.

§2.3 Normative vs. Informative Text

Normative requirements are stated using the RFC 2119 key words defined in §2.1. All other text is informative. Informative passages that provide explanatory context are prefixed with the label **NOTE** — and set in a callout box. Illustrative examples are prefixed with the label **EXAMPLE** —, marked (*INFORMATIVE*), and set in a callout box. Informative content does not establish conformance requirements.

§3 Terminology

The following terms are defined for use in this specification. All terms are listed alphabetically. Terms defined in ADAC Core ≥ 1.0 retain their core definitions; this section defines only terms introduced or specialized by this profile. Defined terms are set in bold on their first normative use.

Term	Definition
acquisition	The process of recording or capturing a master file on-set or on-location, resulting in a raw, unprocessed asset suitable for storage as a master file within an ADAC container.
assembly cut	An initial editorial arrangement of selected takes without final timing or transitions, stored as an edit pipeline within the container. An assembly cut carries the status <code>assembly-cut</code> in its edit pipeline descriptor.
camera roll	A sequential identifier assigned to a physical or virtual recording medium — such as a film roll, solid-state memory card, or recording magazine — during production. The camera roll identifier is used to group clips originating from the same recording medium.
clip	A contiguous segment of a master file, typically representing a single continuous recording from a single camera at a single take, identified by a timecode range and associated with a slate. A clip is annotated using the <code>production:clip</code> linked entity type (see §5.1).
color grade	A creative and corrective process applied to a master file or derivative that adjusts color, contrast, and tonal characteristics without modifying the master file itself. Color grade operations are recorded in the edit pipeline.
dailies	Low-resolution derivatives produced from raw master files immediately after acquisition for editorial review and approval. Dailies are stored in the <code>derivatives/</code> directory of the ADAC container and carry the asset category <code>dailies</code> .
department handoff	A transfer of a container or its contained assets from one production department to another, recorded as a provenance log event. Department handoffs are the primary mechanism by which the container's custody chain is maintained across production stages.
digital image technician (DIT)	A specialist responsible for on-set data management, quality control, and creation of dailies. The DIT is typically the entity that creates the initial ADAC container and populates the production profile metadata file.
edit decision list (EDL)	A structured document specifying the sequence of clips and transitions that constitute an editorial assembly, stored as an edit pipeline file within the <code>edits/</code> directory of the container. EDL interoperability conventions are specified in §6.3.
final cut	A locked editorial assembly that has received all required approvals and is cleared for post-production processing. Its edit pipeline is marked with status <code>locked-picture</code> .
production	The organizational and creative effort to which this profile's assets belong, identified by a unique production identifier (<code>productionId</code>) as defined in §4.3.
rough cut	An intermediate editorial assembly that preserves creative options without final approval. Its edit pipeline is marked with status <code>rough-cut</code> .

Term	Definition
scene	A narrative unit identifying the dramatic context of a recorded clip, referenced by a scene number in accordance with the production's script breakdown.
slate	A physical or virtual clapperboard device used to identify a take on-camera and to provide a synchronization point for audio and video tracks. The visual frame containing the slate SHOULD be annotated as a region of type <code>production:slate</code> (see §5.2).
take	A single camera recording attempt for a given scene and setup, identified by a take number. Full take-level metadata is captured using the <code>production:take</code> linked entity type (see §5.3).
timecode	A numeric addressing system for audio and video frames, expressed in SMPTE timecode format HH:MM:SS:FF where FF represents the frame count within the current second. Drop-frame and non-drop-frame conventions are specified in §6.2.
VFX plate	A master file recorded specifically for visual effects compositing, characterized by requirements for controlled lighting, tracking markers, and chromatic range. VFX plates are annotated using the <code>production:vfx</code> linked entity type (see §5.4).
visual effects (VFX)	Digital image modifications applied to a master file or derivative in post-production. VFX regions and plate metadata are described using the <code>production:vfx</code> linked entity type (see §5.4).

§4 Profile Metadata File

§4.1 File Location

The production profile metadata file MUST be located at the path `metadata/profiles/production.json` within the ADAC container. The file name MUST be exactly `production.json` in lowercase. This file MUST be listed in the container's manifest within the `metadata.profiles` array, as required by ADAC Core ≥ 1.0; see ADAC Core §3.2 for manifest structure.

§4.2 Wrapper Schema

The `production.json` file MUST be a valid JSON object conforming to JSON Schema Draft 2020-12 [JSON-SCHEMA]. The object MUST include the following required properties:

- `profileName` (string): MUST have the value "ADAC-Production".
- `profileVersion` (string): MUST have the value "0.9.0-draft.1".
- `schemaRef` (string, URI): MUST reference the canonical schema URI for this profile version.

