

BRISANET

Generic VOD Metadata driver description



brisanet

Table of Contents

1 Document History	3
2 Summary	4
3 Process.....	5
3.1 Ingestion modes.....	5
4 Metadata files.....	6
4.1 File naming convention	6
4.2 Media and trailer location.....	6
4.3 Media and trailer technology and DRM	7
4.3.1 Technology	7
4.3.2 DRM	8
4.4 assets.json	8
4.4.1 Example file.....	9
4.4.2 Shema.....	Error! Bookmark not defined.
4.4.3 Supported asset metadata.....	11
4.4.4 Supported media metadata.....	12
4.4.5 Notes.....	13
4.5 categories.json.....	13
4.5.1 Example file.....	14
4.5.2 Schema.....	Error! Bookmark not defined.
4.6 series.json.....	15
4.6.1 Example file.....	15
4.6.2 Schema.....	Error! Bookmark not defined.



brisanet

1 Document History

Version	Date	Author(s)	Summary
V1	2019-09-01	SCO	Initial version
V2	2020-12-21	SPH	Update Brisagnet
V3	2020-12-22	SCO	Update Subtitles Type and json tabulation



Today on the market there is multiple customers with custom demands for video ingest metadata and workflow.

For this purpose, we have a generic "product" driver whose format is strongly inspired from of the platform canonical metadata format, with a few simplifications aiming at making implementation simpler for the providers.

The "pseudo-canonical" data structures are received as JSON files. They are converted to a normal canonical and sent to the CMS.

The files are received in a single input folder per ingest, whose name must be the timestamp at which the process of delivery of the files started (something like `/input/1567497600/`). Such naming would allow the driver to be down for a while, and then ingest the folders in the right order. The structure of files inside this "per-ingest folder" depends on the ingestion mode - see below.

In both modes however, the files are moved to another one for archiving purpose after ingestion (e.g. to a folder like `/ingested/1567497600/`).

Also in both modes, images are provided in a subdirectory of the main ingest folder (`/input/1567497600/images/`), and removed after having been copied to the storage used by the CMS.

3.1 Ingestion modes

There are two ingestion modes: incremental and integral.

Incremental mode consists in receiving updates, "deltas", which are applied on top of the current data.

In incremental mode, metadata is put in two subdirectories: `add/` and `remove/`. Integral mode consists in receiving the entirety of the data at every ingest. This is like asking the provider to export the contents of a relational database.

In integral mode, files are put directly in the main ingest folder.

Whether a provider uses integral or incremental mode is configured at the EPG provider or VOD source catalog level.

4.1 File naming convention

For VOD ingests, the following files are received:

- assets.json
- categories.json
- series.json (optional)

For VOD ingests, videos can optionally be provided too. If they are provided, they must be put in a separate subdirectory whose name is the asset's id

(`/input/1567497600/asset1/`), and the video files themselves must start with the asset id, with an optional suffix, which can be (for instance) the bitrate, if multiple MBTS files are provided (eg `/input/1567497600/asset1/asset1_1600.ts`).

For VOD ingests, trailers can also be provided. The difference with "regular" videos is that trailers can be watched even by non-subscribed users. The rules for trailers video files are the same as for the main video content, except the subdirectory in which they must be uploaded is different: `/input/1567497600/asset1_trailers/`.

4.2 Media and trailer location

Both media and trailers have a "location" property which indicates "where" the corresponding video is.

As a reminder, an asset "media", of the platform nomenclature, is a single published multimedia file. A given asset can have multiple media, for instance because of the presence of burned-in subtitles in the video.

Note that if the platform manages video content (transcoding, DRM protection, publication on a CDN), each input media will be duplicated, as there is a published media per streaming protocol. In other words: for a given MP4 file provided as input, the output will be (for instance) one "HLS media", one "DASH media", and one "HSS media". If platform does not manage video workflow, and instead the provider does, then the metadata must provide (at least) one media per supported streaming protocol, and in that case, the location is the URL of the corresponding manifest or playlist file.

