

TechNote – AltitudeCDN™ OmniCache™ Integration with movingimage Enterprise Video Platform

Version 1.0

AltitudeCDN™ OmniCache is a robust proxy cache that enables the efficient delivery of HTTP Live Streaming (HLS), DASH, MP4 progressive-download, and other video content across the enterprise. OmniCache provides a flexible, scalable, and distributed platform that delivers high-quality low-latency video, while optimizing limited enterprise Wide Area Network (WAN) resources.

OmniCache integrates with movingimage Enterprise Video Platform (EVP), allowing you to configure the video platform and supported players to:

- Efficiently deliver videos by retrieving video assets from local OmniCache nodes.
- Take advantage of related OmniCache features, such as load distribution and support for high availability failover.

This guide describes how to configure and support integrated operations between AltitudeCDN OmniCache and movingimage EVP video events.

Contents

Contents	1
Introduction	2
Requirements	2
Solution Summary	3
Example OmniCache Configurations	4
Basic Configuration.....	4
Example of a Rewritten Manifest.....	5
Using Certificates	5
Generalizing Caching Using the Custom Cache Key	6
Examples of Content URI's	7

Introduction

During Enterprise Video Platform(EVP) live and on-demand events, large numbers of viewers attempt to individually connect to external content delivery network (CDN) servers, creating a high demand on WAN network resources that can lead to congestion, poor video quality, and even service interruptions for event viewers or other network clients.

To avoid these issues, AltitudeCDN OmniCache provides local video-content caching features that can be configured to fully support movingimage EVP events. By using OmniCache, retrieval of video assets from external CDN servers is greatly reduced, and shared content is internally distributed to enterprise WAN clients. The flexibility of this approach allows you to properly engineer network bandwidth usage to avoid service degradation, and maintain a high-level quality of service (QoS).

OmniCache is internally integrated with movingimage Enterprise Video Platform and only requires a simple configuration that does not rely on Proxy Auto-config (PAC) setup or DNS spoofing.

Requirements

Using AltitudeCDN OmniCache with movingimage Enterprise Video Platform requires:

Item	Recommendation
OmniCache	OmniCache v1.6.2 or later – The OmniCache must be installed and running on an on-premises Windows or CentOS platform, positioned within the customer network so traffic can reach both the audience and external video source origin. For more information, see the <i>AltitudeCDN OmniCache Deployment Guide</i> .
Enterprise DNS	Enterprise Domain Name System (DNS) configuration identifying the IP address of the internal OmniCache node. Geo-aware DNS may be employed to provide location-specific OmniCache routing.

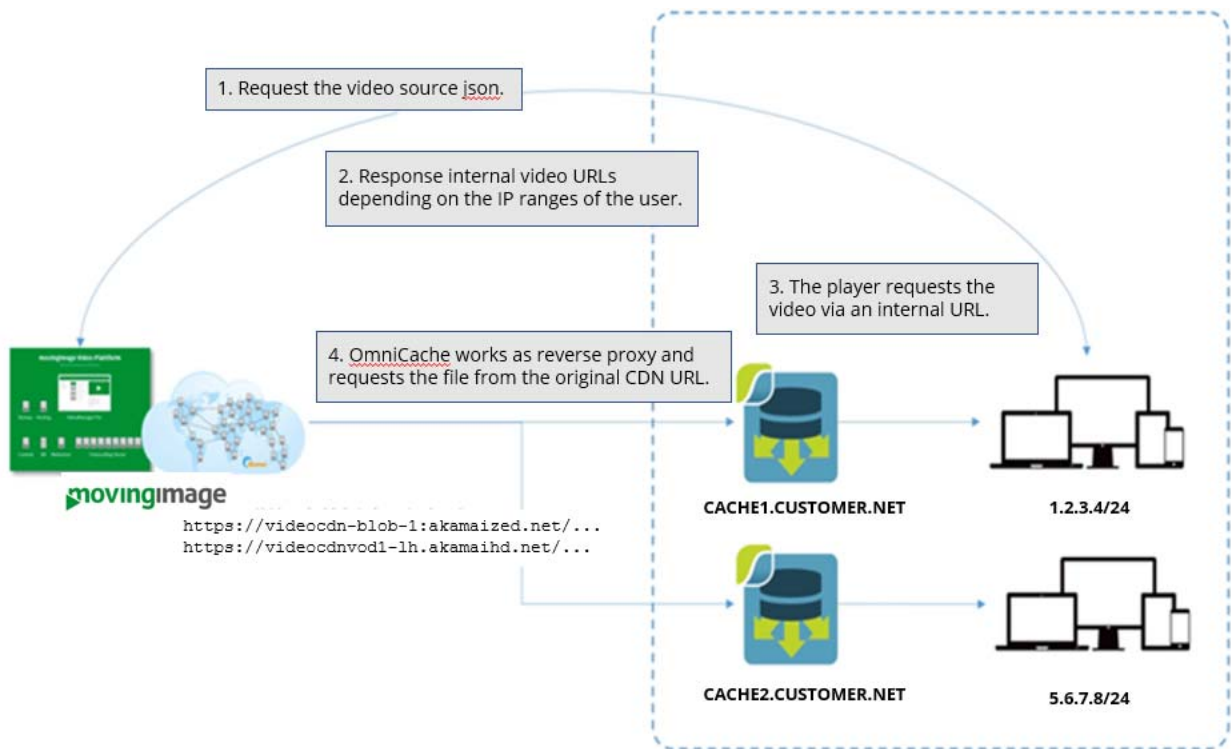
Solution Summary

To use AltitudeCDN OmniCache for movingimage EVP events, movingimage Customer Support must assign a specific DNS hostname for retrieving video assets (for example, cache.customer.net). If you have any custom requirements for steering HTTP requests to the most efficient OmniCache node, you can either:

- Request movingimage Customer Support assign different DNS hostnames based on the source IP address of the HTTP request.
- Use the Enterprise Domain Name System to assign different DNS hostnames based on the source IP address of the HTTP request.

