



This repository Search

[Explore](#) [Gist](#) [Blog](#) [Help](#)

itpp16

kaltura / **nginx-vod-module**[Watch](#) 23[Star](#) 13[Fork](#) 4[155 commits](#)[1 branch](#)[0 releases](#)[2 contributors](#)branch: **master****nginx-vod-module** / +Merge pull request **#63** from kaltura/mss-playready**erankor** authored 3 days ago

latest commit 2ac3bfeffa















conf	avoid wildcard on CORS expose headers in HLS	2 months ago
static	add config files for integration with kaltura CE	2 months ago
test	add a setting for the domain of video segments	17 days ago
vod	add support for playready in mss	3 days ago
LICENSE	license should be AGPL not GPL	3 months ago
README.md	add a setting for the domain of video segments	17 days ago
config	add support for playready in mss	3 days ago
ngx_buffer_cache.c	stat improvements	3 months ago
ngx_buffer_cache.h	stat improvements	3 months ago
ngx_buffer_cache_internal.h	reduce the cache lock time	3 months ago
ngx_child_http_request.c	uri escaping fixes	a month ago
ngx_child_http_request.h	uri escaping fixes	a month ago
ngx_file_reader.c	align conventions	7 months ago

[Code](#)[Issues](#) 1[Pull requests](#) 0[Wiki](#)[Pulse](#)[Graphs](#)

HTTPS clone URL

<https://github.com/kaltura/nginx-vod-module>You can clone with [HTTPS](#), [SSH](#), or [Subversion](#).[Clone in Desktop](#)[Download ZIP](#)

ngx_file_reader.h	align conventions	7 months ago
ngx_http_vod_conf.c	add a setting for the domain of video segments	17 days ago
ngx_http_vod_conf.h	add a setting for the domain of video segments	17 days ago
ngx_http_vod_dash.c	add support for playready in mss	3 days ago
ngx_http_vod_dash.h	add more packagers	5 months ago
ngx_http_vod_dash_commands.h	make the use of dash SegmentTimeline configurable	3 months ago
ngx_http_vod_dash_conf.h	add more packagers	5 months ago
ngx_http_vod_hds.c	add support for encrypted dash (cenc)	3 months ago
ngx_http_vod_hds.h	add more packagers	5 months ago
ngx_http_vod_hds_commands.h	add more packagers	5 months ago
ngx_http_vod_hds_conf.h	add more packagers	5 months ago
ngx_http_vod_hls.c	vod_segments_base_url should affect only segments	17 days ago
ngx_http_vod_hls.h	add more packagers	5 months ago
ngx_http_vod_hls_commands.h	update tests and apply fixes	5 months ago
ngx_http_vod_hls_conf.h	update tests and apply fixes	5 months ago
ngx_http_vod_module.c	add a setting for the domain of video segments	17 days ago
ngx_http_vod_module.h	align conventions	7 months ago
ngx_http_vod_mss.c	add support for playready in mss	3 days ago
ngx_http_vod_mss.h	add more packagers	5 months ago
ngx_http_vod_mss_commands.h	add more packagers	5 months ago
ngx_http_vod_mss_conf.h	add more packagers	5 months ago
ngx_http_vod_request_parse.c	additional strict compilation warnings	4 months ago

 ngx_http_vod_request_parse.h	fix strict compilation issue	3 months ago
 ngx_http_vod_status.c	stat improvements	3 months ago
 ngx_http_vod_status.h	buffer cache changes	7 months ago
 ngx_http_vod_submodule.c	add more packagers	5 months ago
 ngx_http_vod_submodule.h	fix strict compilation issue	3 months ago
 ngx_http_vod_udrm.c	add support for playready in mss	3 days ago
 ngx_http_vod_udrm.h	add support for playready in mss	3 days ago
 ngx_http_vod_utils.c	revert unrelated change	3 days ago
 ngx_http_vod_utils.h	add a setting for the domain of video segments	17 days ago
 ngx_perf_counters.c	bug fix - wrong shared memory size	4 months ago
 ngx_perf_counters.h	stat improvements	3 months ago
 ngx_perf_counters_x.h	add support for encrypted dash (cenc)	3 months ago
 ngx_simple_json_parser.c	add json parser tests	3 months ago
 ngx_simple_json_parser.h	add support for encrypted dash (cenc)	3 months ago

