Once VOD 2.0

Once VOD 2.0 is the next generation content ingest, lifecycle management, media management and dynamic permutation layer (DPL) subsystem for web video monetization and global playback with Once. The new system offers many advantages over its predecessor including simple and modular ingestion formats, cloud-based workflows, faster publishing times, and advanced DPL capabilities for multiple stream formats and content security packaging.

Implementation

New Customer Setup

A new Brightcove Once domain (an "account") will be created on your behalf.

We will need the following information to set up the new domain:

- Customer name
- Customer technical contact - first name, last name, email address, and phone number
- Customer CDN account information exchange
-
    - Brightcove will provide a hostname to use as CDN Origin with your provider along with configuration requirements
    - Upon configuration we will need your CDN edge hostname, which will be utilized during the playback process for video content delivery
- An understanding of how your media is broken up to create catalogs in the system for them

We will supply the following information to you for use in subsequent steps:

- API Key - system generated
- DomainId- system generated
- CatalogId(s) - system generated
- ApplicationId(s) - system generated

Important terms

Domain

The top level account for a Once customer used to tie catalogs, ad provider configurations, applications and publication rules together. In most cases, only one domain will be necessary.

Catalog

A digital library of video assets within a domain. Used for organizing and segregating different types of content. Typical implementations could have catalogs for different content types within a domain such as long form vs. short form or by property type with sports vs. news.

Application(ID)

A URL parameter used to determine preconfigured advertising calls and other business logic. Multiple applications may be configured in a domain to facilitate different configurations as needed.

Existing Customers

If you are currently using the Once service, please contact your account manager to schedule a migration appointment. For the purposes of existing customer testing, a new domain is required to ensure content and configuration segregation from customer production content.

Creating Ad decision server interaction

You can use any ad decisioning server that follows the IAB specification for VAST 2.0 or VAST 3.0, as well as Freewheel Smart XML GET requests or DoubleClick specific formats. Third party ad fill via wrappers will be handled by the Once system and no additional configurations will be necessary. Ad server interactions are configured on a per application basis meaning different ad servers can be used for multiple platforms.

The following information will be needed for proper configurations:

1. Full Ad tag of the desired ad adserver
2. Any dynamic values that should be appended to the ad tag request (e.g. metadata, geo information, device information, or page/site level parameters on request)
3. Any static values that should be appended to the ad tag (i.e. publisher ID)

After the application is configured, Once requests utilizing the application will make a request to the associated Ad Server to retrieve an ad payload. If the Ad Server returns a proper response and includes an ad creative that has not been previously returned, Once will automatically ingest that ad creative video within your domain into the expected encoding renditions. Video requests that receive a new ad response (creative) which is currently being ingested will be served with that slot unmonetized. After ingestion of the ad creative has completed Once will monetize subsequent requests.

Please see the Media Management API section to configure your ad server. Due to the complex nature of ad server configurations, please contact Brightcove for any questions that may arise.

Audience Measurement Systems

The Once system allows for the configuration and beaconing of HTTP based audience tracking URIs for both internal or external use. It allows the utilization of existing investments in reporting or analysis tools including providers like Omniture, Comscore, Nielsen, etc. The beaconing of these URIs can track both ads and/or content with measurements such as quartiles, heartbeats, content segments, start, and end events.

We will need to collect the following information:

1. HTTP endpoint
2. Any dynamic values that should be appended to the ad tag request (e.g. metadata, geo information, device information, or page/site level parameters on request)

These events will be configured on an application level. Please see the beacons section for additional configuration details.

Select transcoding renditions

By default, the following rendition set is configured within your Once domain. Variance is allowed in the renditions assigned to your domain, but at this time we do not allow for changes to each rendition's transcode settings.

RenditionId	BrevityId	Rendition Name	Width	Height	Video Bitrate	Audio Bitrate	Audio Sample Rate	Encoding Profile
5ff484d6-a33d-11e4-bfdb-005056837bc7	sd264	Once 264 ZC 256x144 200.64 B30	256	144	200	64	48	Baseline, 3.0
ac2e7f0b-a345-11e4-bfdb-005056837bc7	sd512	Once 512 ZC 384x216 448.64 B30	384	216	448	64	48	Baseline, 3.0
74ba4d0b-a347-11e4-bfdb-005056837bc7	sd764	Once 764 ZC 480x270 700.64 B30	480	270	700	64	48	Baseline, 3.0
076ea1a2-a35b-11e4-bfdb-005056837bc7	sd1200	Once 1200 ZC 640x360 1104.96 B31	640	360	1104	96	48	Baseline, 3.1
225bd8bb-a577-11e4-bfdb-005056837bc7	sd2000	Once 2000 ZC 960x540 1872.128 M31	960	540	1872	128	48	Main, 3.1

9adf3bc3-a578-11e4-bfdb-005056837bc7	hd3000	Once 3000 ZC 1280x720 2872.128 M31	1280	720	2872	128	48	Main, 3.1
468fb310-a585-11e4-bfdb-005056837bc7	hd4400	Once 4400 ZC 1280x720 4144.256 H40	1280	720	4144	256	48	High, 4.0
8efbb9ca-a586-11e4-bfdb-005056837bc7	hd6500	Once 6500 ZC 1920x1080 6244.256 H40	1920	1080	6244	256	48	High, 4.0
825b7f2a-a31b-11e4-bfdb-005056837bc7	audio	Once 56 ZC Audio Only	0	0	4	56	56	NA

Encoding Selection Guide

These recommended setting apply to VOD and considers content streamed via Wi-Fi and cellular networks. Generally you would select all renditions and leave the decisioning on the robust Once device detection logic; however, we understand that from a storage and cost perspective this may not be ideal. The guide below identifies which renditions apply to your target platform(s) to help you define a subset.

Note: These are *recommended* selections based on platform and lower or higher quality renditions can be selected to reflect business requirements for start time or stream quality.

Rendition Name	iPhone 3+	iPhone 5+	Android < 4.0.3	Android 4.0.3+	Android 4.4+	Tablets	Desktop	OTT/STB
Once 264 ZC 256x144 200.64 B30								
Once 512 ZC 384x216 448.64 B30								
Once 764 ZC 480x270 700.64 B30								
Once 1200 ZC 640x360 1104.96 B31								
Once 2000 ZC 960x540 1872.128 M31								
Once 3000 ZC 1280x720 2872.128 M31								
Once 4400 ZC 1280x720 4144.256 H40								
Once 6500 ZC 1920x1080 6244.256 H40								
Once 56 ZC Audio Only	as needed	as needed						

System Notifications

System notifications allow for a programmatic push based messaging service for major milestones through the content lifecycle. It provides significant insight by proactively notifying on specific events and allows easy integration into any invested publishing systems.

Configuring notifications

We will need to collect the following information:

1. A notification endpoint - SNS topic or a direct API endpoint
2. Types of notifications desired

Please note that the only method supported is POST and the endpoint must allow for this verb. If desired the same endpoint can be used for multiple notification methods, or each can be configured separately. For best practice we would advise using a single endpoint and parsing by each notification event type. There are currently six types of notification milestones supported:

Type	Description
INGEST	The content has passed through the queue and has begun processing.
UPDATE	A metadata update has been successful
TRANSCODE	A rendition of the content has completed processing (does not mean the item is ready to be published)
TIMEDTEXT	A timed text file has been validated and finished processing
ERROR	An error occurred in the processing of the content
PUBLISH	The content has completed processing and is available for delivery

Using a SNS topic for Notifications

The SNS Topic for notifications must be a topic to which the Once AWS account has permission to publish. To enable the Brightcove Once notification system to publish to your topic(s), you will first need to grant permission to the Once IAM ARN: arn:aws:iam::453163911362:root. This is done within your AWS Management Console.