The following table lists the possible use-cases of the driver, in terms of media management:

Use-case	Media location format	Example	Description
video content is not managed by platform	Absolute HTTP URL, generally of the manifest / playlist file	http://cdn.provider.com/asset.m3u8	The provider has its own video distribution infrastructure. The location provided is the origin URL, sent to the end-user as-is or processed (e.g. tokenized) by another driver at asset playback time.
MBTR files are provided for each media	An absolute or relative path on the storage used, with a wildcard to match all video files corresponding to a media	/media/VOD/asset1/asset1_*.ts	For each media, multiple video files are provided - one per bitrate. In this case, there must be a single entry in the list of media for all of the MTBR files; therefore, a wildcard must be used to match them all.
A single video file is provided for each media	An absolute or relative path on the storage used, pointing to the single video file	/media/VOD/asset1/asset1_vostfr.ts	For each media, a single video file is provided. In this case, there is simply a single path toward the video file. Remote URLs can be used (toward an FTP server, S3 bucket, etc.), with the exception of HTTP URLs , which are reserved to the first use-case.

Note that we do not support video content being cut in multiple parts (or "chapters"). No matter which use-case above is used, each video file must contain the entirety of the asset.

4.3 Media and trailer technology and DRM

Both media and trailers have two optional properties: "technology" and "drm".

4.3.1 Technology

The `technology` property indicates "what" the corresponding video is:

- If platform manages video content, the technology provided must be the multimedia container format provided ("MP4", "TS" for MPEG2-TS, etc.).

- If platform does not manage video content, then the media provided is a "published" origin URL for the content. In this case, the technology is the streaming protocol available at that URL (e.g. "HLS", "HSS", "DASH").

If this property is absent, the technology is inferred from the extension of the provided files:

Provided file's extension	Inferred technology
.mp4	mp4
.ts	ts (MPEG-2 TS)
.m3u8	HLS
.mpd	DASH
.ism	HSS
any other	not inferred yet; please use the technology field.

For convenience, this property is case-insensitive.

4.3.2 DRM

This property is only used in the case platform does not manage video content, which means the media provided is a published origin URL for the content. In this case, the URL is that of a manifest, for an asset which might be DRM-protected. The DRM protocol used for this asset can then be provided in the `drm` field; the ID of the DRM resource must also be provided in this case, in the media metadata `drm_resource_id` (see section "Supported media metadata" below). If no DRM resource ID is provided and no `drm` property either, the asset is considered not DRM-protected.

If no DRM resource ID is provided and a `drm` property is provided, the DRM resource id is considered to be the asset ID. Relying on this behavior is not recommended. If a DRM resource ID is provided and no `drm` field is provided, the protocol is inferred from the technology, according to the following mapping:

Media technology	Inferred DRM
DASH	FairPlay
HLS	Widevine
HSS	PlayReady

4.4 assets.json

This file contains the assets, in the platform sense of the term.