README.md

NGINX-based VOD Packager

nginx-vod-module

Features

- On-the-fly repackaging of MP4 files to DASH, HDS, HLS, MSS
- Adaptive bitrate support
- Working modes:
 - i. Local - serve locally accessible files (local disk/NFS mounted)
 - ii. Remote - serve files accessible via HTTP using range requests
 - iii. Mapped - perform an HTTP request to map the input URI to a locally accessible file
- Fallback support for file not found in local/mapped modes (useful in multi-datacenter environments)
- Video codecs: H264, H265 (dash only)
- Audio codecs: AAC
- Audio only/video only files
- Track selection for multi audio/video MP4 files
- Source file clipping (only from I-Frame to P-frame)
- Support for variable segment lengths - enabling the player to select the optimal bitrate fast, without the overhead of short segments for the whole duration of the video
- Serving of files for progressive download playback
- DASH: common encryption (cenc) support
- HLS: Generation of I-frames playlist (EXT-X-I-FRAMES-ONLY)
- HLS: AES-128 encryption support

Limitations

- Only AAC audio is supported (MP3 audio is not)
- Clipping is not supported in progressive download
- I-frames playlist generation is not supported when encryption is enabled
- SAMPLE-AES encryption is not supported
- Tested on Linux only

Build

cd to NGINX source directory and execute:

```
./configure --add-module=/path/to/nginx-vod-module  
make  
make install
```

For asynchronous I/O support add `--with-file-aio` (highly recommended, local and mapped modes only)

```
./configure --add-module=/path/to/nginx-vod-module --with-file-aio
```

To compile nginx with debug messages add `--with-debug`

```
./configure --add-module=/path/to/nginx-vod-module --with-debug
```

To disable compiler optimizations (for debugging with gdb) add `CFLAGS="-g -O0"`

```
CFLAGS="-g -O0" ./configure ....
```

URL structure

Basic URL structure

The basic structure of an nginx-vod-module URL is: `http://<domain>/<location>/<fileuri>/<filename>`

Where:

- domain - the domain of the nginx-vod-module server
- location - the location specified in the nginx conf
- fileuri - a URI to the mp4 file:
 - local mode - the full file path is determined according to the root / alias nginx.conf directives
 - mapped mode - the full file path is determined according to the response from the upstream
 - remote mode - the mp4 file is read from upstream in chunks
 - Note: in mapped & remote modes, the URL of the upstream request is `http://<upstream>/<location>/<fileuri>?<extraargs>` (extraargs is determined by the vod_upstream_extra_args parameter)
- filename - detailed below

Multi URL structure

Multi URLs are used to encode several URLs on a single URL. A multi URL can be used to specify the URLs of several different MP4 files that should be included together in a DASH MPD for example.

The structure of a multi URL is: `http://<domain>/<location>/<prefix>,<middle1>,<middle2>,<middle3>,<postfix>.urlset/<filename>`

The sample URL above represents 3 URLs:

- `http://<domain>/<location>/<prefix><middle1><postfix>.urlset/<filename>`
- `http://<domain>/<location>/<prefix><middle2><postfix>.urlset/<filename>`
- `http://<domain>/<location>/<prefix><middle3><postfix>.urlset/<filename>`

The suffix `.urlset` (can be changed with `vod_multi_uri_suffix`) indicates that the URL should be treated as a multi URL.

URL path parameters

The following parameters are supported on the URL path:

- `clipFrom` - an offset in milliseconds since the beginning of the video, where the generated stream should start. For example, `.../clipFrom/10000/...` will generate a stream that starts 10 seconds into the video.
- `clipTo` - an offset in milliseconds since the beginning of the video, where the generated stream should end. For example, `.../clipTo/60000/...` will generate a stream truncated to 60 seconds.
- `tracks` - can be used to select specific audio/video tracks. The structure of parameter is: `v<id1>-v<id2>-a<id1>-a<id2>...` For example, `.../tracks/v1-a1/...` will select the first video track and first audio track. The default is to include all tracks.