Example policy statement

```
{
    "Sid": "console_pub",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::453163911362:root"
    },
    "Action": "SNS:Publish",
    "Resource": "{Desired_Topic_ARN}"
}
```

Notification examples

Ingest example:

```
{
    "notification": "ingest",
    "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
    "mediaItem": {
        "id": "b8eb0ce9-72f9-4e5c-8772-79d61add60eb",
        "foreignKey": "brightcove-once",
        "domainId": "b207b479-c841-4095-8918-978be9e18dee",
        "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
        "title": "brightcove-once",
        "version": 0,
        "metadata": {
            "key1": "value"
        }
    },
    "details": {
        "message": "Media Item Ingested."
    }
}
```

Update example:

```
{
    "notification": "update",
    "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
    "mediaItem": {
        "id": "b8eb0ce9-72f9-4e5c-8772-79d61add60eb",
        "foreignKey": "brightcove-once",
        "domainId": "b207b479-c841-4095-8918-978be9e18dee",
        "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
        "title": "brightcove-once",
        "version": 0,
        "metadata": {
            "key1": "value"
        }
    },
    "details": {
        "message": "Media Item Metadata Updated."
    }
}
```

Transcode example:

```
{
    "notification": "transcode",
    "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
    "mediaItem": {
        "id": "b8eb0ce9-72f9-4e5c-8772-79d61add60eb",
        "foreignKey": "brightcove-once",
        "domainId": "b207b479-c841-4095-8918-978be9e18dee",
        "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
        "title": "brightcove-once",
        "version": 0,
        "metadata": {
            "key1": "value"
        }
    },
    "details": {
        "message": "Rendition Transcoded.",
        "rendition": {
            "id": "45dd911d-54ac-11e4-9f45-005056835b09",
            "name": "Once 64 Audio-Only MP4 ZC 32x18 9 29.97 B3_0"
        }
    }
}
```

Timed Text example:

```
{
    "notification" : "timedtext",
    "requestId" : "8c337db5-801d-49f8-9f73-ef217ee7fac3",
    "mediaItem" : {
        "id" : "57f25048-f204-4996-8680-21452760a406",
        "foreignKey" : "brightcove-once-subtitle",
        "domainId" : "88f98026-95a7-4fe9-992a-94b6a1580483",
        "catalogId" : "b72da5a6-20ee-4bfb-aa2c-5722b6acc223",
        "title" : "brightcove-once-subtitle9",
        "version" : 0,
        "metadata" : { }
    },
    "timedTextItem" : {
        "id" : "ef5c6aa0-6745-4faa-b1d8-8d4f87a4e12b",
        "alternateId" : "microlanguage",
        "domainId" : "88f98026-95a7-4fe9-992a-94b6a1580483",
        "timedTextType" : "SUBTITLE",
        "version" : 0,
        "languages" : [ "en", "fr" ]
    },
    "details" : {
        "message" : "Timed Text Published."
    }
}
```

Error example:

```
{
    "notification":"error",
    "requestId":"2796350e-2125-4f04-b33a-59488aaa76c7",
    "mediaItem":{
        "id":"b8eb0ce9-72f9-4e5c-8772-79d61add60eb",
        "foreignKey":"brightcove-once",
        "domainId":"b207b479-c841-4095-8918-978be9e18dee",
        "catalogId":"bc6cb7d4-be99-471b-adf3-7c501172b317",
        "title":"brightcove-once",
        "version":0,
        "metadata":{
            "key1": "value"
        }
    },
    "details":{
        "message":"Description of the error event that occurred"
    }
}
```

Publish example:

```
{
    "notification":"publish",
    "requestId":"2796350e-2125-4f04-b33a-59488aaa76c7",
    "mediaItem":{
        "id":"479656c9-ec6d-464b-aba3-4739cdcc019b",
        "foreignKey":"brightcove-once",
        "domainId":"b207b479-c841-4095-8918-978be9e18dee",
        "catalogId":"bc6cb7d4-be99-471b-adf3-7c501172b317",
        "title":"brightcove-once",
        "version":0,
        "metadata":{
            "key1": "value"
        }
    },
    "details":{
        "message":"Media Item Published.",
        "onceUrl":"(Desired ApplicationId will need to be configured to utilize this field)"
    }
}
```

Notification Retry

Typically notifications will succeed on the first attempt, but there are myriad reasons for why it may fail. If the initial attempt does not result in a successful response from the endpoint, Brightcove will attempt to retry the initial delivery attempt up to three times. Any HTTP status codes 400-499, 500-599, connection timeouts, unreachable endpoints, or bad SSL certificates will result in a failed attempt. Each subsequent retry will be 30 seconds apart not including any queueing that may add an additional delay. If the final attempt does not succeed the notification will be dequeued and require a manual intervention. Please see the Status API documentation for re-sending failed notifications.

Content Ingestion

Creating an API-based ingest job

Brightcove's Once API Ingest service is a RESTful API that accepts HTTPS POST requests, supplying JSON as the data structure and passing the API key in the header. When an API Ingest job is successfully submitted, core renditions required for delivery on the Once VOD platform are generated. Videos at your SourceURLs must be progressive formats and publicly addressable URLs. Ingesting files via S3 is also supported by the API Ingest system as a source location. If your files are publicly available, no further action is necessary. If your files require authentication, please see the using S3 buckets section for authorizing the Once platform to access protected URLs. If the request fails, a validation error will be returned in the response body as JSON with details of the error event. Upon success the response will return a Request ID in the content body that can be used to track the job details from the Status API.

As a best practice recommendation your source video should be encoded with a video bitrate between 5-10 Mbps, frame size of 1920x1080 and contain keyframes at least every 6 seconds.

API Base URL:

```
https://api.unicornmedia.com/ingest-api/
```

Syntax:

```
https://api.unicornmedia.com/ingest-api/{domainId}/catalogs/{catalogId}
```

Example:

```
https://api.unicornmedia.com/ingest-api/b207b479-c841-4095-8918-978be9e18dee/catalogs/bc6cb7d4-be99-471b-adf3-7c501172b317
```

Creating an SNS-based ingest job

SNS Content Ingest is the service that allows for the automated ingest of assets and metadata from an Amazon SNS Topic to the Once platform. The SNS Ingest API waits for SNS Messages from all topics to which it is subscribed. Content is passed to the API by posting a Notification message to a topic that SNS Ingest API is subscribed to.

The following steps will be necessary to ensure Brightcove can configure the proper mechanisms to pull content from your AWS account.

1. The SNS Topic ARN
2. Allow the Brightcove AWS account root permission to subscribe to the SNS topic
    - Brightcove Root: **arn:aws:iam::453163911362:root**
    - Contact Brightcove to complete the configuration and submit a subscription request to your topic
3. Allow permissions to the transcoding engine when using S3 buckets with your content storage

Example policy statement

```
{
    "Sid": "console_pub",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::453163911362:root"
    },
    "Action": ["SNS:Subscribe",
    "SNS:Receive"],
    "Resource": "{Desired_Topic_ARN}",
    "Condition": {
        "StringEquals": {
            "SNS:Protocol": "sqs"
        }
    }
}
```

Using S3 buckets

If the bucket is not publicly accessible, the Brightcove transcoding engine will need to be granted access through a bucket policy or to the Brightcove AWS account.

Option #1 - Create a bucket policy

To set up a bucket policy, login to your AWS Console, select your bucket on the left side of the screen, then click "properties". On the last row click "Add bucket policy". We'll put a Bucket Policy together below, which you'll copy into the Bucket Policy text box.

Recommended bucket policy:

```
{
    "Version": "2008-10-17",
    "Id": "ZencoderBucketPolicy",
    "Statement": [
        {
            "Sid": "Stmt1295042087538",
            "Effect": "Allow",
            "Principal": {
            "AWS": "arn:aws:iam::395540211253:root"
        },
            "Action": [
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:PutObjectAcl",
                "s3:PutObject",
                "s3:ListMultipartUploadParts"
            ],
            "Resource": "arn:aws:s3:::MY-BUCKET/*"
        },
        {
            "Sid": "Stmt1295042087538",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::395540211253:root"
            },
            "Action": [
            "s3:ListBucketMultipartUploads",
            "s3:GetBucketLocation"
            ],
            "Resource": "arn:aws:s3:::MY-BUCKET"
        }
    ]
}
```

There is only one change you will have to make to this bucket policy before it's ready to go.

1. Replace **`MY-BUCKET`** with the name of your bucket. Leave the /* at the end of the line, as that will apply the policy to every file within the bucket.
2. Optionally, replace the **`Action`** array with the actions you want to allow. Valid actions that Brightcove may need are:
   Object Permissions - first action array in the example above

   - s3:PutObject - allow Brightcove to upload files to the bucket
   - s3:GetObject - allow Brightcove to download files from the bucket
   - s3:GetObjectAcl - allow Brightcove to get ACL permissions for files
   - s3:PutObjectAcl - allow Brightcove to set ACL permissions for other users on files
   - s3:ListMultipartUploadParts - allow Brightcove to list which parts of a multipart file have been uploaded to the bucket

   Bucket Permissions - second action array in the example above

   - s3:GetBucketLocation - allow Brightcove to get the location of the bucket (US, EU, Asia, etc.)
   - s3:ListBucketMultipartUploads - allow Brightcove to view any multipart uploads currently happening on the bucket

   Example: If you want Zencoder to be able to download and upload files, replace the first Action array with:

   ```
   ["s3:PutObject", "s3:GetObject"]
   ```

Note: If the source files are stored on S3, the URLs should be in the format **`s3://BUCKET/KEY`** to ensure the URLs are signed appropriately.