The following JSON Schema defines the complete wrapper structure for the production profile file:

```

{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://adac-standard.com/schemas/production/production-
profile.schema.json",
  "type": "object",
  "description": "ADAC-Production 0.9.0-draft.1 profile metadata file schema.",
  "required": ["profileName", "profileVersion", "schemaRef", "project",
"assetClassification", "departmentStatus"],
  "properties": {
    "profileName": { "type": "string", "const": "ADAC-Production" },
    "profileVersion": { "type": "string", "const": "0.9.0-draft.1" },
    "schemaRef": { "type": "string", "format": "uri" },
    "project": { "$ref": "#/$defs/project" },
    "assetClassification": { "$ref": "#/$defs/assetClassification" },
    "departmentStatus": { "$ref": "#/$defs/departmentStatus" },
    "deliverySpec": { "$ref": "#/$defs/deliverySpec" }
  }
}

```

EXAMPLE — Minimal wrapper object (INFORMATIVE)

```

{
  "profileName": "ADAC-Production",
  "profileVersion": "0.9.0-draft.1",
  "schemaRef": "https://adac-standard.com/schemas/production/production-
profile.schema.json"
}

```

§4.3 Project Identification

The project property of the profile metadata file MUST be a JSON object. The project object MUST include the following required properties:

- **productionId** (string): MUST be a UUID version 4 formatted as lowercase hyphenated (e.g., "a1b2c3d4-e5f6-7890-abcd-ef1234567890"). This value is the unique identifier for the production project and MUST remain stable across all containers belonging to the same production.
- **productionTitle** (string): The working title of the production.
- **productionType** (string, enum): MUST be one of `feature-film` | `television-series` | `television-episode` | `commercial` | `short-film` | `documentary` | `digital-series` | `other`.
- **productionCompany** (string): The name of the producing entity.
- **shootingFormat** (string, enum): MUST be one of `digital-cinema` | `film-35mm` | `film-16mm` | `hd-video` | `uhd-video` | `vr-360` | `other`.

The following properties are OPTIONAL within the project object:

- `episodeId` (string): For episodic content, the episode identifier.
- `seasonId` (string): For episodic content, the season identifier.
- `directorName` (string): The name of the director of the production.
- `cinematographerName` (string): The name of the director of photography.

§4.4 Asset Classification

The `assetClassification` property MUST be a JSON object. It MUST include the following required property:

- `assetCategory` (string, enum): MUST be one of `raw-footage` | `audio` | `edit-decision-list` | `vfx-plate` | `dailies` | `color-graded` | `locked-picture` | `delivery-master` | `archival-master` | `other`.

The following properties are OPTIONAL within `assetClassification`:

- `cameraRoll` (string): The camera roll identifier for the recording medium from which this master file was acquired.
- `sceneNumber` (string): The scene number from the production's script breakdown.
- `takeNumber` (string): The take number for this recording.
- `cameraDesignation` (string): The camera unit designation (e.g., "A", "B") for multi-camera productions.
- `timecodeIn` (string): SMPTE timecode of the first usable frame of the master file, in format HH:MM:SS:FF (see §6.2).
- `timecodeOut` (string): SMPTE timecode of the last usable frame of the master file.

NOTE — `timecodeIn` and `timecodeOut` define the usable range of the master file for editorial purposes. They are metadata fields and do NOT modify the master file itself. The complete master file, including any pre-roll and post-roll outside this range, is preserved intact in accordance with the ADAC Core master file immutability requirement.

§4.5 Department Routing

The `departmentStatus` property MUST be a JSON object. It MUST include the following required sub-properties:

- `currentDepartment` (string, enum): MUST be one of `acquisition` | `editorial` | `color` | `sound` | `vfx` | `delivery` | `archive`.
- `approvedForNextDepartment` (boolean): Indicates whether the asset has received approval for handoff to the next department in the pipeline.
- `approvalDate` (string): ISO 8601 UTC date-time at which approval was granted (e.g., "2026-05-01T10:00:00Z"). MUST be present when `approvedForNextDepartment` is `true`.
- `approvedBy` (string): The name or identifier of the individual who granted approval. MUST be present when `approvedForNextDepartment` is `true`.

The `departmentNotes` property is OPTIONAL. When present, it MUST be a string containing a free-text note from the current department to the receiving department.

§4.6 Delivery Specifications

The `deliverySpec` property is OPTIONAL. When present, it MUST be a JSON object containing one or more of the following OPTIONAL properties:

- `deliveryFormat` (string, enum): One of `imf` | `dcp` | `broadcast-mxf` | `streaming-h264` | `streaming-h265` | `prores` | `dnxhr` | `other`.
- `deliveryFrameRate` (string, enum): One of `23.976` | `24` | `25` | `29.97` | `30` | `48` | `50` | `59.94` | `60`.
- `deliveryResolution` (string, enum): One of `4k-dci` | `4k-uhd` | `2k-dci` | `1080p` | `720p` | `other`.
- `colorSpace` (string, enum): One of `rec2020` | `rec709` | `dci-p3` | `aces` | `srgb` | `other`.
- `dynamicRange` (string, enum): One of `hdr10` | `hdr10plus` | `dolby-vision` | `hlg` | `sdr` | `other`.