Filename structure

The structure of filename is: `<basename>[<fileparams>][<trackparams>].<extension>`

Where:

- `basename + extension` - the set of options is packager specific (the list below applies to the default settings):
 - `dash` - `manifest.mpd`
 - `hds` - `manifest.f4m`
 - `hls master playlist` - `master.m3u8`

- hls media playlist - index.m3u8
- mss - manifest
- fileparams - can be used to select specific files (URLs) when using multi URLs. For example, manifest-f1.mpd will return an MPD only from the first URL.
- trackparams - can be used to select specific audio/video tracks. For example, manifest-a1.f4m will return an F4M containing only the first audio stream. The default is to include the first audio and first video tracks of each file. The tracks selected on the file name are AND-ed with the tracks selected with the /tracks/ path parameter.

Common configuration directives

vod

- **syntax:** vod segmenter
- **default:** n/a
- **context:** location

Enables the nginx-vod module on the enclosing location. Currently the allowed values for `segmenter` are:

1. `none` - serves the MP4 files as is
2. `dash` - Dynamic Adaptive Streaming over HTTP packetizer
3. `hds` - Adobe HTTP Dynamic Streaming packetizer
4. `hls` - Apple HTTP Live Streaming packetizer
5. `mss` - Microsoft Smooth Streaming packetizer

vod_mode

- **syntax:** vod_mode mode
- **default:** local

- **context:** http, server, location

Sets the file access mode - local, remote or mapped (see the features section above for more details)

vod_status

- **syntax:** vod_status
- **default:** n/a
- **context:** location

Enables the nginx-vod status page on the enclosing location.

vod_multi_uri_suffix

- **syntax:** vod_multi_uri_suffix suffix
- **default:** .urlset
- **context:** http, server, location

A URL suffix that is used to identify multi URLs. A multi URL is a way to encode several different URLs that should be played together as an adaptive streaming set, under a single URL. When the default suffix is used, an HLS set URL may look like: `http://host/hls/common-prefix,bitrate1,bitrate2,common-suffix.urlset/master.m3u8`

vod_segment_duration

- **syntax:** vod_segment_duration duration
- **default:** 10s
- **context:** http, server, location

Sets the segment duration in milliseconds.

vod_bootstrap_segment_durations

- **syntax:** `vod_bootstrap_segment_durations duration`
- **default:** `none`
- **context:** `http`, `server`, `location`

Adds a bootstrap segment duration in milliseconds. This setting can be used to make the first few segments shorter than the default segment duration, thus making the adaptive flavor selection kick-in earlier without the overhead of short segments throughout the video.

vod_align_segments_to_key_frames

- **syntax:** `vod_align_segments_to_key_frames on/off`
- **default:** `off`
- **context:** `http`, `server`, `location`

When enabled, the module forces all segments to start with a key frame. Enabling this setting can lead to differences between the actual segment durations and the durations reported in the manifest (unless `vod_manifest_segment_durations_mode` is set to `accurate`).

vod_segment_count_policy

- **syntax:** `vod_segment_count_policy last_short/last_long/last_rounded`
- **default:** `last_short`
- **context:** `http`, `server`, `location`

Configures the policy for calculating the segment count, for `segment_duration = 10` seconds:

- `last_short` - a file of 33 sec is partitioned as - 10, 10, 10, 3
- `last_long` - a file of 33 sec is partitioned as - 10, 10, 13
- `last_rounded` - a file of 33 sec is partitioned as - 10, 10, 13, a file of 38 sec is partitioned as 10, 10, 10, 8

vod_manifest_segment_durations_mode

- **syntax:** `vod_manifest_segment_durations_mode estimate/accurate`
- **default:** `estimate`
- **context:** `http`, `server`, `location`

Configures the calculation mode of segment durations within manifest requests:

