EIDR

Obtaining an EIDR ID for a DECE CFF Container (DCC) APID

7-May-2012

Document Version 0.94

Preliminary

TABLE OF CONTENTS

1   Overview

This document describes how to get an EIDR ID to use as the APID of a DCC (DECE Common File Format Container). It assumes familiarity with the DECE *Identifiers, Content Structure, Metadata and Bundles Technical Note,[1]* especially Section 5, and with the DECE *Content Metadata Specification.[2]*

Some less common use cases that are allowed or envisioned by the DECE specifications do not currently have EIDR practices defined for them. These are listed at the end of the document in the Open Issues section.

NOTE: This is a preliminary document. Once it is final, the contents will be merged into the technical note referenced above*.*

2   Process

EIDR Encodings

EIDR Encodings are associated with the digital form of an asset, such as a DCC. As described in the DECE technical note mentioned above, EIDR records form a hierarchy of Titles, Editions and Encodings. The Encoding for the DCC is registered below the EIDR Edit used to create the ALID with which the DCC is associated.

The EIDR Edit records for the required ALIDs must have been previously registered in accordance with EIDR best practices and the *Identifiers, Content Structure, Metadata and Bundles Technical Note*.

The metadata needed to register an EIDR Encoding comes from the following:

- The Container metadata as defined in the DECE *Content Metadata Specification*. Specific field usage for EIDR registrations is defined below in the section on Container Metadata and Uniqueness in EIDR.
- The EIDR ID for the record to which the Encoding should be parented. Typically, this is the EIDR Edit used to create the ALID.

---

[1] Available at http://uvvuwiki.com/index.php?title=IDs,_Content_Structure,_Metadata_and_Bundles. For those without access to uvvuwiki.com, the information in this section is also covered in the *EIDR/UV Technical Note*, which can be found with the EIDR technical documentation at http://eidr.org/resources.

[2] Available with the public DECE technical specifications at http://uvvu.com/uv-for-business.php.

Registration Process

Registrants will provide EIDR with a set of XML files each containing valid Container Metadata and a single text-based manifest file. After the EIDR Encoding registrations are completed, the manifest will be returned with the APIDs for the registered Encodings added. The formats of these files are described below.

The expected turnaround time on these registrations will be two business days once the process is mature, within 2-3 weeks of the first registrations. This turnaround time does not include any correction of errors in the submission. If there are errors in the submission, including registrations that are not unique as specified in this document, APIDs will be returned only for the successful registrations. The remainder will need to be corrected and resubmitted. Registrants are strongly encouraged to submit test records representing all of their use cases and workflows well in advance of production registration requests so that any systematic problems can be resolved early.

 Additional, more direct, self-service mechanisms with immediate turnaround will be available from EIDR within 2 months.[3]

DCC Metadata File (One per APID)

As the EIDR Encoding record for an APID is constructed from DCC Container metadata, it is necessary to provide the Container metadata in a file. Each file contains a `MetadataMovie` element as defined in the DECE *Content Metadata Specification*, Section 4.

Each file must be valid Container XML, with all required fields present. Additionally, some optional fields may be required to differentiate the records in EIDR. EIDR requires that Encodings belonging to the same Edit have differentiating metadata. Thus is essential for the successful registration of the DCC's Encodings that one or more of the fields listed in the section below on Container Metadata and Uniqueness in EIDR be specified and different among Encoding registrations under the same Edit.

The XML for each DCC must be in a file named <TitleEtc>_<ProfileEtc>_<ID>.xml, where

- <ID> is the short EIDR ID from the ALID. That is, it's the portion of the ID following 'urn:dece:alid:eidr-s:'.
- <TitleEtc> is an alphanumeric title string the contents and format of which are left to the provider. For episodic content it can contain series, season and episode names and/or numbers.
- <ProfileEtc> is the profile "SD" or "HD". Additional version or variant information can be encoded after the profile as needed, e.g., "SDv2".

---

[3] An upgrade to the EIDR registry and data model tentatively planned for Q4 2012 will allow registrants to use Container metadata directly via EIDR's web UI or web services APIs to register the necessary EIDR Encodings. In the meantime, EIDR can provide a registration tool that works with the XML files described in this document.