If you'd like to build your own Bucket Policy, you can use Amazon's Policy Generator (http://awspolicygen.s3.amazonaws.com/policygen.html). Use **`arn:aws:iam::395540211253:root`** as the AWS Principal to allow Brightcove access.

Option #2 - grant access to Brightcove AWS

Grant Access to the Brightcove AWS account at aws@zencoder.com. If the setup requires the canonical ID you can use:

```
6c8583d84664a381db0c6af0e79b285ede571885fbe768e7ea50e5d3760597dd
```

Example ingest syntax

Request Header

Required HTTP Header:

```
X-BC-ONCE-API-KEY : {yourApiKey}
```

Schema

See Appendix A

Example Body

```
{
    "title": "brightcove-once",
    "foreignKey": "brightcove-once",
    "metadata": {
        "accoundId": "12345",
        "company": "brightcove",
        "product": "once",
        "eventId": "321"
    },
    "media": {
        "sourceURL": "http://mirror.cessen.com/blender.org/peach/trailer/trailer_1080p.mov"
    },
    "priority": "normal",
    "publicationRules": [
        {
            "channel": "a8cf98a9-8946-4bd4-b97c-a2b5dbc635c5",
            "startDate": 1412025402,
            "endDate": 1601414189,
            "clientFilters": [
                {
                    "variableName": "IpAddress",
                    "value": "127.0.0.1",
                    "filterType": "Equals",
                    "isDenied": true
                }
            ],
            "countryRules": [
                {
                    "countryCode": "UK",
                    "isDenied": true
                }
            ]
        }
    ],
    "cuePoints": [
        {
            "valueIn": 10,
            "unit": "Seconds"
        },
        {
            "valueIn": 25,
            "unit": "Seconds"
        }
    ]
}
```

Response

Body (error)

```
{
    "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
    "error": "Validation Errors",
    "fieldErrors": {
        "publicationRule": [
            "Publication rule with end date 0.0 is in the past.",
            "Publication rule end date: 0.0 preceeds 1412025402"
        ]
    }
}
```

Body (success)

```
{
    "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7"
}
```

Brief Object Explanation

The table below can be used as a quick start for the major fields required for content ingestion along with a description. Please see the ingest schema for a full object mapping. If the required fields are not present in the API request; a response body will be returned with a corresponding validation error. Note: While we recommend the use of an optional field **`title`** for search and human readability, only the following highlighted fields are required for all ingest job requests.

Object	Sub-Object	Required	Description
title		No	Title of the asset, used for subsequent notifications and item lookup
foreignKey		Yes	Unique identifier per domain to indicate a media item and its associations
keywords		No	Keywords used to describe the media item
description		No	Brief description of the media item
metadata		No	Key-value pairs that can be stored with a media item and dynamically retrieved for audience beaconing or ad server requests
media		Yes	Container for the defined source video
	sourceURL	Yes	Location of the video media item
timedText		No	Defining or association closed captioning and/or subtitles to ingest in conjunction to the video media item
	media	Yes	Container for the defined source document of a subtitle or flagged as passthrough when the document is embedded on the video container
	timedTextType	Yes	Defines the timed text output as a subtitle, closed-caption, or embedded
	languages	Yes	Country code defined by ISO 639-1
	alternateId	No	Defines a timed text output when multiple variations of the same language is utilized. Output will be named as "{language}-{alternateId}"
priority		No	Reserved for future use.
publicationRules		No	Sets timing, geo filtering, and client filters to restrict content based on business requirements. By default, publication rules are open for 10 years from when account was opened. Please see Media Management for configuration details.
	channel	Yes	The channel Id associated with the domain; by default, a domain contains a single channel. If there are additional channels in a domain, this must be specified
	startDate	Yes	Begin content availability in epoch time (seconds)
	endDate	Yes	End content availability in epoch time (seconds)
	clientFilters	No	Allows the setting of client restrictions based on IP Address, useragent, or referring host
	countryRules	No	Geo restrictions based on ISO 639-1 country codes.
cuePoints		No	Segmentation points of the mediaitem to insert midroll ads (when available and configured).
	valueIn	Yes	Value of the cuepoint in relation to the content duration; This value must be an integer
	unit	Yes	Unit of measurement for the valueIn; currently only "seconds" is supported

Captioning / Subtitles

Brightcove Once enables multinational publishers to provide subtitles or closed-captioning to global audiences. These subtitles or closed captions can be rendered by players and native OS playback components that allow the viewer to choose the language of their choice, either via OS default localization settings or via a UI selection. Subtitles should not be confused with closed captions for the purpose of this implementation. While both utilize timed-text and in many cases in a similar manner, it is important to make the distinction from a regulatory and user experience perspective. In some cases, both will be required as a given piece of content could be published in North America and rest-of-world simultaneously. As digital technology evolves, closed-captioning will no longer be restrained by formatting by embedded captions and can be used in conjunction with timed-text outputs.

Subtitles input/output:

Subtitles		**Output**				
		embedded 608	**SRT**	**WebVTT (in-manifest)**	**WebVTT sidecar**	**DFXP**
Input	**Sidecar SRT**		✓	✓	✓	**[1-1]**
	Sidecar WebVTT		✓	✓	✓	**[1-1]**
	Sidecar DFXP		✓	✓	✓	✓

Notes:

[1-1] A simple timed-text input format to a rich output is not recommended, although supported by using default styling to work with most platforms.

Current Restrictions/Limitations

For SRT:

- Positioning and styling are not supported.
- Text formatting derived from HTML tags will be treated as text.
- Output as a sidecar through Once UX only

For WebVTT:

- Input files must not be segmented.
- Sidecar output file will be outputted as a full file.
- Positioning and styling are not supported.
- Text formatting derived from HTML tags will be treated as text.
- Output as a sidecar through Once UX only

For DFXP:

- Multiple languages in <div> tags in a single document is supported, but the languages must be defined at ingest.
- Output files will not support more than one <div> tag with its associated language. Each language will be a separate sidecar file.
- Output as WebVTT in-manifest for HLS will be segmented and follow the IETF HLS draft version 14.

Example Once UX presentation:

```
...
<uo:timedText>
    <uo:item type="subtitle" language="en">
        <![CDATA[http://.../en/content.{output_extension}]]>
    </uo:item>
    <uo:item type="subtitle" language="fr" alternateId="{altId}">
        <![CDATA[http://.../fr/content.{output_extension}]]>
    </uo:item>
</uo:timedText>
...
```

Note: The **`alternateId`** attribute will only be rendered if declared on ingest.

Example in-manifest WebVTT (master playlist):

```
...
#EXT-X-MEDIA:TYPE=SUBTITLES,URI="http://.../content.m3u8",GROUP-ID="subs",LANGUAGE="en",NAME="English"
#EXT-X-MEDIA:TYPE=SUBTITLES,URI="http://.../content.m3u8",GROUP-ID="subs",LANGUAGE="fr",NAME="French{altId}"
...
```

Note: The Name of the language code will have the alternateId appended in parentheses if declared on ingest. This allows for multiple files with the same language, microlanguages, or multiple selected audio streams.

Example in-manifest WebVTT (media playlist):

```
...
#EXTINF:240.000,0
http://.../segment1.vtt
#EXTINF:240.000,0
http://.../segment2.vtt
...
#EXT-X-ENDLIST
```

Closed-Captioning input/output:

Closed-Captioning				**Output**		
		CEA-608	**SRT**	**WebVTT (in-manifest)**	**WebVTT sidecar**	**DFXP**
Input	**CEA-608**	[2-1]				
	Sidecar SRT		✓		✓	[2-2]
	Sidecar WebVTT		✓		✓	[2-2]
	Sidecar DFXP	[2-1]	✓		✓	✓
	Sidecar SAMI	[2-1]				
	Sidecar SCC	✓				

Notes:

[2-1] If the input document (or file) contains multiple languages, only the first language is selected. The file format must be in a MP4/M4V video container with caption tracks in CEA-608 format.