- `estimate` - reports the duration as configured in `nginx.conf`, e.g. if `vod_segment_duration` has the value `10000`, an HLS manifest will contain `#EXTINF:10`
- `accurate` - reports the exact duration of the segment, taking into account the frame durations, e.g. for a frame rate of `29.97` and `10` second segments it will report the first segment as `10.01`.
`accurate` mode also takes into account the key frame alignment, in case `vod_align_segments_to_key_frames` is on

vod_secret_key

- **syntax:** `vod_secret_key string`
- **default:** `empty`
- **context:** `http`, `server`, `location`

Sets the secret that is used to generate the TS encryption key, if empty, no encryption is performed.

vod_duplicate_bitrate_threshold

- **syntax:** `vod_duplicate_bitrate_threshold threshold`
- **default:** `4096`
- **context:** `http`, `server`, `location`

The bitrate threshold for removing identical bitrates, streams whose bitrate differences are less than this value will be considered identical.

vod_https_header_name

- **syntax:** `vod_https_header_name name`
- **default:** `empty`
- **context:** `http`, `server`, `location`

Sets the name of an HTTP header whose existence determines whether the request was issued over HTTPS. If not set, the decision is made according to the protocol used to connect to the nginx server. A common scenario for using this setting is a load-balancer placed before the nginx that performs SSL-offloading.

vod_segments_base_url

- **syntax:** `vod_segments_base_url url`
- **default:** `empty`
- **context:** `http`, `server`, `location`

Sets the base URL (usually domain only) that should be used for delivering video segments. When empty, the host header sent on the request will be used as the domain. The scheme (`http/https`) used in the returned URLs is determined by:

- the value of `vod_segments_base_url`, if it starts with `http://` or `https://`
- the existence of a request header whose name matches the value of `vod_https_header_name`, if `vod_https_header_name` is not empty
- the type of connection used to connect to the nginx server The setting currently affects only HLS.

vod_moov_cache

- **syntax:** `vod_moov_cache zone_name zone_size`
- **default:** `off`
- **context:** `http`, `server`, `location`

Configures the size and shared memory object name of the moov atom cache

vod_response_cache

- **syntax:** `vod_response_cache zone_name zone_size`
- **default:** `off`
- **context:** `http, server, location`

Configures the size and shared memory object name of the response cache. The response cache holds manifests and other non-video content (like DASH init segment, HLS encryption key etc.). Video segments are not cached.

vod_initial_read_size

- **syntax:** `vod_initial_read_size size`
- **default:** `4K`
- **context:** `http, server, location`

Sets the size of the initial read operation of the MP4 file.

vod_max_moov_size

- **syntax:** `vod_max_moov_size size`
- **default:** `128MB`
- **context:** `http, server, location`

Sets the maximum supported MP4 moov atom size.

vod_cache_buffer_size

- **syntax:** `vod_cache_buffer_size size`
- **default:** `256K`

- **context:** http, server, location

Sets the size of the cache buffers used when reading MP4 frames.

vod_child_request

- **syntax:** vod_child_request
- **default:** n/a
- **context:** location

Configures the enclosing location as handling nginx-vod module child requests (remote/mapped modes only) There should be at least one location with this command when working in remote/mapped modes. Note that multiple vod locations can point to a single location having vod_child_request.

vod_child_request_path

- **syntax:** vod_child_request_path path
- **default:** none
- **context:** location

Sets the path of an internal location that has vod_child_request enabled (remote/mapped modes only)

vod_upstream

- **syntax:** vod_upstream upstream_name
- **default:** none
- **context:** http, server, location

Sets the upstream that should be used for reading the MP4 file (remote mode) or mapping the request URI (mapped mode).

vod_upstream_host_header

- **syntax:** `vod_upstream_host_header host_name`
- **default:** `the host name of original request`
- **context:** `http`, `server`, `location`

Sets the value of the HTTP host header that should be sent to the upstream (remote/mapped modes only).

vod_upstream_extra_args

- **syntax:** `vod_upstream_extra_args "arg1=value1&arg2=value2&..."`
- **default:** `empty`
- **context:** `http`, `server`, `location`

