



Query String Variant Selection with TV Repackager

Document no EDGS-215
Version A2
Created on 2025-11-20

CONFIDENTIAL
©Copyright AgileTV, 2025

Contents

1 Introduction 2

1.1 Confidentiality Notice 2

1.2 About this Document 2

1.3 References 2

1.4 History 2

2 Query string variant selection 2

2.1 Limitations 2

2.2 Overview 2

2.3 Bitfield construction 3

2.3.1 Variant flags 3

2.3.2 Configuration 3

2.3.3 Variant mask 4

1 Introduction

1.1 Confidentiality Notice

This document is confidential and may not be reproduced, distributed or used for any purpose other than by the recipient for the assessment, evaluation and use of AgileTV products, unless written permission is given in advance by AgileTV.

1.2 About this Document

This document describes how to set up filtering based on query strings in order to select what variants to include in multi-bitrate manifest served by the SW Repackager.

This method of filtering variants can be used in addition to the rule-based presentations of the SW Repackager that can filter variants based on variant properties such as bitrate, codec or language.

1.3 References

[1] EDGS-187 - Software Live Ingest User Guide

[2] EDGS-176 - SW Repackager User Guide

1.4 History

Version	Date	Changes
A1	2019-11-01	Initial version
A2	2022-02-22	Drop unmanaged VOD support

2 Query string variant selection

2.1 Limitations

Variant selection based on HTTP query strings can only be used to include or exclude variants from the manifest created by the SW Repackager, it cannot be used to change the order of the variants or enforce rules based on e.g. the bitrate or the codecs used. Please see the Presentations section of the SW Repackager User Guide for details on how to perform this type of variant filtering.

The use of query strings for variant selection is limited to sources of type `channel` or `managed VOD` content that is recorded from channel source. Ingested managed VoD content, e.g from SMIL + MP4, cannot be filtered using this method.

2.2 Overview

The query string filtering of variants is based on combining two bitfields. The first bitfield comes from the query string parameter `p=` in the incoming request to the SW Repackager. This is called the variant mask. Below are two example requests using variant mask query strings, one for a DASH manifest and one for an HLS playlist:

`http://example.com/__cl/cg:live/__c/channel1/__op/drm/__f/manifest.mpd?p=98`

`http://example.com/__cl/cg:live/__c/sports/__op/drm/__f/index.m3u8?p=610`

The second bitfield comes from the variant's track configuration in Live Ingest. These bits are called the variant flags. The image below shows a client requesting a channel manifest while including the variant mask `p=18` in the request (1).

The repackager then fetches the channel definition from the ingest node including the list of variants (2 and 3).

For each variant the repackager compares the `variantFlags` value with the variant mask provided by the client in a bitwise AND operation. If any bit is equal in the two bitfields the variant is included in the resulting manifest, otherwise it is excluded (4).

Finally the remaining variants are used to generate the manifest and stream it to the client (5).

After the filtering based on the variant mask has been performed the rules of the current presentation filters are applied.