[2-2] A simple timed-text input format to a rich output is not recommended, although supported by using default styling to work with most platforms.

Current Restrictions/Limitations

For CEA-608:

- Sidecar output will not be generated for extracted closed captioning
- Output as Closed-Captioning will be presented in HLS manifests following the IETF HLS draft version 14 (https%3A%2F%2Ftools.ietf.org%2Fhtml%2Fdraft-pantos-http-live-streaming-14&sa=D&sntz=1&usg=AFQjCNFY7fXFI6xzbBR-Q6PdJMbU7Zyu6w).

For SRT:

- Positioning and styling are not supported.
- Text formatting derived from HTML tags will be treated as text.
- Output as a sidecar through Once UX only

For WebVTT:

- Input files must not be segmented.
- Sidecar output file will will be outputted as a full file.
- Positioning and styling are not supported.
- Text formatting derived from HTML tags will be treated as text.
- Output as a sidecar through Once UX only

For DFXP:

- Multiple languages in <div> tags in a single document is supported, but the languages must be defined at ingest.
- Output files will not support more than one <div> tag with its associated language. Each language will be a separate sidecar file.
- Output as Closed-Captioning will be presented in HLS manifests following the IETF HLS draft version 14 (https%3A%2F%2Ftools.ietf.org%2Fhtml%2Fdraft-pantos-http-live-streaming-14&sa=D&sntz=1&usg=AFQjCNFY7fXFI6xzbBR-Q6PdJMbU7Zyu6w).
- As CEA-608 output:
-
    - Animation, layout, and region features are not supported.
    - Styles can be inline or referenced by ID, but only the tts:textAlign property is applied.
    - Nested DIV and P tags are ignored
    - SPAN tags are ignored
    - Explicit breaks may be inserted with BR tags

For SAMI:

- Inline styles are not supported
- Layout-related styling tags are not supported
- Explicit breaks may be inserted by using separate P tags.
- Output as Closed-Captioning will be presented in HLS manifests following the IETF HLS draft version 14 (https://www.google.com/url?q=https%3A%2F%2Ftools.ietf.org%2Fhtml%2Fdraft-pantos-http-live-streaming-14&sa=D&sntz=1&usg=AFQjCNFY7fXFI6xzbBR-Q6PdJMbU7Zyu6w).

Example Once UX presentation:

```
...
<uo:timedText>
    <uo:item type="caption" language="en">
        <![CDATA[http://.../en/content.{output_extension}]]>
    </uo:item>
    <uo:item type="caption" language="fr" alternateId="{altId}">
        <![CDATA[http://.../fr/content.{output_extension}]]>
    </uo:item>
</uo:timedText>
...
```

Note: the **`alternateId`** attribute will only be rendered if declared on ingest.

Status API

The Status API is a RESTful API which allows customers to query the processing statuses of a particular Request ID (returned in the response body of an ingest request). Only the HTTP GET verb is supported and must be made over SSL. If you do not know the Request ID you can query by Foreign Key, which will return a list of all the Request IDs for that particular item. The status data returned will be summary of the major milestones along with any errors that occurred during processing.

API Base URL:

`https://api.unicornmedia.com/status-api/`

Checking a Job Status

Syntax:

`https://api.unicornmedia.com/status-api/statuses/{referenceId}`

`https://api.unicornmedia.com/status-api/statuses?catalogId={catalogId}&foreignkey={foreignKey}`

Status Definitions:

Type	**Description**
PENDING	The activity or overall job has been received and scheduled in a queue
PROCESSING	The activity or overall job is currently being processed. The overall status will remain in processing if any activity has not been completed.
SKIPPED	The particular activity for the job was not configured or necessary for processing.
WARN	A particular activity has failed that does not impact playback, but the overall job has succeeded. (e.g. timed text)
ERROR	The activity or overall job has failed with an error. Any activity error will result in an overall job error.
COMPLETE	The activity has completed successfully.

Example: Status API Job Requests

Request ID (GET):

`https://api.unicornmedia.com/status-api/statuses/2796350e-2125-4f04-b33a-59488aaa76c7`

Foreign Key & Catalog (GET):

`https://api.unicornmedia.com/status-api/statuses?catalogId=bc6cb7d4-be99-471b-adf3-7c501172b317&foreignkey=brightcove-once`

Request header

Required HTTP Header:

`X-BC-ONCE-API-KEY : {yourApiKey}`

Response(s)

Body (Request ID)

```
{
    "status": "COMPLETE",
    "startTime": 1415648939000,
    "completeTime": 1415649217000,
    "steps": {
        "ingest": {
            "name": "ingest",
            "status": "COMPLETE",
            "startTime": 1415648939000,
            "completeTime": 1415648942000,
            "input": {
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "keywords": {
                    "brightcove",
                        "once"
                },
                "metadata": {
                    "accoundId": "12345",
                    "company": "brightcove",
                    "product": "once",
                    "eventId": "321"
                },
                "media": {
                    "sourceURL": "http://mirror.cessen.com/blender.org/peach/trailer/trailer_1080p.mov"
                },
                "notifications": [],
                "publicationRules": [],
                "cuePoints": [
                    {
                        "valueIn": 10,
                        "unit": "Seconds"
                    }, {
                        "valueIn": 25,
                        "unit": "Seconds"
                    }
                ]
            },
            "output": {
                "mediaItemId": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "hasChanged": true,
                "description": {
                    "url": "http://mirror.cessen.com/blender.org/peach/trailer/trailer_1080p.mov",
                    "lastModified": "Wed, 07 May 2014 22:21:59 GMT",
                    "etag": "\"b2f673c3426acf1:0\"",
                    "fileSize": 60047891
                }
            }
        },
        "transcode": {
            "name": "transcode",
            "status": "COMPLETE",
            "startTime": 1415648968000,
            "completeTime": 1415649216000,
            "output": {
                "renditions": [
                    "3345fa40-bbb6-11e3-87e6-005056835b09",
                    "45dd911d-54ac-11e4-9f45-005056835b09",
                    "8249c274-b49d-11e3-87e6-005056835b09",
                    "824d3f86-b49d-11e3-87e6-005056835b09",
                    "825072a0-b49d-11e3-87e6-005056835b09",
                    "8253a884-b49d-11e3-87e6-005056835b09",
                    "8256d34a-b49d-11e3-87e6-005056835b09",
                    "825a2774-b49d-11e3-87e6-005056835b09",
                    "825d7a04-b49d-11e3-87e6-005056835b09",
                    "8260dd62-b49d-11e3-87e6-005056835b09",
                    "d34d2cfb-3478-11e4-8b0b-005056835b09",
                    "d34d2d07-3478-11e4-8b0b-005056835b09"
                ]
            }
        },
        "notification": {
            "name": "notification",
            "status": "COMPLETE",
            "startTime": 1415649020000,
            "completeTime": 1415649217000,
            "input": {
                "mediaItemId": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "version": 0,
                "durationMS": 87754,
                "notificationOverrides": []
            },
            "output": {
                "sentNotification": false
            }
        },
        "publish": {
            "name": "publish",
            "status": "COMPLETE",
            "startTime": 1415649217000,
            "completeTime": 1415649217000,
            "output": {
                "version": 0,
                "duration": 87754
            }
        }
    }
}
```

Body (Foreign Key & Catalog)

```
{
    "requests": [
        {
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "startTime": 1412214761091,
            "href": "https://api.unicornmedia.com/status-api/statuses/2796350e-2125-4f04-b33a-59488aaa76c7"
        },
        {
            "requestId": "afa0b15e-1354-4348-8917-8185ef11c2cb",
            "startTime": 1412116933401,
            "href": "https://api.unicornmedia.com/status-api/statuses/afa0b15e-1354-4348-8917-8185ef11c2cb"
        }
    ]
}
```

Checking Notification Statuses

Syntax:

https://api.unicornmedia.com/status-api/notifications/{requestId}

Status Definitions:

Type	Description
PENDING	The activity has been received and scheduled in a queue
PROCESSING	The activity is currently being processed.
COMPLETE	The activity has completed successfully.
FAILED	The activity has encountered a problem and failed

Example: Notification Statuses

Request (GET):

https://api.unicornmedia.com/status-api/notifications/2796350e-2125-4f04-b33a-59488aaa76c7 (about:blank)

Request header

Required HTTP Header:

`X-BC-ONCE-API-KEY : {yourApiKey}`

Response Body

```
{
    "notifications": [{
        "sentTime": 1416417989795,
        "event": "ingest",
        "id": "26",
        "status": "COMPLETE",
        "notification": {
            "notification": "ingest",
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "mediaItem": {
                "id": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "title": "brightcove-once",
                "version": 0,
                "metadata": {}
            },
            "details": {
                "message": "Media Item Ingested."
            }
        },
        "targets": [
            "http://api-endpoint.com"
        ]
    },
    {
        "sentTime": 1416516506046,
        "event": "transcode",
        "id": "77",
        "status": "COMPLETE",
        "notification": {
            "notification": "transcode",
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "mediaItem": {
                "id": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "title": "brightcove-once",
                "version": 0,
                "metadata": {}
            },
            "details": {
                "message": "Rendition Transcoded.",
                "rendition": {
                    "id": "3345fa40-bbb6-11e3-87e6-005056835b09",
                    "name": "Once 3600 MP4 ZC 1280x720 96 29.97 H3_2"
                }
            }
        },
        "targets": [
            "http://api-endpoint.com"
        ]
    },
    {
        "sentTime": 1416516513655,
        "event": "transcode",
        "id": "129",
        "status": "COMPLETE",
        "notification": {
            "notification": "transcode",
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "mediaItem": {
                "id": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "title": "brightcove-once",
                "version": 0,
                "metadata": {}
            },
            "details": {
                "message": "Rendition Transcoded.",
                "rendition": {
                    "id": "8260dd62-b49d-11e3-87e6-005056835b09",
                    "name": "Once 2500 MP4 ZC 960x540 96 29.97 M3_1"
                }
            }
        },
        "targets": [
            "http://api-endpoint.com"
        ]
    },
    {
        "sentTime": 1416516503504,
        "event": "transcode",
        "id": "132",
        "status": "COMPLETE",
        "notification": {
            "notification": "transcode",
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "mediaItem": {
                "id": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "title": "brightcove-once",
                "version": 0,
                "metadata": {}
            },
            "details": {
                "message": "Rendition Transcoded.",
                "rendition": {
                    "id": "825d7a04-b49d-11e3-87e6-005056835b09",
                    "name": "Once 1900 MP4 ZC 960x540 96 29.97 M3_1"
                }
            }
        },
        "targets": [
            "http://api-endpoint.com"
        ]
    }, {
        "sentTime": 1416418159982,
        "event": "publish",
        "id": "151",
        "status": "COMPLETE",
        "notification": {
            "notification": "publish",
            "requestId": "2796350e-2125-4f04-b33a-59488aaa76c7",
            "mediaItem": {
                "id": "2d1d6fe7-6924-4d1d-ba06-d49b2d9f5f4b",
                "foreignKey": "brightcove-once",
                "domainId": "b207b479-c841-4095-8918-978be9e18dee",
                "catalogId": "bc6cb7d4-be99-471b-adf3-7c501172b317",
                "title": "brightcove-once",
                "version": 0,
                "metadata": {}
            },
            "details": {
                "message": "Media Item Published."
            }
        },
        "targets": [
            "http://api-endpoint.com"
        ]
    }]
}
```

Resending Notifications

Syntax:

```
https://api.unicornmedia.com/status-api/notifications/{requestId}/{notificationType}
```

```
https://api.unicornmedia.com/status-api/notifications/{requestId}/{notificationType}?id={event}
```

Example: Resending All Notifications by Type

Request

```
https://api.unicornmedia.com/status-api/notifications/2796350e-2125-4f04-b33a-59488aaa76c7/transcode
```

Request header

Required HTTP Header:

```
X-BC-ONCE-API-KEY : {yourApiKey}
```

Response Body

```
{
    "transcode": [
        {
            "id": "77",
            "submitted": true
        },
        {
            "id": "129",
            "submitted": true
        },
        {
            "id": "132",
            "submitted": true
        }
    ]
}
```

Example: Resending a Single Notification from an Array

Request

`https://api.unicornmedia.com/status-api/notifications/2796350e-2125-4f04-b33a-59488aaa76c7/transcode?id=77`

Request header

Required HTTP Header:

`X-BC-ONCE-API-KEY : {yourApiKey}`

Response

```
{
    "transcode": [
    {
        "id": "77",
        "submitted": true
    }
    ]
}
```

Media Management API

The Media Management API is a RESTful API for accessing and configuring the components of the domain for an authorized user. The multi-faceted and cascading configuration hierarchy gives the domain owner the ability to set-up domain, application, catalog and mediaitem level settings. The authorized user has the ability to create, read, update and delete anything from ad configurations to metadata on independent media items. It provides easy to use methods for fronting UI components in an existing CMS or through a programmatic function.

API Base URL:

https://api.unicornmedia.com/media-management-api/

Request header

Required HTTP Header

`X-BC-ONCE-API-KEY : {yourApiKey}`

Domain

Get Domain details

Returns the essential information of the domain

Endpoint: GET `/domains/{domainId}`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{DomainId}`

or

`https://api.unicornmedia.com/media-management-api/{DomainName}`

Response:

```
{
    "id": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "name": "BrightcoveOnce"
}
```

Get Domain renditions

Returns a collection of the transcode renditions available to the domain.

Endpoint: GET `/domains/{domainId}/renditions`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{DomainId}/renditions`

Response:

```
{
    "results": [
        {
            "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7",
            "name": "OV2 Once 1200 ZC 640x360 1104.96 29.97 B31",
            "default": true
        },
        {
            "id": "468fb310-a585-11e4-bfdb-005056837bc7",
            "name": "OV2 Once 4400 ZC 1280x720 4144.256 29.97 H40",
            "default": true
        },

    ]
}
```

Get Domain rendition details

Returns the settings of the selected transcode rendition indicated by the `renditionId` .

Endpoint: GET `/domains/{domainId}/renditions/{reditionId}`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{DomainId}/renditions/{reditionId}`

Response:

```
{
    "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7",
    "name": "OV2 Once 1200 ZC 640x360 1104.96 29.97 B31",
    "width": 640,
    "height": 360,
    "videoBitRate": 1104,
    "audioBitRate": 96,
    "codecsValue": "mp4a.40.2,avc1.42001f"
}
```

Catalog

Get Catalogs

Returns a collection of all catalogs within the domain. This method fetches 20 catalogs per page, returns the **`totalResult`** to indicate the total number of catalogs and provides previous or next page requests within the body.

Endpoint: GET **`/domains/{domainId}/catalogs`**

Example: GET

`/domains/{domainId}/catalogs`

Response:

```
{
    "results": [
        {
            "id": "9482da98-4ad2-490d-983b-42c17fe06b81",
            "name": "1-2-8",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        },
        {
            "id": "1cd8a599-13b6-45e4-8a94-7bad7a5c457e",
            "name": "New catalog",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        }
    ],
    "prev": null,
    "next": "https://api.unicornmedia.com/media-management-api/domains/4eca7ac5-3954-416d-bb23-e65aa511b85a/cata
    "totalResults": 27
}
```

Get Catalog Details

Returns the essential information of the catalog.

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}`

Response:

```
{
    "id": "62191781-a933-49d6-831f-83bdf51a26ac",
    "name": "New Catalog",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
}
```

Get Catalog Renditions

Returns a collection of the transcode renditions available to the catalog.

Endpoint: GET `/domains/{domainId}/catalogs/{catalogId}/renditions`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/renditions`

Response:

```
{
    "results": [
        {
            "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7",
            "name": "OV2 Once 1200 ZC 640x360 1104.96 29.97 B31"
        },
        {
            "id": "225bd8bb-a577-11e4-bfdb-005056837bc7",
            "name": "OV2 Once 2000 ZC 960x540 1872.128 29.97 M31"
        }
    ]
}
```

Get Catalog Rendition Details

Returns the settings of the selected transcode rendition indicated by the `renditionId`.

Endpoint: GET `/domains/{domainId}/catalogs/{catalogId}/renditions/{renditionId}`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/renditions/{renditionId}`

Response:

```
{
    "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7",
    "name": "OV2 Once 1200 ZC 640x360 1104.96 29.97 B31",
    "width": 640,
    "height": 360,
    "videoBitRate": 1104,
    "audioBitRate": 96,
    "codecsValue": "mp4a.40.2,avc1.42001f"
}
```

Replace Catalog Renditions

Replaces the currently configured catalog transcode renditions with the set defined in the request body.

Note: this method is for replacing. For adding individual renditions please see add catalog renditions.