Extra query string arguments that should be added to the upstream request (remote/mapped modes only). The parameter value can contain variables.

vod_connect_timeout

- **syntax:** `vod_connect_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for connecting to the upstream (remote/mapped modes only).

vod_send_timeout

- **syntax:** `vod_send_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for sending data to the upstream (remote/mapped modes only).

vod_read_timeout

- **syntax:** `vod_read_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for reading data from the upstream (remote/mapped modes only).

vod_path_mapping_cache

- **syntax:** `vod_path_mapping_cache zone_name zone_size`
- **default:** `off`
- **context:** `http`, `server`, `location`

Configures the size and shared memory object name of the path mapping cache (mapped mode only).

vod_path_response_prefix

- **syntax:** `vod_path_response_prefix prefix`
- **default:** `<?xml version="1.0" encoding="utf-8"?><xml><result>`
- **context:** `http`, `server`, `location`

Sets the prefix that is expected in URI mapping responses (mapped mode only).

vod_path_response_postfix

- **syntax:** `vod_path_response_postfix postfix`
- **default:** `</result></xml>`
- **context:** `http`, `server`, `location`

Sets the postfix that is expected in URI mapping responses (mapped mode only).

vod_max_path_length

- **syntax:** `vod_max_path_length length`
- **default:** `1K`
- **context:** `http`, `server`, `location`

Sets the maximum length of a path returned from upstream (mapped mode only).

vod_fallback_upstream

- **syntax:** `vod_fallback_upstream upstream_name`
- **default:** `none`
- **context:** `http`, `server`, `location`

Sets an upstream to forward the request to when encountering a file not found error (local/mapped modes only).

vod_fallback_connect_timeout

- **syntax:** `vod_fallback_connect_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for connecting to the fallback upstream (local/mapped modes only).

vod_fallback_send_timeout

- **syntax:** `vod_fallback_send_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for sending data to the fallback upstream (local/mapped modes only).

vod_fallback_read_timeout

- **syntax:** `vod_fallback_read_timeout timeout`
- **default:** `60s`
- **context:** `http`, `server`, `location`

Sets the timeout in milliseconds for reading data from the fallback upstream (local/mapped modes only).

vod_proxy_header_name

- **syntax:** `vod_proxy_header_name name`
- **default:** `X-Kaltura-Proxy`
- **context:** `http`, `server`, `location`

Sets the name of an HTTP header that is used to prevent fallback proxy loops (local/mapped modes only).

vod_proxy_header_value

- **syntax:** `vod_proxy_header_value name`
- **default:** `dumpApiRequest`
- **context:** `http`, `server`, `location`

Sets the value of an HTTP header that is used to prevent fallback proxy loops (local/mapped modes only).

vod_clip_to_param_name

- **syntax:** `vod_clip_to_param_name name`
- **default:** `clipTo`
- **context:** `http, server, location`

The name of the clip to request parameter.

vod_clip_from_param_name

- **syntax:** `vod_clip_from_param_name name`
- **default:** `clipFrom`
- **context:** `http, server, location`

The name of the clip from request parameter.

vod_tracks_param_name

- **syntax:** `vod_tracks_param_name name`
- **default:** `tracks`
- **context:** `http, server, location`

The name of the tracks request parameter.

vod_performance_counters

- **syntax:** `vod_performance_counters zone_name`
- **default:** `off`
- **context:** `http, server, location`

Configures the shared memory object name of the performance counters

Configuration directives - DRM

vod_drm_enabled

- **syntax:** `vod_drm_enabled on/off`
- **default:** `off`
- **context:** `http, server, location`

When enabled, the module encrypts the media segments according to the response it gets from the drm upstream. Currently supported only for dash.

vod_drm_clear_lead_segment_count

- **syntax:** `vod_drm_clear_lead_segment_count count`
- **default:** `1`
- **context:** `http, server, location`

Sets the number of clear (unencrypted) segments in the beginning of the stream. A clear lead enables the player to start playing without having to wait for the license response.