Once OmniCache is configured, relevant HTTP requests from client browsers or video players are directed to an appropriate node that operates as a reverse-proxy:

- Video asset URIs issued from movingimage players are constructed in a format so that the OmniCache node retrieves upstream video assets from an external CDN when the content is not present in the cache.
- Following the initial HTTP request, OmniCache's unique routing features can additionally provide request routing that re-writes manifests to distribute subsequent request loads across a scalable set of OmniCache nodes:
 - Dynamic auto-discovery of other OmniCache nodes provides high availability in the event of a node failure.
 - If the local node is unreachable, or unable to serve the HTTP request, the player automatically retrieves video assets from the external CDN.



Example OmniCache Configurations

The sections below provide examples of how you can configure OmniCache nodes to process movingimage Enterprise Video Platform video assets.

Note: The examples below only show the relevant sub-section of the OmniCache configuration file.

See the *AltitudeCDN OmniCache Deployment Guide* and *AltitudeCDN OmniCache Reference Manual* for complete information on configuring OmniCache.

Basic Configuration

This example illustrates the deployment of a single OmniCache node that is configured as follows:

- Uses proxyEngines to operate as a reverse-proxy, and identifies the host identity of the cache.
- Uses routingEngine so that the video asset manifests are rewritten to target the OmniCache node.

```
"proxyEngines": [{
  "port": 443,
  "mode": "reversehttps",
  "httpsCertificateNames": ["cache.customer.net"]
}],
.
.
.
"routingEngine": {
  "standard": {
    "routingRules": [
      {
        "matches": [
          "0.0.0.0/0"
        ],
        "destinations": [
          {
            "http": cache.customer.net,
            "https": cache.customer.net,
            "weight": 100
          }
        ]
      }
    ]
  },
  "hls": "default",
  "hds": "default",
  "dash": "default"
}
```

Example of a Rewritten Manifest

The raw manifest data obtained from movingimage does not contain OmniCache directed URI's. The following example shows how the OmniCache routingEngine rewrites the manifest, so the player retrieves video assets from the appropriate OmniCache server:

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=256000,RESOLUTION=192x108,CODECS="avc1.66.30,mp4a.40.2"
https://cache.customer.net/omnicacheroutehls/https/videocdnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/index_0_av.m3u8?null=0
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=600000,RESOLUTION=512x288,CODECS="avc1.66.30,mp4a.40.2"
https://cache.customer.net/omnicacheroutehls/https/videocdnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/index_1_av.m3u8?null=0
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=834000,RESOLUTION=640x360,CODECS="avc1.66.30,mp4a.40.2"
https://cache.customer.net/omnicacheroutehls/https/videocdnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/index_2_av.m3u8?null=0
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1667000,RESOLUTION=854x480,CODECS="avc1.66.30,mp4a.40.2"
https://cache.customer.net/omnicacheroutehls/https/videocdnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/index_3_av.m3u8?null=0
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=2371000,RESOLUTION=1280x720,CODECS="avc1.66.30,mp4a.40.2"
https://cache.customer.net/omnicacheroutehls/https/videocdnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/index_4_av.m3u8?null=0
```

Using Certificates

If an OmniCache node is configured as a reverse-proxy, SSL/TLS certificates that match the customer hostname assigned to the node must be installed. Correctly generated and issued certificates allow the OmniCache node to serve authenticated content for the assigned DNS hostname (for example, cache.customer.net).

Generalizing Caching Using the Custom Cache Key

HTTP requests for video assets from client browsers typically include suffix arguments and tokens in their content URIs. These arguments should not be included in the caching criteria for an OmniCache node, so video assets that are retrieved on behalf of one client can be served to other client HTTP requests.

The OmniCache cacheEngine contains the proprietary customCacheKey property that allows you to ignore the additional arguments:

```
"rules":[
  {
    "request":{
      "matchesMode":"or",
      "matches":[
        {
          "host":".*akamaihd.net"
        }
      ]
    },
    "customCacheKey":{
      "matches":[
        {
          "uriPattern":"(.*)\\?.*",
          "uriReplacement":"$1"
        }
      ]
    },
    "action":{
      "matches":[
        {
          "contentType":".*application/vnd.apple.mpegurl.*",
          "cache":true,
          "fixedTtl":true,
          "defaultTtl":2
        },
        {
          "contentType":".*video/MP2T.*",
          "cache":true,
          "fixedTtl":true,
          "defaultTtl":86400
        }
      ]
    }
  }
]
```

Examples of Content URI's

The following example is an HLS M3U8 Manifest URI as supplied by an appropriately configured movingimage EVP account:

```
https://cache.customer.net/https/vidoeodnvod1-vh.akamaihd.net/i/9NXLzvt4CWWib-  
E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/master.m3u8?set-cc-  
attribute=cc?hdnea=exp=1508108847~acl=/i/9NXLzvt4CWWib-  
E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/*~hmac=16979b3912468377a  
51637b1e0eae240991614fa942cdc1ac11fae39d473c844
```

The following example is an HLS ts segment URI as seen by a client browser/player, where the OmniCache routingEngine has already re-written the manifest:

```
https://cache.customer.net/omnicacheroutehls/https/vidoeodnvod1-  
vh.akamaihd.net/i/9NXLzvt4CWWib-  
E3ri2r1r/5V7aUkKrRxi6DH1SioNsf9.aaa,256,596,828,1656,2356,.mp4.csmil/segment1_2_av.ts?null=0
```