PallyCon CLI Packager Guide

(Version 3.6.4, 2020.12.18)

Table of contents

1. Overview

This document explains about PallyCon Multi-DRM cloud service's content packaging concept and how to use the packager tool.

In PallyCon Multi-DRM service, there are five types to content packaging.

1. **NCG** : INKA Entworks' proprietary DRM Spec. (NCG : Netsync Content Guard)

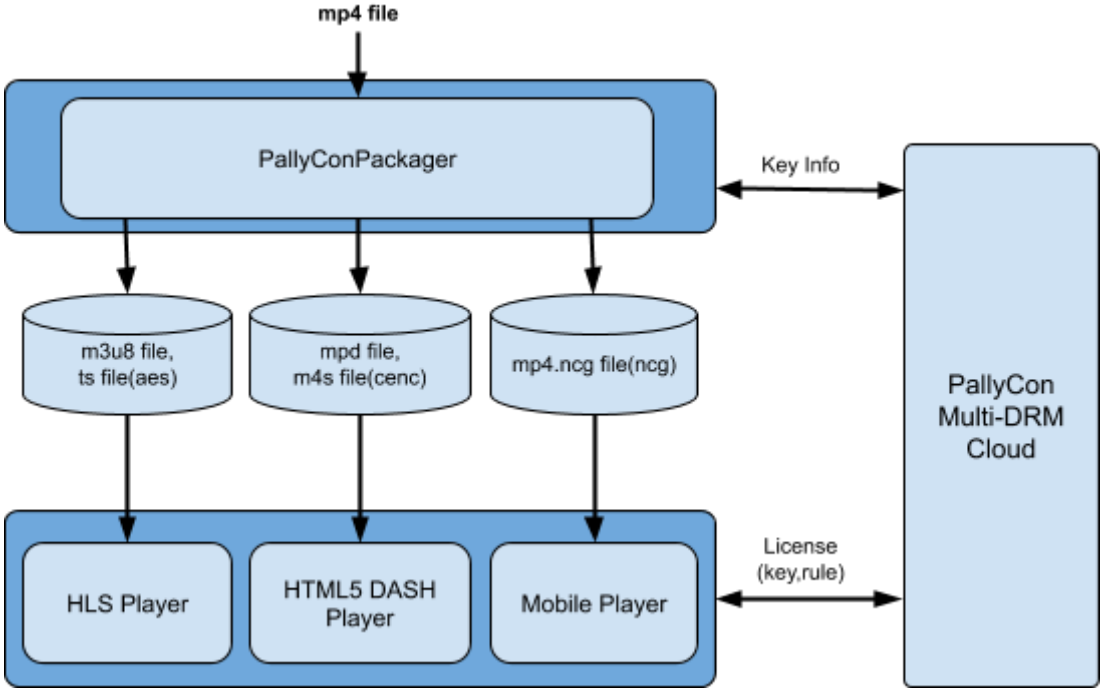
2. **CMAF** : Common Media Application Format output with Widevine, PlayReady, FPS DRM

3. **DASH** : An encryption spec for MPEG-DASH CENC and used with Widevine Modular and PlayReady DRM.

4. **HLS** : An encryption spec for HLS-AES and used with FPS(FairPlay Streaming) DRM.

5. **HLS-NCG** : Clear key of HLS-AES encryption is managed by NCG DRM.

2. Integration Structure and Environment



PallyConPackager runs with PallyCon Multi-DRM Cloud server.

PallyCon Cloud server manage Content Key data by service site, and issues license by finding Key(CEK) bound with CID when Player requests DRM License.

※ There must be available disk space on the system at least two-times larger than contents.

※ The file names are only allowed in English and numbers.

3. Command Line Arguments

- **You can use Shaka's command options in conjunction with the options below.**

Name	Value	Required	Description
--site_id	string	Y	PallyCon Service Site ID (4 bytes)
--access_key	string	Y	Access authentication Key (can be found on PallyCon Admin Site) - Go to 'Settings' page on Admin site
--content_id	string	Y	Content ID manual input, maximum 200 bytes If this parameter is omitted, Gateway callback page will be used to get CID for the content
--cmaf	bool	Y	CMAF(Widevine, PlayReady, FPS) packaging
--dash	bool		DASH-CENC(Widevine, PlayReady) packaging
--hls	bool		HLS-AES(FPS) packaging
--ncg	bool		NCG packaging
--hls_ncg	bool		HLS-AES(NCG) Packaging Packaging content using NCG DRM after HLS-AES encryption
-i (--input_file)	string	Y	filepath When two or more files are input, it is applied to Adaptive-Streaming packaging [Additional options] name : track name (:name=<value>) lang : audio track language (:lang=<value>) video_bitrate : video track bandwidth (:video_bitrate:<value>)
-o (--output_dir)	string	N	Output folder name. (Overwrite if use '-f' option) Default : Create output directory in current location
--config_file	string	N	A file path that stores fixed values such as Site ID and Access key. Json and XML format are supported. Default : Json (xml parsing is applied if the file extension is 'xml')
--clear_lead	string	N	Encryption disabled section Default : 0 (seconds)
--skip_audio_encryption	bool	N	Audio track encryption omit
--multi_key	bool	N	Apply multiple key sets for each track
--max_sd_height	string	N	Set the SD height limit Default : 480
--max_hd_height	string	N	Set the HD height limit Default : 1080