4.4.1 Example file

```
{
  "version": "1.1",
  "assets": [
    {
      "id": "e5dd7acd-c432-45d6-8d3c-6508bbb61a22", // mandatory string (idAssetProvider)
      "title": { // mandatory
        "eng": "lorem ipsum ..."
      },
      "subtitle": { // optional (AD_SUBTITLE)
        "eng": "lorem ipsum ..."
      },
      "synopsis": { // mandatory
        "eng": "lorem ipsum ..."
      },
      "morality_level": 18, // mandatory
      "adult": false, // mandatory
      "duration": 3600000, // mandatory, in ms
      "series_id": "523c4c5c-68c5-4e00-a98b-754b9e9f4b71", // Optional. Empty string is also OK, if not a series
      "season_id": "759bab6c-1168-4ed5-ab91-b4ec41501daa", // mandatory if series_id provided. Empty string OK for non-series content.
      "season_episode_number": 7, // mandatory if series_id provided. Episode number in the season
      "genres": [ // mandatory (can be empty if no genres)
        "horror",
        "love",
        "college"
      ],
      "available_from": 1566047956123, // mandatory. UTC timestamp, in ms
      "available_to": 1666047956456, // mandatory. UTC timestamp, in ms
      "metadata": {}, // mandatory (but can be empty). Arbitrary key/value pairs. Keys are strings. Values can be text, numbers, booleans, or even structures such as lists of maps.
      "images": [ // mandatory (can be empty).
        {
          "id": "7b4ef709-51d8-438c-b2b1-4123dc5c6754", "type": "poster", // mandatory. Types to be agreed beforehand (usual ones are "poster" and "background")
        }
      ]
    }
  ]
}
```

```
"language": "eng", // optional, defaults to null. ISO 639-3
language code
// Mandatory
// the image location. Valid values are: // -
Absolute remote URLs:
// -
http(s)://(user:password@)server.com(:80)/a/b/c/d.png // - ftp(s)://(user:password@)my-ftp
server(:21)/vod/posters.123456.svg
// - sftp://(user:password@)my-ftp
server(:22)/vod/posters.123456.svg
// - s3://keyId:keySecret@aws-region/bucket
name/test.png
// - Relative local URLs:
// - images/123.jpeg
"location":
"https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Earth_by_the_EPIC_Team_on_21_April_2018.png/280px
Earth_by_the_EPIC_Team_on_21_April_2018.png"
}
],
"medias": [ // Mandatory; list of video files. {
  "id": "37409ea7-8a21-4899-9b18-fce68cc22f71", // Mandatory string
  "type": "TRAILER", // Mandatory. "TRAILER" for trailers, "MEDIA" for the actual movie / episode
  "title": "lorem ipsum...", // Mandatory string. If you don't have a title, you can repeat the id.

  // Mandatory
  // the media location. Valid values are: // -
Absolute remote URLs:
// -
http(s)://(user:password@)server.com(:80)/a/b/c/d.ts // - ftp(s)://(user:password@)my-ftp
server(:21)/vod/trailers.123456.mp4
// - sftp://(user:password@)my-ftp
server(:22)/vod/movies.123456.mkv
// - s3://keyId:keySecret@aws-region/bucket
name/test.ts
// - Relative local URLs:
// - medias/123.raw
//
// In case multiple files are to be considered (for instance multiple resolutions), you must use globs such
as: /vod/123456_*.mp4 to match files
/vod/123456_240.mp4, /vod/123456_360.mp4, /vod/123456_480.mp4,
and /vod/123456_720.mp4 "location": "https://server.com/movie.mp4", "technology": "mp4",
```

```

"audio_languages": [ "eng"], // Mandatory; list of audio
languages included (ISO-639-3 code)
"subtitles_languages": [ "fra" ], // Mandatory; list of cap-
tion languages included (ISO-639-3 code)
"subtitles": [
{
"language": "eng", // list of audio languages included
(ISO-639-3 code)
"location": "https://server.com/movie_str_eng.srt" }, {
"language": "fra",
"location": "https://server.com/movie_str_fra.srt" }, {
"language": "eng",
"location": "https://server.com/movie_str_fra.srt" }
],
"metadata": {} // Mandatory, but can be empty. Arbitrary
key/value pairs. Keys are strings. Values can be text, num-
bers, booleans, or even structures such as lists of maps }
]
}
]
}

```

4.4.2 Supported asset metadata

While arbitrary key/value pairs are accepted, some keys are pre-defined for common use-cases, making it easier to standardize most projects.

Keep in mind that only VOD has been specified for now, and that even for EPG, some metadata would be attached to the broadcast, not the asset.