§5 Linked Entity Definitions

This section defines the five linked entity types introduced by the ADAC-Production profile. All types use the `production:` namespace prefix. Linked entities are embedded within region annotation files in the `regions/` directory of the ADAC container in accordance with ADAC Core ≥ 1.0 linked entity conventions; see ADAC Core §5 for the linked entity model. Each type is defined by a JSON Schema conforming to JSON Schema Draft 2020-12 [JSON-SCHEMA].

§5.1 `production:clip`

Type identifier: `production:clip`

Purpose: Annotates a temporal region of a master file as a discrete clip, associating it with its slate information, take number, and editorial selects status. This is the primary linked entity type for video clip annotation in the ADAC-Production profile.

Property	Type	Required	Description
<code>clipId</code>	string	REQUIRED	Unique clip identifier in kebab-case (e.g., <code>clip-001</code>). MUST be unique within the container.
<code>sceneNumber</code>	string	REQUIRED	The scene number from the production's script breakdown.
<code>takeNumber</code>	string	REQUIRED	The take number for this clip.
<code>cameraDesignation</code>	string	REQUIRED	The camera unit designation (e.g., "A", "B").

Property	Type	Required	Description
timecodeIn	string	REQUIRED	SMPTE timecode of the first frame of the clip in format HH:MM:SS:FF (see §6.2).
timecodeOut	string	REQUIRED	SMPTE timecode of the last frame of the clip.
circled	boolean	OPTIONAL	When true, indicates the take has been selected by the director or camera operator as preferred. A circled take SHOULD be offered as the primary option during editorial assembly.
transcriptionNotes	string	OPTIONAL	On-set notes associated with this clip, transcribed from the camera report or script supervisor's notes.
colorTemperature	integer	OPTIONAL	Kelvin value of the dominant light source as recorded on-set.

JSON Schema:

```
{
  "$schema": "https://json-schema.org/draft/2020-12/schema",
  "$id": "https://adac-standard.com/schemas/production/clip.schema.json",
  "type": "object",
  "description": "ADAC-Production linked entity: a discrete clip region of a master file.",
  "required": ["clipId", "sceneNumber", "takeNumber", "cameraDesignation", "timecodeIn", "timecodeOut"],
  "properties": {
    "clipId": { "type": "string", "description": "Unique clip ID, kebab-case." },
    "sceneNumber": { "type": "string", "description": "Scene number from script breakdown." },
    "takeNumber": { "type": "string", "description": "Take number." },
    "cameraDesignation": { "type": "string", "description": "Camera unit designation." },
    "timecodeIn": { "type": "string", "description": "SMPTE timecode of first frame, HH:MM:SS:FF." },
    "timecodeOut": { "type": "string", "description": "SMPTE timecode of last frame, HH:MM:SS:FF." },
    "circled": { "type": "boolean", "description": "True if this take is director-selected as preferred." },
    "transcriptionNotes": { "type": "string", "description": "On-set notes for this clip." },
    "colorTemperature": { "type": "integer", "description": "Dominant light source color temperature in Kelvin." }
  }
}
```

EXAMPLE — production:clip linked entity instance (INFORMATIVE)

```
{
  "entityType": "production:clip",
  "clipId": "clip-001",
  "sceneNumber": "42A",
  "takeNumber": "3",
  "cameraDesignation": "A",
  "timecodeIn": "10:22:14:00",
  "timecodeOut": "10:23:08:12",
  "circled": true,
  "transcriptionNotes": "Best performance. Slight hair movement at end -
check in post.",
  "colorTemperature": 5600
}
```

§5.2 production:slate

Type identifier: production:slate

Purpose: Annotates an image region containing a visible slate or clapperboard, capturing the slate data that appears on-camera. This linked entity type provides a machine-readable representation of the information displayed on the physical or virtual slate at the head of each take.

Property	Type	Required	Description
slateId	string	REQUIRED	Unique slate identifier, kebab-case (e.g., region-slate-01).
productionTitle	string	REQUIRED	Production title as it appears on the slate.
sceneNumber	string	REQUIRED	Scene number as it appears on the slate.
takeNumber	string	REQUIRED	Take number as it appears on the slate.
cameraDesignation	string	REQUIRED	Camera unit designation as it appears on the slate.
roll	string	REQUIRED	Camera roll identifier as it appears on the slate.
date	string	REQUIRED	Shoot date in format YYYY-MM-DD.
syncType	string (enum)	REQUIRED	Synchronization method: manual for hand-clap, electronic for MOS or genlock sync.
directorName	string	OPTIONAL	Director's name as it appears on the slate.
cinematographerName	string	OPTIONAL	Director of photography's name as it appears on the slate.