Endpoint: PUT **`/domains/{domainId}/catalogs/{catalogId}/renditions`**

Example: PUT

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/renditions`

Input:

```
[
    {
        "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7"
    },
    {
        "id": "225bd8bb-a577-11e4-bfdb-005056837bc7"
    }
]
```

Object	Sub-Object	Required	Description
id		Yes	The **`renditionId`** of the desired rendition

Response:

```
[
    {
        "id": "076ea1a2-a35b-11e4-bfdb-005056837bc7"
    },
    {
        "id": "225bd8bb-a577-11e4-bfdb-005056837bc7"
    }
]
```

Add Catalog Renditions

Adds the selected rendition as indicated by the renditionId to the catalog.

Only renditions that are already configured for the domain may be added to a catalog. By default, catalogs will acquire all of the defined **`default`** renditions. If a rendition desired is not configured on the domain, please configure it on the domain then add it to the catalog.

Note: Adding new renditions to a catalog will not cascade to the existing items in the catalog. It will only ensure future items will acquire the new rendition.

Endpoint: POST **`/domains/{domainId}/catalogs/{catalogId}/renditions`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/renditions`

Input:

```
{
    "id": "5ff484d6-a33d-11e4-bfdb-005056837bc7"
}
```

Object	Sub-Object	Required	Description
id		Yes	The **renditionId** of the desired rendition

Response:

```
{
    "id": "5ff484d6-a33d-11e4-bfdb-005056837bc7"
}
```

Delete Catalog Rendition

Deletes the selected rendition indicated by the **renditionId** from the rendition set.

Endpoint: DELETE **/domains/{domainId}/catalogs/{catalogId}/renditions/{renditionId}**

Example: DELETE

https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/renditions/{renditionId}

Response:

```
{
    "id": "5ff484d6-a33d-11e4-bfdb-005056837bc7"
}
```

Create Catalog

Creates a new catalog within the indicated **domainId** . The catalog will acquire all transcode renditions set as default on the domain and the domain publication rules unless the renditions are defined in the input.

Endpoint: POST **/domains/{domainId}/catalogs**

Example: POST

https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs

Input:

```
{
    "name":"New-Test-Catalog",
    "renditions": [
        {
            "id": "5ff484d6-a33d-11e4-bfdb-005056837bc7"
        },
        {
            "id": "ac2e7f0b-a345-11e4-bfdb-005056837bc7"
        }
        }
    ]
}
```

Object	Sub-Object	Required	Description
Name		Yes	The desired name of the catalog
renditions		No	Indicates a collection of **renditionIds** to be defined. If excluded, the catalog will acquire the defaulted domain renditions

	id	Yes	Only declared if you intent to define a sub set of the renditions from the domain

Response:

```
{
    "id": "59dec118-562b-4bc6-b39f-66d2920a913a",
    "name": "New-Test-Catalog",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
}
```

Update Catalog name

Updates the catalog **`name`** in the indicated **`domainId`** by setting its **`catalogId`** with the updated name in the request body.

Note: A **`catalogId`** is system generated and can not be updated.

Endpoint: PUT **`/domains/{domainId}/catalogs`**

Example: PUT

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs`

Input:

```
{
    "id": "59dec118-562b-4bc6-b39f-66d2920a913a",
    "name":"Updated-Test-Catalog-Name"
}
```

Object	Sub-Object	Required	Description
id		Yes	The catalogId to be updated. This system generated value can not be changed
name		Yes	The desired name of the catalog indicated by the **`catalogId`**

Response:

```
{
    "id": "59dec118-562b-4bc6-b39f-66d2920a913a",
    "name": "Updated-Test-Catalog-Name"
}
```

Media Items

Get MediaItems

Returns a collection of all media items within a catalog. This method fetches 20 media items per page, returns the totalResult to indicate the total number of mediaitems and provides the previous or next page requests within the body.

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}/mediaItems`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems`

Response

```
{
    "results": [
        {
            "id": "efe70c1f-ebd2-4c5e-856a-a54a8e97415f",
            "catalogId": "991d5ec7-8eb2-41c9-850c-e5c3411bebd9",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
            "isAd": false,
            "foreignKey": "28-feb-02-updated",
            "title": "28-feb-02",
            "draftVersion": 1,
            "publishedVersion": 1
        },
        {
            "id": "bd0e9dd6-45a2-4f55-95d7-c6ca887a3c9c",
            "catalogId": "991d5ec7-8eb2-41c9-850c-e5c3411bebd9",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
            "isAd": false,
            "foreignKey": "28-feb-01",
            "title": "28-feb-01",
            "draftVersion": 0,
            "publishedVersion": 1
        }
    ],
    "prev": null,
    "next": "https://api.unicornmedia.com/media-management-api/domains/4eca7ac5-3954-416d-bb23-e65aa511b85a/cata
    "totalResults": 28
}
```

Get MediaItem details

Returns the essential information of the media item.

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`

Response

```
{
    "id": "efe70c1f-ebd2-4c5e-856a-a54a8e97415f",
    "catalogId": "991d5ec7-8eb2-41c9-850c-e5c3411bebd9",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "foreignKey": "28-feb-02-updated",
    "title": "28-feb-02",
    "draftVersion": 1,
    "publishedVersion": 1,
    "keywords": [ ],
    "timedText": [ ],
    "cuePoints": [
        {
            "unit": "Seconds",
            "valueIn": 10
        },
        {
            "unit": "Seconds",
            "valueIn": 25
        }
    ],
    "publicationRules": [
        "6169df7e-d023-4daf-90ff-21e290f16162"
    ],
    "metadata": {
        "JobID": "someJobId",
        "PassThruMetadata": "hello world",
        "foo": "bar",
        "hello": "world"
    }
}
```

Update MediaItem

Updates the media item metadata for the indicated mediaItemId. Values defined on update will overwrite any existing values unless otherwise specified.

Note: System generated Id's and versions can not be modified.

Endpoint: POST **`/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`

Input

```
{
    "id": "efe70c1f-ebd2-4c5e-856a-a54a8e97415f",
    "catalogId": "991d5ec7-8eb2-41c9-850c-e5c3411bebd9",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "foreignKey": "Updated-fk",
    "title": "28-feb-02",
    "keywords": [ ],
    "cuePoints": [
        {
            "unit": "Seconds",
            "valueIn": 10
        },
        {
            "unit": "Seconds",
            "valueIn": 25
        }
    ],
    "publicationRules": [
        "6169df7e-d023-4daf-90ff-21e290f16162"
    ],
    "metadata": {
        "JobID": "someJobId",
        "PassThruMetadata": "hello world",
        "foo": "bar",
        "hello": "world"
    }
}
```

Object	Sub-Object	Required	Description
id		Yes	The `mediaItemId` . This value can not be updated.
catalogId		Yes	The `catalogId` . This value can not be updated.
domainId		Yes	The `domainId` . This value can not be updated.
foreignKey		No	The `foreignKey` of the `mediaItemId` should only be provided if intended to be updated.
title		No	Title of the media item. Its value is not required, but recommended for readability.
keywords		No	A comma-delimited list of keywords.
cuePoints		No	A collection of mid-roll ad breaks to be set on the media item.
	unit	Yes	The unit of measure for the cue point value. Currently only "seconds" is supported.
	valueIn	Yes	The value of the desired cue point location.
publicationRules		No	A collection of `publicationRuleIds` to be associated with the media item. Including this field will overwrite any existing publication rules on the media item.
metadata		No	A collection of metadata as key value pairs associated with the media item. Including this field will overwrite any existing metadata defined at ingest.

Response

```
{
    "id": "efe70c1f-ebd2-4c5e-856a-a54a8e97415f",
    "catalogId": "991d5ec7-8eb2-41c9-850c-e5c3411bebd9",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "foreignKey": "Updated-fk",
    "title": "28-feb-02",
    "draftVersion": 1,
    "publishedVersion": 1,
    "keywords": [ ],
    "timedText": [ ],
    "cuePoints": [
        {
            "unit": "Seconds",
            "valueIn": 10
        },
        {
            "unit": "Seconds",
            "valueIn": 25
        }
    ],
    "publicationRules": [
        "6169df7e-d023-4daf-90ff-21e290f16162"
    ],
    "metadata": {
        "JobID": "someJobId",
        "PassThruMetadata": "hello world",
        "foo": "bar",
        "hello": "world"
    }
}
```

Delete MediaItem

Deletes the MediaItem specified by the `mediaItemId`

Endpoint: DELETE `/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}`

Response

```
{
  "delete": "MediaItem: 1eb36535-1105-440e-a722-381c9dcf504d scheduled for deletion"
}
```

Publication Rules

Get Domain publication rules

Returns a collection of the domain `publicationRuleIds` .