Name in the ingest file	Type	Example	Platform equivalent	Description
country	string	"Spain"		AD_CTRYNATI
production_year	integer	2007		TS_PRODYEAR
release_year	integer	2007		TS_RELEDATE
morality_level_full	string	"R"		AD_MLRAWVAL
morality_level_type	string	"MPAA"		AD_MLTYPENA
summary	array of strings	["part 1", "part 2"]		AD_SUMMARY

georestriction	string	"bel"	AD_GEO-REST	Three-letter country code, as defined in ISO 3166 (alpha-3).
scores	dictionary (set of key/value pairs) with values as numbers	{ "Rotten Tomatoes": 88, "Telerama": 3 }	many AD_SCORxxxx metadata	
budget	integer	2000000	AD_BUDGET	
tags	array of strings	["Blockbuster", "highlight"]	AD_LIKEYWOR	

external_ids	dictionary (set of key/value pairs) with values as strings	{ "IMDB": "tt1038988" }	AD_EXTERIDS	
sport_championship	string	"Ligue 1 Conforama"	AD_SPORTCHM	
sport_event	string	"Nantes / Montpellier"	AD_SPORTEVT	
sport_location	string	"La Beaujoire"	AD_SPORTLOC	
actors	array of strings	["Manuela Velasco"]	AD_LIACTORS	
directors	array of strings	["Jaume Balagueró"]	AD_LIREALIS	
tvod_price	integer	5	AD_TCMPRICE	
external_links	dictionary (set of key/value pairs) with values as strings	{ "website": "http://alpha-networks.tv" }	AD_EXTLINKS	

4.4.3 Supported media metadata

While arbitrary key/value pairs are accepted, some keys are pre-defined for common use cases, making it easier to standardize most projects

Name in the ingest file	Type	Example	Platform equivalent	Description
drm_resource_id	string	48f84734-9248-4009-90d4-ea70cd04bf7	io.tucano.drm-ResourceId	For assets that have already been provisioned in a DRM system

4.4.4 Notes

Translations

No distinction is made between translated titles (synopsis, subtitles...) and "technical" (or default) titles (synopsis, subtitles...). Instead, the default will be decided by the following priority list:

- platform's language
- first language in the provided list

In most cases, only one language will be provided

4.5 categories.json

This file contains the categories, in the platform sense of the term, as a flat list. This allows the integrator to prepare the data to be exported as a relational database, and then simply "dump" the content of the tables.

4.5.1 Example file

```
{
  "version": "1.1",
  "categories": [
    {
      "id": "7b4ef709-51d8-438c-b2b1-4123dc5c6756", // Mandatory string
      "name": { // mandatory
        "eng": "lorem ipsum ...."
      },
      "description_translations": { // optional
        "eng": "lorem ipsum ..."
      },
      "hidden": false, // mandatory; whether the category
is hidden to end-users
      "morality_level": 18, // mandatory; morality level of the category. Users under this age won't be able to see it
      "parental_control": false, // mandatory boolean; whether to require
parental control (pin code) to access this category
      "adult": false, // mandatory boolean; whether this category contains
adult content
      "downloadable": false, // Optional (default is
null), meaning "use the value on the asset".
      "offline": false, // Optional (default is null), meaning
      "use the value on the asset".
      "metadata": {}, // mandatory (but can be empty). Arbitrary
key/value pairs. Keys are strings. Values can be text,
numbers, booleans, or even structures such as lists of maps.
      "images": [ // mandatory (can be empty)
        {
          "id": "7b4ef709-51d8-438c-b2b1-4123dc5c6754", "type": "poster", // mandatory. Types to be agreed beforehand (usual ones
are "poster" and "background")
          "language": "eng", // optional, defaults to null. ISO 639-3 language code
          // Mandatory
          // the image location. Valid values are:
          // - Absolute remote URLs:
          // -
http(s)://(user:password@)server.com(:80)/a/b/c/d.png // - ftp(s)://(user:password@)my-ftp
server(:21)/vod/posters.123456.svg
          // - sftp://(user:password@)my-ftp
server(:22)/vod/posters.123456.svg
          // - s3://keyId:keySecret@aws-region/bucket
name/test.png
          // - Relative local URLs:
          // - images/123.jpeg
          "location":
"https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Earth_by_the_EPIC_Team_on_21_April_2018.png/280px
Earth_by_the_EPIC_Team_on_21_April_2018.png"
        }
      ]
    }
  ]
}
```



```
],
  "series_id": "523c4c5c-68c5-4e00-a98b-754b9e9f4b71",
  // optional (default: null). Reference to a series defined
  in series.json. Mandatory if season_id provided
  "season_id": "759bab6c-1168-4ed5-ab91-b4ec41501daa",
  // optional (default: null). Reference to a season defined
  in series.json.
  "parent_id": "", // mandatory. Reference to the parent cate-
  gory, as defined in categories.json. If there is no pa-
  rent, null or empty.
  "assets": [
    "123c4c5c-68c5-4e00-a98b-754b9e9f4b71"
  ]
}
]
```