EXAMPLE — production:slate linked entity instance (INFORMATIVE)

```
{
  "entityType": "production:slate",
  "slateId": "region-slate-01",
  "productionTitle": "The Last Meridian",
  "directorName": "A. Vasquez",
  "cinematographerName": "C. Park",
  "sceneNumber": "42A",
  "takeNumber": "3",
  "cameraDesignation": "A",
  "roll": "A014",
  "date": "2026-04-15",
  "syncType": "electronic"
}
```

§5.3 production:take

Type identifier: production:take

Purpose: Provides comprehensive metadata for a single recorded take, suitable for data management use cases requiring full take-level detail beyond what `production:clip` provides. A temporal region carrying a `production:take` entity MAY also carry a `production:clip` entity; when both are present, the `production:clip` entity provides the editorial-facing summary and the `production:take` entity provides the complete production data management record.

Property	Type	Required	Description
takeId	string	REQUIRED	Unique take identifier, kebab-case.
rating	string (enum)	REQUIRED	Director's take rating: <code>circle</code> <code>hold</code> <code>false-start</code> <code>no-good</code> .
continuityNotes	string	OPTIONAL	Script supervisor's continuity notes for this take.
lensInfo	object	OPTIONAL	Lens information object with sub-properties: <code>focalLength</code> (integer, mm), <code>aperture</code> (string, e.g., "T2.8"), <code>focus</code> (string, e.g., "4.5m").
isoValue	integer	OPTIONAL	Camera ISO setting for this take.
whiteBalance	integer	OPTIONAL	Camera white balance setting in Kelvin.
cameraHeight	string (enum)	OPTIONAL	Camera height: <code>ground</code> <code>low</code> <code>standard</code> <code>high</code> <code>overhead</code> <code>aerial</code> .

Property	Type	Required	Description
cameraMovement	string (enum)	OPTIONAL	Camera movement type: <code>static</code> <code>handheld</code> <code>dolly</code> <code>crane</code> <code>steadicam</code> <code>drone</code> <code>other</code> .

EXAMPLE — production:take linked entity instance (INFORMATIVE)

```
{
  "entityType": "production:take",
  "takeId": "take-42a-003",
  "rating": "circle",
  "continuityNotes": "Actor enters from camera left. Costume: grey jacket, no hat.",
  "lensInfo": {
    "focalLength": 50,
    "aperture": "T2.8",
    "focus": "4.5m"
  },
  "isoValue": 800,
  "whiteBalance": 5600,
  "cameraHeight": "standard",
  "cameraMovement": "steadicam"
}
```

§5.4 production:vfx

Type identifier: `production:vfx`

Purpose: Annotates a region of a master file that contains visual effects plate content, providing tracking and vendor coordination metadata. This linked entity type supports VFX supervisor workflows and enables downstream VFX facility teams to discover plate characteristics directly from the ADAC container without consulting external databases.

Property	Type	Required	Description
vfxId	string	REQUIRED	Unique VFX region identifier, kebab-case.
shotCode	string	REQUIRED	The VFX shot identifier from the VFX supervisor's breakdown (e.g., "VFX_042_010").
plateType	string (enum)	REQUIRED	Plate type: <code>clean-plate</code> <code>tracking-plate</code> <code>bg-plate</code> <code>fg-plate</code> <code>matte-painting</code> <code>other</code> .
trackingMarkers	boolean	REQUIRED	<code>true</code> if tracking markers are present in the plate region; <code>false</code> otherwise.

Property	Type	Required	Description
vendorName	string	OPTIONAL	The name of the VFX facility assigned to this shot.
chromaKeyColor	string (enum)	OPTIONAL	Chroma key color: green blue none other.
deliveryDueDate	string	OPTIONAL	ISO 8601 date by which the VFX composite is due (e.g., "2026-09-01").
vfxNotes	string	OPTIONAL	Free-text notes from the VFX supervisor for the receiving facility.

EXAMPLE — production:vfx linked entity instance (INFORMATIVE)

```
{
  "entityType": "production:vfx",
  "vfxId": "vfx-042-bg",
  "shotCode": "VFX_042_010",
  "plateType": "bg-plate",
  "trackingMarkers": true,
  "vendorName": "Horizon Visual Effects",
  "chromaKeyColor": "green",
  "deliveryDueDate": "2026-09-01",
  "vfxNotes": "12 tracking markers. Please preserve full sensor aperture - do not crop."
}
```

§5.5 production:audio

Type identifier: production:audio

Purpose: Annotates a temporal region of an audio master file with track assignment and synchronization metadata. This linked entity type supports location sound workflows, multi-track audio management, and picture-to-audio sync verification in post-production.