vod_drm_max_info_length

- **syntax:** `vod_drm_max_info_length length`
- **default:** `4K`
- **context:** `http, server, location`

Sets the maximum length of a drm info returned from upstream.

vod_drm_upstream

- **syntax:** `vod_drm_upstream upstream_name`
- **default:** `none`
- **context:** `http, server, location`

Sets the upstream that should be used for getting the DRM info for the file.

vod_drm_connect_timeout

- **syntax:** `vod_drm_connect_timeout timeout`
- **default:** `60s`
- **context:** `http, server, location`

Sets the timeout in milliseconds for connecting to the upstream.

vod_drm_send_timeout

- **syntax:** `vod_drm_send_timeout timeout`
- **default:** `60s`
- **context:** `http, server, location`

Sets the timeout in milliseconds for sending data to the upstream.

vod_drm_read_timeout

- **syntax:** `vod_drm_read_timeout timeout`
- **default:** `60s`
- **context:** `http, server, location`

Sets the timeout in milliseconds for reading data from the upstream.

vod_drm_info_cache

- **syntax:** `vod_drm_info_cache zone_name zone_size`
- **default:** `off`
- **context:** `http, server, location`

Configures the size and shared memory object name of the drm info cache.

vod_drm_request_uri

- **syntax:** `vod_drm_request_uri uri`
- **default:** `$uri`
- **context:** `http`, `server`, `location`

Sets the uri of drm info requests, the parameter value can contain variables. In case of multi url, \$uri will one of the sub URLs (a separate drm info request is issued per sub URL)

Configuration directives - DASH

vod_dash_absolute_manifest_urls

- **syntax:** `vod_dash_absolute_manifest_urls on/off`
- **default:** `on`
- **context:** `http`, `server`, `location`

When enabled the server returns absolute URLs in MPD requests

vod_dash_manifest_file_name_prefix

- **syntax:** `vod_dash_manifest_file_name_prefix name`
- **default:** `manifest`
- **context:** `http`, `server`, `location`

The name of the MPD file (an mpd extension is implied).

vod_dash_init_file_name_prefix

- **syntax:** `vod_dash_init_file_name_prefix name`

- **default:** `init`
- **context:** `http`, `server`, `location`

The name of the MP4 initialization file (an mp4 extension is implied).

vod_dash_fragment_file_name_prefix

- **syntax:** `vod_dash_fragment_file_name_prefix name`
- **default:** `frag`
- **context:** `http`, `server`, `location`

The name of the fragment files (an m4s extension is implied).

Configuration directives - HDS

vod_hds_manifest_file_name_prefix

- **syntax:** `vod_hds_manifest_file_name_prefix name`
- **default:** `manifest`
- **context:** `http`, `server`, `location`

The name of the HDS manifest file (an f4m extension is implied).

vod_hds_fragment_file_name_prefix

- **syntax:** `vod_hds_fragment_file_name_prefix name`
- **default:** `frag`
- **context:** `http`, `server`, `location`

The prefix of fragment file names, the actual file name is `frag-f<file-index>-v<video-track-index>-a<audio-track-index>-Seg1-Frag<index>` .

Configuration directives - HLS

`vod_hls_absolute_master_urls`

- **syntax:** `vod_hls_absolute_master_urls on/off`
- **default:** `on`
- **context:** `http`, `server`, `location`

When enabled the server returns absolute playlist URLs in master playlist requests

`vod_hls_absolute_index_urls`

- **syntax:** `vod_hls_absolute_index_urls on/off`
- **default:** `on`
- **context:** `http`, `server`, `location`

When enabled the server returns absolute segment URLs in media playlist requests

`vod_hls_absolute_iframe_urls`

- **syntax:** `vod_hls_absolute_iframe_urls on/off`
- **default:** `off`
- **context:** `http`, `server`, `location`

When enabled the server returns absolute segment URLs in iframe playlist requests

`vod_hls_master_file_name_prefix`

- **syntax:** `vod_hls_master_file_name_prefix name`
- **default:** `master`
- **context:** `http`, `server`, `location`