The filenames should only use alphanumeric characters plus hyphen (ASCII 45), underscore (ASCII 95) and period (ASCII 46).

For example, for a request to register two different DCC encodings of Top Gun with an ALID of urn:dece:alid:eidr-s:0EFB-02CD-126E-8092-1E49-W, the two files could be named:

> TopGun_SD_0EFB-02CD-126E-8092-1E49-W.xml and

> TopGun_HD_0EFB-02CD-126E-8092-1E49-W.xml,

each containing a single `MetadataMovie` element.

The XML metadata files must be accompanied by a text-based manifest. Each line of the manifest contains the following tab-separated fields:

- Filename: The name of the DCC metadata file, as defined above
- EIDR Parent ID: The full EIDR ID of the record to which this Encoding should be parented. Typically, this is the ID of the Edit used in the ALID and is the same as the EIDR in the filename, but it can be different as outlined in the mezzanine file use case below in Common Practices. If it is the same, the string "SAME" may be used here instead of the EIDR ID.

The manifest file is named using the following convention: manifest<-suffix>.txt. <–suffix> is optional, and can be used by registrants to relate the APID requests to their internal processes.

The manifest and all the XML files must be in the same directory inside a zip file.

Care should be taken to ensure that:

- The manifest file is valid and in the format specified above.
- The filenames for the DCC metadata files conform to the practice outlined above.
- The Container metadata is well-formed XML and validates against the schema.
- The EIDR ID for each parent is an existing record in EIDR as described in Section 3 below.
- The Container metadata for the registration differs from that of any other registered Encoding under that same parent as described in Section 4 below.

Output from registration process

After registration, a response file will be returned with each line containing these tab-separated fields:

- Filename: As defined above
- EIDR Parent ID: If supplied as input or "SAME"
- Result: Either the resulting APID (in urn:dece:alid:eidr-s format) or "ERROR" .
- If an error, one of the following error codes:
  - Bad DCC XML
  - Bad parent
  - Duplicate
  - Other error

- If an error, optionally additional error details, such as the invalid element or the EIDR of the duplicate.

3   Common Practices

APIDs and ALIDs

In the most common case, an ALID is associated with two Containers (SD and HD), and would therefore require two APIDs.

DECE is designed to support many-to-many mappings for certain outlying use cases, although this document does not yet describe those cases. (See Open Issues below.) It is important to note that just because it's possible does not mean it's a good idea. Keep to the simple case if at all possible.

The ALID corresponds either to the EIDR Edit of an Episode record or to the Edit of a root Movie or TV record. In the latter case, that root record could also be a Composite of other EIDR records.

The following sections instruct how to structure ALIDs for different content structures.

Non-episodic

The following diagram illustrates how and HD and SD Container APIDs map to Edit ALIDs:
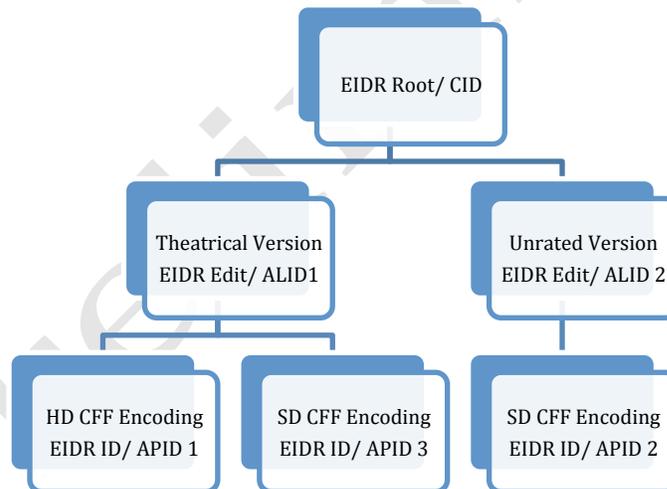
The parent is an essential element of identification. Note that even if APID2 and APID1 are in the same format and are identical in all the required metadata fields, they will still be considered different EIDR objects because they have different parents.