Property	Type	Required	Description
audioId	string	REQUIRED	Unique audio region identifier, kebab-case.
trackAssignment	string (enum)	REQUIRED	Track purpose: dialogue production-sound wild-sound ambience music foley sound-effects mix-other.
channelCount	integer	REQUIRED	Number of audio channels in this region.
sampleRate	integer	REQUIRED	Sample rate in Hz. MUST be one of 48000 or 96000.

Property	Type	Required	Description
bitDepth	integer	REQUIRED	Bit depth. MUST be one of 24 or 32.
timecodeSync	string	REQUIRED	SMPTE timecode at which this audio region begins (see §6.2).
syncSource	string (enum)	REQUIRED	Synchronization method: <code>timecode</code> <code>clap</code> <code>manual</code> .
recordistName	string	OPTIONAL	Name of the production sound recordist.
boomOperatorName	string	OPTIONAL	Name of the boom operator.
microphoneType	string	OPTIONAL	Microphone type or model used for this recording (e.g., "Schoeps MK41").

EXAMPLE — production:audio linked entity instance (INFORMATIVE)

```
{
  "entityType": "production:audio",
  "audioId": "audio-42a-003-boom",
  "trackAssignment": "dialogue",
  "channelCount": 2,
  "sampleRate": 48000,
  "bitDepth": 24,
  "timecodeSync": "10:22:14:00",
  "syncSource": "timecode",
  "recordistName": "M. Okonkwo",
  "boomOperatorName": "T. Johansson",
  "microphoneType": "Schoeps MK41"
}
```

§6 Edit Pipeline Extensions

§6.1 Production Edit Operations

The ADAC Core edit pipeline (stored in the `edits/` directory) stores non-destructive JSON operation sequences that describe editorial transformations without modifying master files; see ADAC Core §4.5 for the base edit pipeline model. This section defines production-specific operation types that extend the core set.

Each production edit operation object MUST include an `operationType` property (string) identifying the operation type. The following production-specific `operationType` values are defined by this profile:

operationType	Description	Key Parameters
<code>clip-select</code>	Selects a clip region from a master file for inclusion in an editorial assembly. The referenced clip MUST be identified by a <code>clipId</code> matching a <code>production:clip</code> linked entity in the <code>regions/</code> directory.	<code>clipId</code> (string, REQUIRED); <code>assemblyPosition</code> (integer, OPTIONAL)
<code>clip-dissolve</code>	Applies a dissolve transition between two consecutive clip regions in an assembly. The transition is described non-destructively; it does not modify either clip's master file.	<code>fromClipId</code> (string, REQUIRED); <code>toClipId</code> (string, REQUIRED); <code>durationFrames</code> (integer, REQUIRED)
<code>color-grade-lut</code>	Applies a look-up table (LUT) color grade to a clip region. The <code>lutPath</code> MUST reference a file present in the <code>derivatives/</code> directory of the container. This operation does NOT modify the master file.	<code>clipId</code> (string, REQUIRED); <code>lutPath</code> (string, REQUIRED)
<code>audio-sync</code>	Synchronizes an audio region to a video region by a specified timecode offset. This operation records the sync relationship non-destructively and does not re-encode either the video or audio master file.	<code>videoClipId</code> (string, REQUIRED); <code>audioId</code> (string, REQUIRED); <code>offsetFrames</code> (integer, REQUIRED, may be negative)
<code>retime</code>	Records a speed ramping operation applied to a clip region (e.g., slow motion, undercrank). The speed factor is applied editorially; it does not modify the master file.	<code>clipId</code> (string, REQUIRED); <code>speedFactor</code> (number, REQUIRED, e.g., 0.5 for 50% speed); <code>outputFrameRate</code> (string, OPTIONAL)

§6.2 Timecode Conventions

All timecode values in `production:clip`, `production:slate`, `production:take`, and `production:audio` linked entities MUST use the SMPTE timecode format as defined in [SMPTE-ST12].

Non-drop-frame timecode MUST use the colon as the frame-count separator: `HH:MM:SS:FF`.

Drop-frame timecode MUST be distinguished from non-drop-frame timecode by appending a semicolon as the frame separator: `HH:MM:SS;FF`.

Implementations that do not support drop-frame timecode MUST reject containers whose `timecodeIn` or `timecodeOut` values use the semicolon separator and MUST surface a descriptive error to the calling application.

NOTE — Drop-frame timecode (HH:MM:SS;FF) is standard for U.S. broadcast productions operating at 29.97 fps. Non-drop-frame timecode (HH:MM:SS:FF) is standard for cinema productions at 24 fps, European broadcast at 25 fps, and integer-rate productions at 30 fps. Implementations targeting both markets MUST be capable of processing both formats.

§6.3 EDL Interoperability

Edit decision list files stored in the `edits/` directory as ADAC edit pipeline files MAY carry a companion EDL export in CMX 3600 format [CMX3600] as a derivative file in the `derivatives/` directory. The CMX 3600 derivative file MUST be listed in the container's manifest. The edit pipeline file MUST reference the CMX 3600 derivative via an `edlExportRef` property whose value is the manifest path of the derivative file.