The name of the HLS master playlist file (an m3u8 extension is implied).

vod_hls_index_file_name_prefix

- **syntax:** vod_hls_index_file_name_prefix name
- **default:** index
- **context:** http, server, location

The name of the HLS media playlist file (an m3u8 extension is implied).

vod_hls_iframes_file_name_prefix

- **syntax:** vod_hls_iframes_file_name_prefix name
- **default:** iframes
- **context:** http, server, location

The name of the HLS I-frames playlist file (an m3u8 extension is implied).

vod_hls_segment_file_name_prefix

- **syntax:** vod_hls_segment_file_name_prefix name
- **default:** seg
- **context:** http, server, location

The prefix of segment file names, the actual file name is `seg-<index>-v<video-track-index>-a<audio-track-index>.ts`.

vod_hls_encryption_key_file_name

- **syntax:** vod_hls_encryption_key_file_name name
- **default:** encryption.key
- **context:** http, server, location

The name of the encryption key file name (only relevant when vod_secret_key is used).

Configuration directives - MSS

vod_mss_manifest_file_name_prefix

- **syntax:** vod_mss_manifest_file_name_prefix name
- **default:** manifest
- **context:** http, server, location

The name of the manifest file (has no extension).

Sample configurations

Local configuration

```
http {
    upstream fallback {
        server kalhls-a-pa.origin.kaltura.com:80;
    }

    server {
        open_file_cache          max=1000 inactive=5m;
        open_file_cache_valid    2m;
        open_file_cache_min_uses 1;
        open_file_cache_errors   on;

        aio on;

        location /content/ {
            vod;
            vod_mode local;
        }
    }
}
```

```
    vod_moov_cache moov_cache 512m;
    vod_fallback_upstream fallback;

    root /web/content;

    gzip on;
    gzip_types application/vnd.apple.mpegurl;

    expires 100d;
    add_header Last-Modified "Sun, 19 Nov 2000 08:52:00 GMT";
}
}
```

Mapped configuration

```
http {
    upstream kalapi {
        server www.kaltura.com:80;
    }

    upstream fallback {
        server kalhls-a-pa.origin.kaltura.com:80;
    }

    server {

        open_file_cache          max=1000 inactive=5m;
        open_file_cache_valid    2m;
        open_file_cache_min_uses 1;
        open_file_cache_errors   on;

        aio on;

        location /__child_request__/ {
```

```
        internal;
        vod_child_request;
    }

    location ~ ^/p/\d+/(sp/\d+)?serveFlavor/ {
        vod;
        vod_mode mapped;
        vod_moov_cache moov_cache 512m;
        vod_secret_key mukkaukk;
        vod_child_request_path /__child_request__/;
        vod_upstream kalapi;
        vod_upstream_host_header www.kaltura.com;
        vod_upstream_extra_args "pathOnly=1";
        vod_path_mapping_cache mapping_cache 5m;
        vod_fallback_upstream fallback;

        gzip on;
        gzip_types application/vnd.apple.mpegurl;

        expires 100d;
        add_header Last-Modified "Sun, 19 Nov 2000 08:52:00 GMT";
    }
}
}
```

Remote configuration

```
http {
    upstream kalapi {
        server www.kaltura.com:80;
    }

    server {
        location /__child_request__/ {
            internal;
        }
    }
}
```

```
    vod_child_request;
}

location ~ ^/p/\d+/(sp/\d+)?serveFlavor/ {
    vod;
    vod_mode remote;
    vod_moov_cache moov_cache 512m;
    vod_secret_key mukkaukk;
    vod_child_request_path /__child_request__/;
    vod_upstream kalapi;
    vod_upstream_host_header www.kaltura.com;

    gzip on;
    gzip_types application/vnd.apple.mpegurl;

    expires 100d;
    add_header Last-Modified "Sun, 19 Nov 2000 08:52:00 GMT";
}
}
```

Copyright & License

All code in this project is released under the [AGPLv3 license](#) unless a different license for a particular library is specified in the applicable library path.

Copyright © Kaltura Inc. All rights reserved.