Episodic

For episodic content, the model at the APID level is identical to the simple case – the presence of a more complex hierarchy above it does not affect the Encoding.
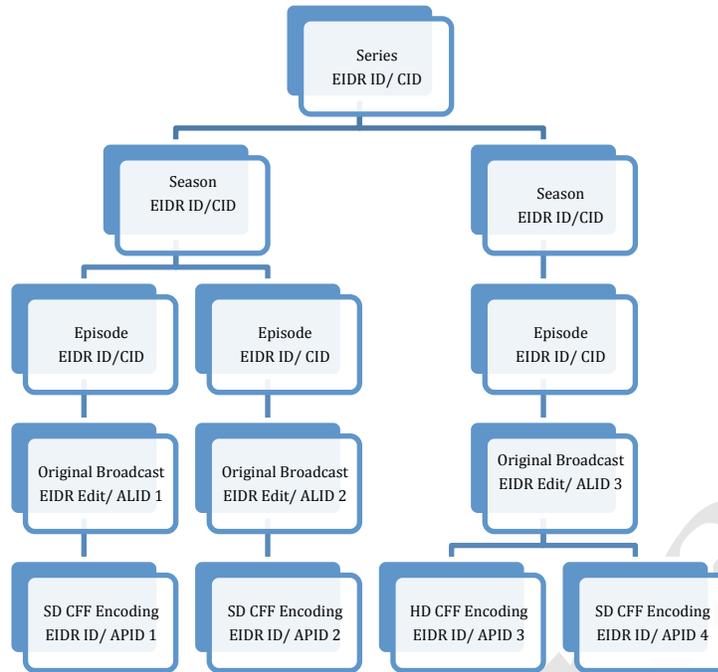
Composites

Composites are likely rare in DECE, but included here for completeness. One use case is the registration of a feature length version of a short mini-series.

Composites behave very much like other root records in EIDR, but have relationships to indicate the content from which they were derived. For purposes of ALIDs and APIDs these additional relationships are not important. The structure from the Composite root record through the Edit record down to the Encoding record is identical to the other cases.

If all three items were being distributed in DECE, there would be three complete hierarchies, one for each of the Shorts and one for a feature-length composite of the two. The diagram below shows this. Each of the three viewed separately is just like the simple case given above.

Note that the Composite is built of the two short Edits, not from their abstract roots. .
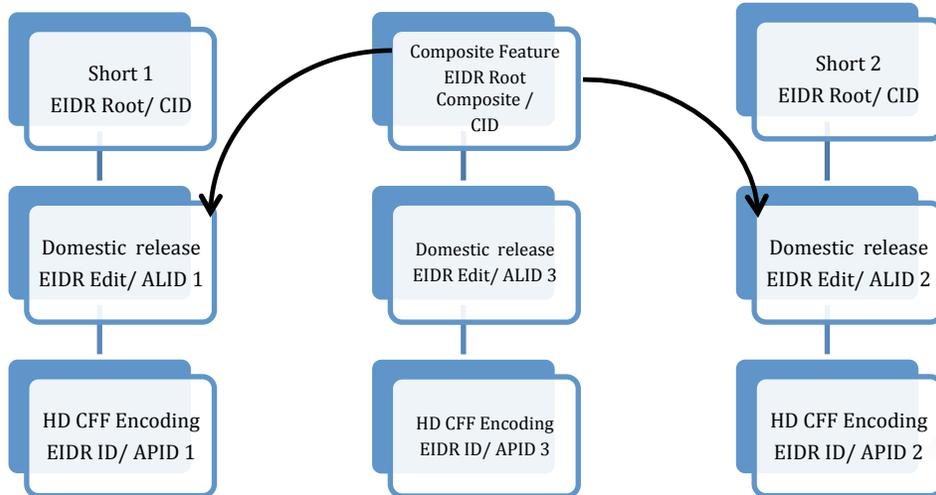
4 Container Metadata and Uniqueness in EIDR

This section defines what Container metadata is used in EIDR Encoding metadata. As discussed above, EIDR Encoding records must have metadata that distinguishes any two records related to the same Edit. Failure by a registrant to provide different metadata with an Encoding registration under the same Edit as an existing one will cause the attempted registration to be rejected with an error of "Duplicate".