NOTE — The CMX 3600 export is a derivative file provided for interoperability with legacy editorial systems. It is not the normative editorial record. The normative editorial record is the ADAC JSON edit pipeline file. In the event of any discrepancy between the JSON edit pipeline and the CMX 3600 export, the JSON edit pipeline governs.

§7 Region Annotation Extensions

§7.1 Image Regions (Film Frames, Slates)

Production image regions annotate still frames extracted from a master file, including slate frames and reference frames. Image regions MUST use pixel-rectangle coordinates expressed as the following four integer properties:

- `x` (integer): The pixel coordinate of the left edge of the region.
- `y` (integer): The pixel coordinate of the top edge of the region.
- `width` (integer): The width of the region in pixels.
- `height` (integer): The height of the region in pixels.

A slate image region MUST be annotated with a `production:slate` linked entity (see §5.2). A VFX plate image region that contains tracking markers SHOULD be annotated with a `production:vfx` linked entity (see §5.4) in addition to any other applicable linked entities.

§7.2 Temporal Regions (Audio/Video Clips)

Production temporal regions annotate contiguous time ranges within audio or video master files. Temporal regions MUST use SMPTE timecode range coordinates expressed as the following two string properties:

- `timecodeIn` (string): SMPTE timecode of the first frame of the region (see §6.2 for format).
- `timecodeOut` (string): SMPTE timecode of the last frame of the region.

A video clip temporal region MUST be annotated with at least one `production:clip` linked entity. A video clip temporal region MAY additionally carry a `production:take` linked entity when full take-level detail is required.

An audio temporal region MUST be annotated with a `production:audio` linked entity.

§7.3 Multi-Track Regions (Audio Channels)

When a master file contains multiple audio tracks — such as multi-track location recordings or multi-channel production audio — each track constitutes a distinct temporal region in the `regions/` directory of the container. A separate region file MUST be created for each audio track.

Each region's `production:audio` linked entity MUST specify its `channelCount` and `trackAssignment` properties. Multi-track regions representing the same timecode range SHOULD share the same `timecodeIn` and `timecodeOut` values. Multi-track regions representing recordings made simultaneously SHOULD reference a common `syncSource` value to indicate that they were recorded in sync.

NOTE — Multi-track audio regions are modeled as sibling regions within the same container, not as nested sub-regions. This is consistent with the ADAC Core region model, which does not support nested regions; see ADAC Core §5.3 for the flat region structure requirement.

§8 Interoperability

§8.1 Interaction with ADAC Core

The production profile uses the following ADAC Core infrastructure components without modification:

Core Component	Production Profile Usage
<code>manifest.json</code>	Lists all files in the container including <code>metadata/profiles/production.json</code> , all region annotation files, edit pipeline files, and CMX 3600 derivative exports.
<code>metadata/core.json</code>	Carries ADAC Core descriptive metadata for the container. The production profile does NOT modify this file.
<code>master/</code>	Stores raw acquisition master files (camera originals, audio masters). Master files MUST NOT be modified by any production profile operation.
<code>derivatives/</code>	Stores dailies, access copies, color-graded derivatives, LUT files referenced by <code>color-grade-lut</code> operations, and CMX 3600 EDL exports.
<code>edits/</code>	Stores ADAC JSON edit pipeline files for assembly cuts, rough cuts, and final cuts. Each edit pipeline file carries production-specific operation types defined in §6.1.
<code>regions/</code>	Stores region annotation files carrying <code>production:clip</code> , <code>production:slate</code> , <code>production:take</code> , <code>production:vfx</code> , and <code>production:audio</code> linked entities.
<code>provenance/log.json</code>	Records department handoff events as provenance log entries. Each handoff changes the <code>currentDepartment</code> value in the production profile metadata file and appends a corresponding entry to the provenance log.
<code>checksums.json</code>	Provides fixity values for all files in the container. The production profile metadata file and all region annotation files MUST be listed in <code>checksums.json</code> .
<code>signatures/</code>	Carries signed archival attestation of custody events, including department handoffs.

§8.2 Coexistence with Other Profiles

The production profile MAY coexist with the ADAC-Preservation profile [ADAC-PRESERVATION] in a single container. When both profiles are present, the preservation profile provides institutional archival description; the production profile provides production operational metadata. The two profiles operate in separate namespaces and MUST NOT define conflicting values for the same container property.

The `production:` namespace MUST NOT be used by any profile other than this specification. Implementations that encounter `production:-`namespaced linked entities in a container that does not declare ADAC-Production conformance SHOULD treat those entities as unrecognized and MAY surface a warning to the user.