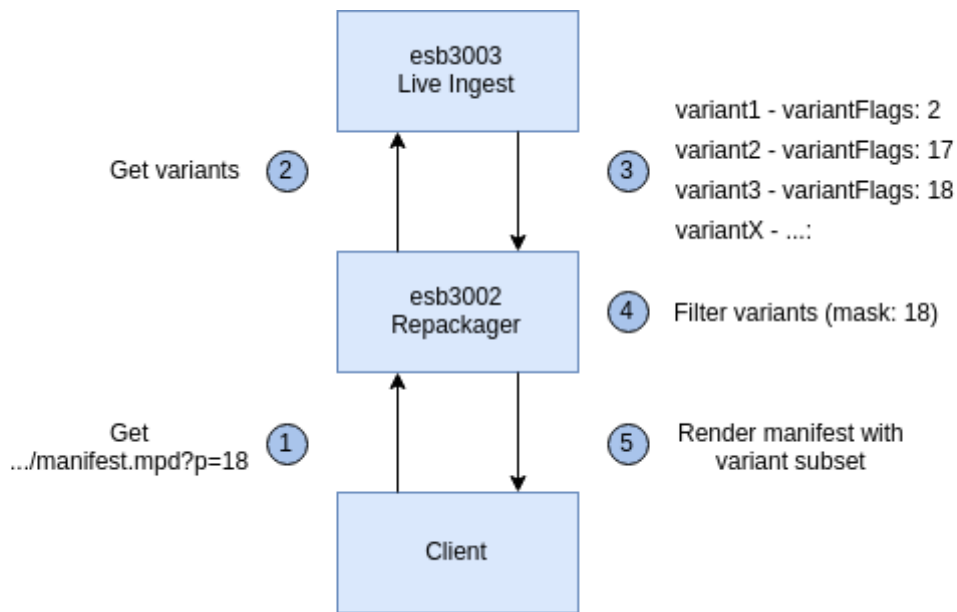


Figure 1: Variant selection overview

2.3 Bitfield construction

2.3.1 Variant flags

The meaning of each bit in the variant flags is use-case specific and depends on the reasons for including or excluding a variant. A reason could be device compatibility such as codec support, or the ability to play HD content.

For example to filter variants based on the following properties:

- SD vs HD video
- Stereo vs surround audio
- HEVC decoder support

SD can be defined as bit 1 (least significant bit) of the variant flags, HD as bit 2, and so on. With this definition in place, `variantFlags` can be set on all tracks in the channel based on the track's properties.

For instance, an HEVC video variant should have the bits for HEVC and HD set to 1, so it is given a `variantFlags` value of 18.

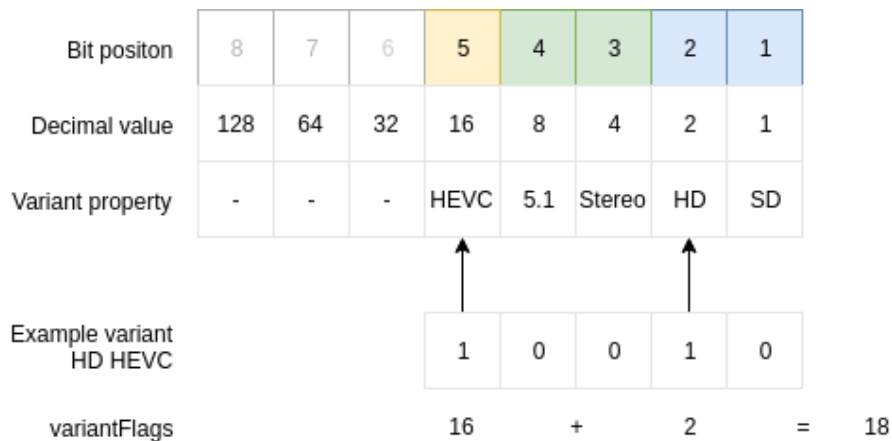


Figure 2: variantFlags construction

2.3.2 Configuration

Variant flags are configured as a track property named `variantFlags` in the channel configuration in the Live Ingest server. For example, to set the variant flags of the track `v1` in the first input source of the channel `ch1` using `confcli`:

```
confcli services.liveIngest.channels.ch1.inputSources.0.tracks.v1.variantFlags 28
```

If the `variantFlags` parameter is missing in the configuration of a track, or if it is set to the special value `-1`, then the variant will match any non-zero `p=` mask sent by the client.

The `variantFlags` parameter is a 32-bit integer number, so up to 32 different properties can be filtered using variant flags.

2.3.3 Variant mask

The variant masks to include in the manifest requests are calculated in the same way as the variant flags. For instance, to get a manifest with only SD video and stereo audio, bit 1 and 3 should be set, so providing the query string `p=5` will give the desired result:

Bit position	8	7	6	5	4	3	2	1
Decimal value	128	64	32	16	8	4	2	1
SD + Stereo				0	0	1	0	1
variantFlags	4 + 1 = 5							

Figure 3: SD + Stereo variant mask example

Similarly, to get a manifest with only HEVC content and 5.1 audio, bit 4 and 5 of the variant mask should be set, resulting in `p=24`.

Bit position	8	7	6	5	4	3	2	1
Decimal value	128	64	32	16	8	4	2	1
HEVC + 5.1				1	1	0	0	0
variantFlags	16 + 8 = 24							

Figure 4: HEVC + 5.1 variant mask example

If no `p=` query string is provided in the manifest request, all variants will be included. However, additional variant filtering is still applied based on the repackager's presentation settings.