Endpoint: GET `/domains/{domainId}/publicationRules`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/publicationRules`

Response

```
[
    "c039f7e3-5b3d-4aec-a8d9-6346ccc57dd5",
    "147184ae-1a51-4a63-bc10-3a9debbe03fa",
    "b44f0463-7c69-45a2-8dfe-aa556c333b74",
    "84658fcc-efe6-4dba-99e9-37601fbcf1c9",
    "671a090b-9cc7-4362-a3b1-4794c1f15326",
    "5d1d5044-05f0-4be2-9dad-323c9c6adf8f",
    "e202e955-6653-4f33-91f0-c004b65c8a1e",
    "a9ff5331-9eb8-45b3-8fc5-2a00bfb84642"
]
```

Get Domain PublicationRule details

Returns the domain publication rule settings of the indicated `publicationRuleId` .

Endpoint: GET `/domains/{domainId}/publicationRules/{publicationRuleId}`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id": "c039f7e3-5b3d-4aec-a8d9-6346ccc57dd5",
    "startDate": 1425776400,
    "endDate": 1741465620,
    "clientFilters": [
        {
            "variableName": "IpAddress",
            "value": "127.0.0.1",
            "filterType": "Equals",
            "isDenied": false
        }
    ],
    "countryRules": [
        {
            "countryCode": "US",
            "isDenied": false
        }
    ]
}
```

Get Catalog publication rules

Returns a collection of the catalog `publicationRuleIds` .

Endpoint: GET `/domains/{domainId}/catalogs/{catalogId}/publicationRules`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/publicationRules`

Response

```
[
    "215c1ab4-2f4c-43b2-b569-4cfd6eb6f432",
    "2ef2157d-95df-4ae7-b3ce-e3bb5beab226",
    "3d514905-f4aa-4ef8-b876-d31b1e35a682",
    "9177495a-1904-4e38-9fcb-2a49cab1f10d",
    "95e29752-d2e5-410c-9061-e188e7c05a30",
    "9c3c10b1-cac9-43c6-bc2a-70ebcbfa421d",
    "ab8ca1de-379d-4dc7-9974-e5cdfab37a8c",
    "ccf41081-d3f6-49b4-8f62-0d742aa91385"
]
```

Get Catalog PublicationRule details

Returns the catalog publication rule settings of the indicated **`publicationRuleId`** .

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}/publicationRules/{publicationRuleId}`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id": "215c1ab4-2f4c-43b2-b569-4cfd6eb6f432",
    "startDate": 1425776400,
    "endDate": 1741465620,
    "clientFilters": [
        {
            "variableName": "IpAddress",
            "value": "127.0.0.1",
            "filterType": "Equals",
            "isDenied": false
        }
    ],
    "countryRules": [
        {
            "countryCode": "US",
            "isDenied": false
        }
    ]
}
```

Get MediaItem publication rules

Returns a collection of publication rules of the specified **`mediaItemId`** .

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules`

Response

```
[
    "13354579-f935-4fbe-bfe5-6c279eea80cc",
    "3ae6e832-d4ac-4755-bf51-9b19f87e05a3",
    "452ef886-3dff-43e5-a9da-6600bc2dee6f",
    "71462fc3-2e2e-4442-8d04-7e9971277fef",
    "923a48f3-a03d-4e5a-93d2-9e5e51f62923",
    "a0580ee2-efed-4419-b07e-3559415fc033",
    "e40f658d-18b2-4d34-9568-18004f8d9697",
    "fcea23dd-83c9-4a64-8f1c-59663694bf38"
]
```

Get MediaItem PublicationRule details

Returns the media item publication rule settings of the indicated **`publicationRuleId`** .

Endpoint: GET **`/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules/{publicationRuleId}`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id": "13354579-f935-4fbe-bfe5-6c279eea80cc",
    "startDate": 1424826000,
    "endDate": 1740535320,
    "clientFilters": [
        {
            "variableName": "IpAddress",
            "value": "127.0.0.1",
            "filterType": "Equals",
            "isDenied": false
        }
    ],
    "countryRules": [ ]
}
```

Create Domain PublicationRule

Creates a new domain publication rule within the indicated **`domainId`** .

Note: The creation of a new domain publication rule does not cascade down to existing catalogs or media items. It will ensure future catalogs will acquire the new publication rule.

Endpoint: POST **`/domains/{domainId}/publicationRules`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/publicationRules`

Input

```
{
    "startDate": 1412538800,
    "endDate": 1838635000,
    "clientFilters": [
        {
            "variableName": "UserAgent",
            "value": "chrome",
            "filterType": "Contains",
            "isDenied": true
        }
    ],
    "countryRules": [
        {
            "countryCode": "IN",
            "isDenied": false
        }
    ]
}
```

Object	Sub-Object	Required	Description
startDate		Yes	Begin content availability in epoch time (seconds)
endDate		Yes	End content availability in epoch time (seconds)
clientFilters		No	Allows the setting of client restrictions based on **`IpAddress`** , **`UserAgent`** , or **`ReferringHost`**
	variableName	Yes	The name of the client filter to key off of. Its value can be **`IpAddress`** , UserAgent or ReferringHost
	value	Yes	The desired value to be associated with the variable
	filterType	Yes	The filter type can contain the following arguements: **`Equals`** , **`NotEquals`** , **`In`** , **`NotIn`** , **`Contains`** , NotContains, **`StartsWith`** , **`NotStartsWith`** , **`EndsWith`** , or **`NotEndsWith`**
	isDenied	Yes	A boolean value of true or false for the argument
countryRules		No	A collection of country codes to be granted access or denied
	countryCode	Yes	Geo restriction based on ISO 639-1 country codes
	isDenied	Yes	A boolean value of true or false for the argument

Response

```
{
    "id": "d5c8dea0-fae7-46b2-8cfd-a5db5ac24a28"
}
```

Create Catalog PublicationRule

Creates a new catalog publication rule within the indicated **`catalogId`** .

Note: The creation of a new catalog publication rule does not cascade down to existing media items. It will ensure future media items ingested to the catalog will acquire the new publication rule.

Endpoint: POST **`/domains/{domainId}/catalogs/{catalogId}/publicationRules`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/publicationRules`

Input

```
{
    "startDate": 1412538800,
    "endDate": 1838635000,
    "clientFilters": [
        {
            "variableName": "UserAgent",
            "value": "chrome",
            "filterType": "Contains",
            "isDenied": true
        }
    ],
    "countryRules": [
        {
            "countryCode": "IN",
            "isDenied": false
        }
    ]
}
```

Object	Sub-Object	Required	Description
startDate		Yes	Begin content availability in epoch time (seconds)
endDate		Yes	End content availability in epoch time (seconds)
clientFilters		No	Allows the setting of client restrictions based on `IpAddress` , `UserAgent` , or `ReferringHost`
	variableName	Yes	The name of the client filter to key off of. Its value can be `IpAddress` , UserAgent or ReferringHost
	value	Yes	The desired value to be associated with the variable
	filterType	Yes	The filter type can contain the following arguements: `Equals` , `NotEquals` , `In` , `NotIn` , `Contains` , NotContains, `StartsWith` , `NotStartsWith` , `EndsWith` , or `NotEndsWith`
	isDenied	Yes	A boolean value of true or false for the argument
countryRules		No	A collection of country codes to be granted access or denied
	countryCode	Yes	Geo restriction based on ISO 639-1 country codes
	isDenied	Yes	A boolean value of true or false for the argument

Response

```
{
    "id": "d5c8dea0-fae7-46b2-8cfd-a5db5ac24a28"
}
```

Create MediaItem PublicationRule

Creates a new publication rule for the indicated `mediaItemId` .

Endpoint: POST `/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules`

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules`

Input