§8.3 Standards Alignment

The following external standards are relevant to ADAC-Production implementations:

Standard	Relationship to This Profile
SMPTE ST 12-1:2014 [SMPTE-ST12]	Normative reference for timecode format (see §6.2). All timecode values in this profile MUST conform to SMPTE ST 12-1.
SMPTE ST 2065 (ACES)	The <code>aces</code> value in the <code>colorSpace</code> enum (see §4.6) refers to the Academy Color Encoding System as defined in SMPTE ST 2065.
SMPTE ST 2067 (IMF) [SMPTE-IMF]	IMF packages are a delivery output format produced from an ADAC-Production container. The <code>imf</code> value in <code>deliveryFormat</code> references this standard.
AAF [AAF]	Advanced Authoring Format is an interchange reference. AAF files may be stored as derivatives or edit pipeline companions within the container.
MXF [MXF]	Material Exchange Format is referenced by the <code>broadcast-mxf</code> delivery format enum value. MXF containers may carry master files for certain production formats.

NOTE — ADAC-Production is an archival metadata profile, not a delivery format. IMF and DCP packages are delivery outputs produced from an ADAC-Production container; they are not replacements for it. The ADAC container preserves the complete provenance, fixity record, and production metadata of the assets from which delivery packages are derived.

§8.4 Round-Trip Fidelity

Implementations that open and re-save an ADAC-Production conformant container MUST preserve all `production:` linked entities present in the container's region annotation files. Implementations MUST preserve all edit pipeline operation objects defined in §6.1. Implementations MUST preserve all region annotation files they do not explicitly modify. An implementation MUST NOT remove or alter any `production:` linked entity unless the user has explicitly directed that modification.

§8.5 Graceful Ignorance

A reader that does not support the ADAC-Production profile MUST be able to open and process any ADAC container carrying this profile by ignoring the `metadata/profiles/production.json` profile file and all `production:`-namespaced linked entities, without failure. The presence of the production profile MUST NOT cause a non-production-aware ADAC Core reader to fail, emit an error, or refuse to open the container.

§9 Complete Profile Example

This section presents a complete, multi-department handoff scenario to illustrate how all components of the ADAC-Production profile work together in practice. All content in this section is informative.

§9.1 Scenario Description

A digital image technician (DIT) wraps a raw 4K digital cinema camera master file from Scene 42A, Take 3 of the feature film *The Last Meridian* into an ADAC container. The container is fully annotated with production profile metadata and region annotations, and then handed off to the editorial department. The container directory structure is as follows:

```
the-last-meridian_s42a_t3_a014.adac/  
  manifest.json  
  checksums.json  
  metadata/  
    core.json  
    profiles/  
      production.json  
  master/  
    A014C003_2604150K_CANON.mxf  
  derivatives/  
    A014C003_2604150K_PROXY.mp4  
  regions/  
    region-slate-01.json  
    region-clip-001.json  
  provenance/  
    log.json
```

§9.2 Example 1 — production.json Profile Metadata File

EXAMPLE — production.json (INFORMATIVE)

```
{  
  "profileName": "ADAC-Production",  
  "profileVersion": "0.9.0-draft.1",  
  "schemaRef": "https://adac-standard.com/schemas/production/production-  
profile.schema.json",  
  "project": {  
    "productionId": "f47ac10b-58cc-4372-a567-0e02b2c3d479",  
    "productionTitle": "The Last Meridian",  
    "productionType": "feature-film",  
    "productionCompany": "Threshold Pictures, LLC",
```

```

    "shootingFormat": "digital-cinema",
    "directorName": "A. Vasquez",
    "cinematographerName": "C. Park"
  },
  "assetClassification": {
    "assetCategory": "raw-footage",
    "cameraRoll": "A014",
    "sceneNumber": "42A",
    "takeNumber": "3",
    "cameraDesignation": "A",
    "timecodeIn": "10:22:14:00",
    "timecodeOut": "10:23:08:12"
  },
  "departmentStatus": {
    "currentDepartment": "editorial",
    "approvedForNextDepartment": true,
    "approvalDate": "2026-04-15T19:30:00Z",
    "approvedBy": "R. Nakamura (DIT)"
  },
  "departmentNotes": "Circled take. Camera slightly soft at head - usable
from 10:22:14:12. Proxy included.",
  "deliverySpec": {
    "deliveryFormat": "imf",
    "deliveryFrameRate": "24",
    "deliveryResolution": "4k-dci",
    "colorSpace": "dci-p3",
    "dynamicRange": "sdr"
  }
}

```

§9.3 Example 2 — Region Annotation Files (Clip and Slate)

EXAMPLE — region-clip-001.json (INFORMATIVE)

```

{
  "regionId": "region-clip-001",
  "masterFile": "master/A014C003_2604150K_CANON.mxf",
  "regionType": "temporal",
  "timecodeIn": "10:22:14:00",
  "timecodeOut": "10:23:08:12",
  "linkedEntities": [
    {
      "entityType": "production:clip",
      "clipId": "clip-001",
      "sceneNumber": "42A",
      "takeNumber": "3",
      "cameraDesignation": "A",
    }
  ]
}