4.6 series.json

4.6.1 Example file

```
{
  "version": "1.1",
  "series": [
    {
      "id": "523c4c5c-68c5-4e00-a98b-754b9e9f4b71", // mandatory
      string
      "external_ids": { // mandatory; can be em-
      pty "tmdb": 123,
      "imdb": "tt5834204",
      "anmdb": "b375bbfc-6a76-4046-a486-b444af96cdc4" },
      "title": { // mandatory
      "eng": "lorem ipsum ..."
      },
      "synopsis": { // mandatory
      "eng": "lorem ipsum ..."
      },
      "images": [ // mandatory
      {
        "id": "7b4ef709-51d8-438c-b2b1-4123dc5c6754", "type":
        "poster", // mandatory
        "language": "eng", // optional (default null) // Man-
        datory
        // the image location. Valid values are: // -
        Absolute remote URLs:

```

```
// - http(s)://(user:password@)server.com(:80)/a/b/c/d.png // -  
ftp(s)://(user:password@)my-ftp server(:21)/vod/posters.123456.svg  
// - sftp://(user:password@)my-ftp server(:22)/vod/posters.123456.svg  
// - s3://keyId:keySecret@aws-region/bucket name/test.png  
// - Relative local URLs:  
// - images/123.jpeg  
"location":  
"https://upload.wikimedia.org/wikipedia/commons/thumb/d/d9/Earth_by_the_EPIC_Team_on_21_April_2018.png/280pxEarth_by_the_EPIC_Team_on_21_April_2018.png"  
},  
"seasons": [ // mandatory  
{  
  "id": "759bab6c-1168-4ed5-ab91-b4ec41501daa", // mandatory  
  "external_ids": { // optional  
    "tmdb": 123,  
    "imdb": 456,  
    "anmdb": "b375bbfc-6a76-4046-a486-b444af96cdc4" },  
  "season_number": 1,  
  "title": { // mandatory  
    "eng": "lorem ipsum ..."  
  },  
  "synopsis": { // mandatory  
    "eng": "lorem ipsum ..."  
  },  
  "images": [ // mandatory  
    {  
      "id": "7b4ef709-51d8-438c-b2b1-4123dc5c6754", "type":  
"poster", // mandatory  
      "language": "eng", // optional  
      // Mandatory  
      // the image location. Valid values are: // - Absolute remote URLs:  
      // - http(s)://(user:password@)server.com(:80)/a/b/c/d.png // -  
      ftp(s)://(user:password@)my-ftp server(:21)/vod/posters.123456.svg  
      // - sftp://(user:password@)my-ftp server(:22)/vod/posters.123456.svg  
      // - s3://keyId:keySecret@aws-region/bucket name/test.png  
      // - Relative local URLs:  
      // - images/123.jpeg
```

```
"location":  
  "https://upload.wikimedia.org/wikipedia/com-  
mons/thumb/d/d9/Earth_b  
y_the_EPIC_Team_on_21_April_2018.png/280px  
Earth_by_the_EPIC_Team_on_21_April_2018.png"  
}  
]  
}  
]  
}  
]  
}
```