--max_uhd1_height	string	N	Set the UHD1 height limit Default : 2160
--fragment_duration	string	N	Fragment duration
--segment_duration	string	N	Segment duration
--on_demand	bool	N	Apply on-demand profile for DASH packaging
--output_single_file	bool	N	Output HLS packaging output as fMP4
--mpd_filename	string	N	Manifest file name for DASH
--m3u8_filename	string	N	Master manifest file name for HLS
--subtitle	string	N	Subtitle file path [Additional options] name : track name (:name=<value>) lang : subtitle language (:lang=<value>)
--generate_tracktype_manifests	bool	N	Generate multiple manifests for each track resolution
--enable_average_bandwidth_mpd	bool	N	Use average bandwidth for each track in MPD Default : False
--skip_pallycon_custom_info	bool	N	Skip PallyCon custom info in manifest file(mpd, m3u8) Default : False
--stop_indicator	bool	N	Hide the packaging indicator
--quiet	bool	N	Hide the output log

Parameters required when using external key

(--site_id and --access_key options are not used when using external keys.)

Name	Value	Required	Description
--enable_raw_key_encryption	bool	Y	Whether to use external key
--provider	string	N	Provider filed in Widevine PSSH Default : inkaentworks
--license_url	string	N	License acquisition URL - Default : https://license.pallycon.com/ri/licenseManager.do
--keys	string	Y	Pair of key and key id(Hex)
--ncg_cek	string	Y (If use NCG DRM)	32 bytes NCG encryption key(Hex)
--iv	string	N	16 bytes Initialization Vector(Hex)
--pssh	string	N	Manually specify PSSH for PlayReady and Widevine

4. Error Code

Result format

```
<?xml version="1.0" encoding="UTF-8"?>
<PallyconPackager>
<RESULT>0</RESULT>
</PallyconPackager>
```

Error code(RESULT): “0” if succeeded , other code if failed.

Error Code	Description
0	Success
1101	No parameter input
1102	Invalid input parameter (refer to INFO)
1103	Invalid number of parameters
1201	Cannot find the file in the path
1202	Cannot access the file (access right or filename issue)
1203	Failed to create file/folder. (too long path)
1204	Failed to create file/folder. (too long filename)
1205	Failed to create file/folder. (Insufficient disk space)
1206	Failed to move the file to the folder.
1207	Failed to get file size.
2001	An error occurred while sending request to server.
2002	An error returned from server. (refer to INFO)
2003	Invalid block size
2004	No private key found.
2005	Invalid private key

5. Example

- You need to sign up on PallyCon cloud service to get your site id and access key

HLS-NCG Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> **--hls_ncg** -i <input file> -o <output directory>

HLS Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> **--hls** -i <input file> -o <output directory>

DASH Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> **--dash** -i <input file> -o <output directory>

CMAF Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> **--cmaf** -i <input file> -o <output directory>

NCG Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> **--ncg** -i <input file> -o <output directory>

Adaptive-Streaming Packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> --dash --hls **-i <input file1> <input file2> <input file3>** -o <output directory>

Using external key(NCG, HLS-NCG)

#./PallyConPackager --site_id <site id> **--content_id <content id> --enable_raw_key_encryption --ncg_cek <32bytes key> --ncg --hls_ncg** -i <input file> -o <output directory>

Using external key(DASH, HLS)

#./PallyConPackager **--content_id <content id> --dash --hls --enable_raw_key_encryption --keys <key pair (e.g. label=:key_id=<16 bytes key id>:key=<16 bytes key>)>** -i <input file> -o <output directory>

Using configuration file

#./PallyConPackager **--config_file <configuration file path>** -i <input file> -o <output directory> --content_id <content_id>

[Configuration file example]

config.txt (Default = Json)

```
{"site_id":"your site id", "access-key":”your access key"}
```

config.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
        <site-id>your site id</site-id>
        <access-key>your access key</access-key>
</config>
```

Subtitles packaging (Supported only external text file in vtt format)

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> -i <input file> -o <output directory> **--subtitle <subtitle file path1>:lang=en <subtitle file path2>:lang=ko:name=Korean <subtitle file path3>:lang=fr:name=French ...**

Live packaging (DASH or HLS)

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> -o <output directory> **--dash(or --hls) -i <input streams (e.g. udp://127.0.0.1:1234)> --preserved_segments_outside_live_window 10 --time_shift_buffer_depth 60**

※ Real-time packaging and playback testing can be performed using web servers such as IIS (Windows) or Apache / Nginx (Linux).

※ The only live stream protocol supported by Packager is UDP. If you want to package streams of other unsupported protocols, you can redirect using ffmpeg as follows:

```
# ffmpeg -i <input stream> -f mpegts -vcodec copy -acodec copy udp://127.0.0.1:1234
```

※ The live stream packaging function supported by the CLI packager is suitable for simple development tests or small services, and is not recommended for large-scale services that provide a large number of live channels. For these services, please use a commercial live streaming solution such as AWS Elemental or Wowza Streaming Engine.

Multi-key packaging

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> -o <output directory> --dash **-i <input file1> <input file2> --multi_key**

Multi-key packaging (external key)

#./PallyConPackager -o <output directory> --dash **-i <input file1> <input file2> --content_id <content id> --enable_raw_key_encryption --keys <key pair (e.g. label=SD:key_id=<16 bytes key id>:key=<16 bytes key>,label=HD:key_id=<16 bytes key id>:key=<16 bytes key>,label=AUDIO:key_id=<16 bytes key id>:key=<16 bytes key>)>**

Multiple manifests for each track

#./PallyConPackager --site_id <site id> --access_key <access key> --content_id <content id> -o <output directory> --dash --multi_key **-i <input file1> <input file2> ... --generate_tracktype_manifests**