```

```

    "timecodeIn": "10:22:14:00",
    "timecodeOut": "10:23:08:12",
    "circled": true,
    "transcriptionNotes": "Strong performance. Pick up from 10:22:14:12
past head softness.",
    "colorTemperature": 5600
  },
  {
    "entityType": "production:take",
    "takeId": "take-42a-003",
    "rating": "circle",
    "continuityNotes": "Actor enters camera left. Grey jacket, no hat.",
    "lensInfo": {
      "focalLength": 50,
      "aperture": "T2.8",
      "focus": "4.5m"
    },
    "isoValue": 800,
    "whiteBalance": 5600,
    "cameraHeight": "standard",
    "cameraMovement": "steadicam"
  }
]
}

```

EXAMPLE — region-slate-01.json (INFORMATIVE)

```

{
  "regionId": "region-slate-01",
  "masterFile": "master/A014C003_2604150K_CANON.mxf",
  "regionType": "image",
  "frameTimecode": "10:22:13:22",
  "coordinates": {
    "x": 120,
    "y": 80,
    "width": 640,
    "height": 360
  },
  "linkedEntities": [
    {
      "entityType": "production:slate",
      "slateId": "region-slate-01",
      "productionTitle": "The Last Meridian",
      "directorName": "A. Vasquez",
      "cinematographerName": "C. Park",
      "sceneNumber": "42A",
      "takeNumber": "3",
      "cameraDesignation": "A",
    }
  ]
}

```

```
    "roll": "A014",
    "date": "2026-04-15",
    "syncType": "electronic"
  }
]
}
```

§9.4 Example 3 — Provenance Log Entry (Department Handoff)

EXAMPLE — provenance/log.json entry recording the camera-to-editorial handoff (INFORMATIVE)

```
{
  "events": [
    {
      "eventId": "evt-2026-04-15-001",
      "eventType": "department-handoff",
      "eventDateTime": "2026-04-15T19:30:00Z",
      "agent": {
        "name": "R. Nakamura",
        "role": "Digital Image Technician",
        "organization": "Threshold Pictures, LLC"
      },
      "fromDepartment": "acquisition",
      "toDepartment": "editorial",
      "containerFixityVerified": true,
      "fixityAlgorithm": "SHA-256",
      "note": "Handoff of Scene 42A Take 3 camera original and proxy. Circled
take approved by director A. Vasquez on-set."
    }
  ]
}
```

§10 References

§10.1 Normative References

Tag	Reference
[ADAC-CORE]	ADAC 1.0 — Archival Digital Asset Container Format Specification. Version 1.0, 2026. InnoVadens, LLC.

Tag	Reference
[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997. Internet Engineering Task Force.
[RFC8174]	Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words," BCP 14, RFC 8174, May 2017. Internet Engineering Task Force.
[JSON-SCHEMA]	Wright, A., Andrews, H., Hutton, B., "JSON Schema: A Media Type for Describing JSON Documents," JSON Schema Draft 2020-12. Internet Engineering Task Force.
[ISO8601]	International Organization for Standardization, "Date and time — Representations for information interchange," ISO 8601:2019.
[SMPTE-ST12]	Society of Motion Picture and Television Engineers, "SMPTE ST 12-1:2014 — Television — Time and Control Code," 2014.

§10.2 Informative References

Tag	Reference
[ADAC-STYLE]	ADAC Specification Style Guide. Version 1.0, 2026. InnoVadens, LLC.
[ADAC-PRESERVATION]	ADAC-Preservation 1.0 — Preservation Profile Specification. Version 1.0 (Draft), 2026. InnoVadens, LLC.
[ZIP-APPNOTE]	PKWARE, Inc., "ZIP File Format Specification," APPNOTE.TXT, Version 6.3.10.
[SMPTE-IMF]	SMPTE ST 2067 — Interoperable Master Format (IMF). Society of Motion Picture and Television Engineers.
[CMX3600]	CMX Corporation, "CMX Edit Decision List Format Specification," CMX 3600, 1987.
[AAF]	Advanced Authoring Format Association, "Advanced Authoring Format (AAF) Object Specification," Version 1.1.
[MXF]	International Organization for Standardization, "Material Exchange Format (MXF) — File Format Specification," ISO/IEC 13818-1:2019 / SMPTE ST 377.

About the Editor

ADAC is edited and maintained by **John Vaden**, a genealogist and technical workflow contributor with decades of experience in genealogical research and in shaping data practices and process standards across analyst and software development teams. His work on ADAC emphasizes clarity,

interoperability, and practical implementation for genealogists, developers, and archival practitioners. InnoVadens, LLC serves as the legal steward of the ADAC Standard.

Document History

- v1.0 — Initial release by Editor (John Vaden); AI-assisted drafting and refinement