- EIDR uses the following Container metadata fields to differentiate Encodings:
    - o /MetadataMovie/ContentMetadata/DECEMediaProfile
    - o Elements and sub-elements of /MetadataMovie/TrackMetadata/Track/Video
        - ▪ Picture/AspectRatio
        - ▪ SubtitleLanguage (if present)
        - ▪ SubtitleLanguage@closed (if present)
        - ▪ Notes:
            - • ColorType: The Encoding must use the ColorType of the parent EIDR Edit (DECE ALID). An Encoding with a new ColorType requires a new Edit, not just a new Encoding.
            - • Required fields NOT used for differentiation
                - o Encoding/Codec – always 'H.264, MPEG-4 Part 10'
                - o Encoding/CodecType – always 'IANA:h264'
    - o Elements of /MetadataMovie/TrackMetadata/Track/Audio:
        - ▪ Channels, Type, Language, Language@dubbed (if present)
        - ▪ Encoding/Codec
        - ▪ Notes

- Encoding/CodecType: If this field changes, Encoding/Codec must change as well so the two fields continue match, so it is only necessary to consider one of them.
- Required Fields NOT used for differentiation
  - Encoding/BitrateMax, Encoding/SampleRate, Encoding/SampleBitDepth
- Elements of /MetadataMovie/TrackMetadata/Track/Subtitle
  - Format, Language
  - Format@SDImage, Format@HDImage (if present)
  - Required Elements NOT used for differentiation
    - FormatType – always `'SMPTE 2052-1 Timed Text'`
- If present, /MetadataMovie/ContainerVersionReference
  - This must change if the container itself has changed but the metadata used for differentiation has not, including:
    - Changes to Audio/Encoding/BitrateMax, Audio/Encoding/SampleRate, Audio/Encoding/SampleBitDepth
    - Change to a Video Track
    - Addition of a video Track
    - Removal of a video Track
    - Any other re-encoding that does not change the metadata used for differentiation (for example adjustments in a VBR encoding.)
- If present, /MetadataMovie@MetadataVersionReference
  - This must change if the metadata in the container has changed but nothing else has.

For the current implementation, an Encoding differs from any other Encoding if any of the above fields differ. If a change is made in a field that is not in the above list, use a changed value of ContainerVersionReference or MetadataVersionReference.

5   Open Issues

This document covers the short-term needs of the DECE ecosystem and the currently stable use cases. The remainder of this document lists use cases of which we are aware but for which no practices have yet been defined. If you have others or if you intend to implement any of the use cases below in the next six months, please let us know.

One APID Fulfilling Multiple ALIDs

Note that the following represents a capability in DECE that is 'possible' but not recommended. We are including this for completeness.

DECE supports using multiple ALIDs to represent a single title to differentiate rights; for example assigning different ALIDs in different regions. One could fulfill all these ALIDs with a single Container, resulting in a mapping of multiple ALIDs to a single APID.

These ALIDs do not represent items that match current studio practices for EIDR Edits. In those practices, Edits represent versions of a title that differ in specific aspects, primarily video cut or music.

There are various solutions. Alternate forms of ALIDs could be constructed. Or EIDR could be extended to support IDs for distribution constructs such as Products, SKUs or UPCs that can differ even when referring to the same content item. The latter has been a topic of extensive discussion among EIDR's studio members.

If you are interested in using this scenario, please contact us to discuss it. Some solutions could have a long lead-time.

Future DECE Media Package Support

The DECE Media Package (DMP) does not currently require an identifier, although we expect this requirement to change. While a DCC only contains a single video program, the DMP can contain multiple DCCs, each with its on APID, along with other material. These could be trailers, value added material, other titles or interactive material all delivered in DCC form with their own ALIDs. Identification for DMPs using EIDR will be defined as necessary.

Next steps

EIDR will be publishing additional cookbooks on practices in areas such as franchises, bundles and movie series so that you'll be able to register these as needed in EIDR.

Appendix A: Alternate EIDR Practice - Mezzanine Files

Mezzanine files are not strictly within the scope of DECE, but some EIDR members have proposed practices to register them. This section is only relevant to those whose EIDR workflows involve registering encodings that derive from a mezzanine file as children of that mezzanine file.

In this case, the mezzanine Encoding must be registered with EIDR before the DCC Encoding since the EIDR ID of the parent mezzanine file must be provided in the DCC Encoding's registration fields. For that reason, the EIDR ID of the mezzanine must be supplied in the manifest (see above).