```
{
    "startDate": 1412538800,
    "endDate": 1838635000,
    "clientFilters": [
        {
            "variableName": "UserAgent",
            "value": "chrome",
            "filterType": "Contains",
            "isDenied": true
        }
    ],
    "countryRules": [
        {
            "countryCode": "IN",
            "isDenied": false
        }
    ]
}
```

Object	Sub-Object	Required	Description
startDate		Yes	Begin content availability in epoch time (seconds)
endDate		Yes	End content availability in epoch time (seconds)
clientFilters		No	Allows the setting of client restrictions based on `IpAddress` , `UserAgent` , or `ReferringHost`
	variableName	Yes	The name of the client filter to key off of. Its value can be `IpAddress` , UserAgent or ReferringHost
	value	Yes	The desired value to be associated with the variable
	filterType	Yes	The filter type can contain the following arguements: `Equals` , `NotEquals` , `In` , `NotIn` , `Contains` , NotContains, `StartsWith` , `NotStartsWith` , `EndsWith` , or `NotEndsWith`
	isDenied	Yes	A boolean value of true or false for the argument
countryRules		No	A collection of country codes to be granted access or denied
	countryCode	Yes	Geo restriction based on ISO 639-1 country codes
	isDenied	Yes	A boolean value of true or false for the argument

Response

```
{
    "id": "d5c8dea0-fae7-46b2-8cfd-a5db5ac24a28"
}
```

Delete Domain PublicationRule

Deletes the publication rule for the domain as indicated by the `publicationRuleId` .

Note: Deleting a publication rule from the domain does not cascade down to existing catalogs and media items. Any item that has already acquired the rule will not be modified. It will ensure future catalogs do not acquire this rule.

Endpoint: DELETE `/domains/{domainId}/publicationRules/{publicationRuleId}`

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id": "c039f7e3-5b3d-4aec-a8d9-6346ccc57dd5"
}
```

Delete Catalog PublicationRule

Deletes the specified publication rule for the catalog as indicated by the **`catalogId`** .

Note: Deleting a publication rule from the catalog does not cascade down to existing media items. Any item that has already acquired the rule will not be modified. It will ensure future media items in the catalog do not acquire this rule.

Endpoint: DELETE **`/domains/{domainId}/catalogs/{catalogId}/publicationRules/{publicationRuleId}`**

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id" : "215c1ab4-2f4c-43b2-b569-4cfd6eb6f432"
}
```

Delete MediaItem PublicationRule

Deletes the specified publication rule for the mediaItem as indicated by the **`mediaItemId`** .

Endpoint: DELETE **`/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules/{publicationRuleId}`**

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/catalogs/{catalogId}/mediaItems/{mediaItemId}/publicationRules/{publicationRuleId}`

Response

```
{
    "id":"13354579-f935-4fbe-bfe5-6c279eea80cc"
}
```

Applications

Get Applications

Returns a collection of all applications within a domain. This method fetches 20 applications per page, returns the **`totalResult`** to indicate the total number of media items and provides the previous or next page requests within the body.

Endpoint: GET **`/domains/{domainId}/applications`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications`

Response

```
{
    "results": [
        {
            "id": "79e36ed1-4f2d-4f03-9b49-23d19d34e971",
            "name": "New Player Template",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        },
        {
            "id": "7aa9fcfe-ea9d-449c-80b6-fd5e7b7a4b72",
            "name": "Player- Single and Multi Ads",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        },
        {
            "id": "fbfe16ab-0d11-4ca1-b944-1480970c4bf9",
            "name": "Player- Multi Ads",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        }
    ],
    "prev": null,
    "next": null,
    "totalResults": 5
}
```

Get Application details

Returns the essential information of the application including the ad configurations. The response will include individual ad slots "pre-roll", "mid-roll", and/or "post-roll" for direct VAST configurations and a single "ad" server configurations that have ad break information determined by the ad decisioning server (e.g. Freewheel SmartXML, DFP ad rules).

Endpoint: GET **`/domains/{domainId}/applications/{applicationId}`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}`

Response

```
{
    "id": "7aa9fcfe-ea9d-449c-80b6-fd5e7b7a4b72",
    "name": "Player- Single and Multi Ads",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "adConfig": {
        "preRoll": {
            "id": "5840b954-5094-4e13-b49a-319115342011",
            "name": "prenewUpdated",
            "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
            "adServerName": "Fireworks waterfall",
            "adPosition": "preroll"
        }
    }
}
```

Note: **`adConfig`** can be of type single or multi.

Single adConfig example:

```
"adConfig": {
    "preRoll": {
        "id": "5840b954-5094-4e13-b49a-319115342011",
        "name": "prenewUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "preroll"
    },
    "midRoll": {
        "id": "5840b954-5094-4e13-b49a-319115342011",
        "name": "Mid",
        "adServerId": "1111-2222-33333-4444-aaaabbbbccccddd",
        "adServerName": "Test Ad Server",
        "adPosition": "midroll"
    },
    "postRoll": {
        "id": "5840b954-5094-4e13-b49a-319115342011",
        "name": "Post",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "postroll"
    }
}
```

Multi adConfig example:

```
"adConfig": {
    "ad": {
        "id": "5840b954-5094-4e13-b49a-319115342011",
        "name": "DFP",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "dfp"
    }
}
```

Get Application ad configurations

Returns a collection of ad configurations on the indicated applicationId. The response will include individual ad slots "pre-roll", "mid-roll", and/or "post-roll" for direct VAST configurations and a single "ad" server configurations that have ad break information determined by the ad decisioning server (e.g. Freewheel SmartXML, DFP ad rules).

Endpoint: GET **`domains/{domainId}/applications/{applicationId}/adConfig`**

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}/adConfig`

Response

```
{
    "preRoll": {
        "id": "6a38cbc1-7257-4a9b-99fa-121d4f971768",
        "name": "pre",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "preroll"
    },
    "midRoll": {
        "id": "9989ad10-62a6-4f82-828d-d63ed464a33e",
        "name": "mid",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "midroll"
    },
    "postRoll": {
        "id": "c244fd41-7188-41a4-ba5f-2f6edd6a67fa",
        "name": "postnew",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "postroll"
    }
}
```

Add or Replace adConfig on an Application

Adds or Replaces the currently configured adconfigs. The input adconfig can be individual slots "pre-roll", "mid-roll", and/or "post-roll" for direct VAST campaigns OR a single "ad" campaign to indicate the ad decisioning server will determine the ads breaks (e.g. Freewheel SmartXML, DFP ad rules).

Endpoint: POST **`domains/{domainId}/applications/{applicationId}/adConfig`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}/adConfig`

Input (direct VAST):

```
{
    "preRoll": {
        "name": "preRollUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d"
    },
    "postRoll": {
        "name": "postRollUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d"
    }
}
```

Input (smartxml or dfp):

```
{
    "ad": {
        "name": "postRollUpdated",
        "adPosition": "smartxml",
        "adServerId": "72b453a5-37b5-5fa1-6deb-2958363ade2"
    }
}
```

Object	Sub-Object	Required	Description

preRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The **`adServerId`** intended to be used with this slot
midRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The **`adServerId`** intended to be used with this slot
postRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The **`adServerId`** intended to be used with this slot
ad		No	Only used for Freewheel SmartXML or DFP ad rules where ad slots are defined by the ad decisioning server. Can not be used with defined prerolls, midrolls, and/or postrolls
	name	Yes	An arbitrary name for user readability purposes only
	adPosition	Yes	Must be "smartxml" or "dfp"
	adServerId	Yes	The **`adServerId`** intended to be used with this slot

Response

```
{
    "id": "7aa9fcfe-ea9d-449c-80b6-fd5e7b7a4b72",
    "name": "Player- Single and Multi Ads",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "adConfig": {
        "postRoll": {
        "id": "b04116aa-5656-45e3-adc3-c2eda4c6127e",
        "name": "prenewUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "postroll"
        }
    }
}
```

Create Application

Creates a new application within the indicated domainId.

Endpoint: POST **`/domains/{domainId}/applications`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications`

Input

```
{
    "name":"New_Application",
    "preRollAdCount":1,
    "midRollAdCount":2,
    "postRollAdCount":3
}
```

	Sub-		

Object	Object	Required	Description
name		Yes	Title of the application for readability
preRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in the slot. By default this value is set to 0, but is required when using direct VAST tags.
midRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in each slot. By default this value is set to 0, but is required when using direct VAST tags.
postRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in the slot. By default this value is set to 0, but is required when using direct VAST tags.

Response

```
{
  "id": "a0f7465b-6357-469f-98cc-3f9d8cb7987f",
  "name": "New_Application",
  "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
  "preRollAdCount": 1,
  "midRollAdCount": 2,
  "postRollAdCount": 3
}
```

Update Application

Updates the standard settings of the application.

Endpoint: POST **`/domains/{domainId}/applications/{applicationId}`**

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}`

Input

```
{
    "name":"New_Application",
    "preRollAdCount":1,
    "midRollAdCount":2,
    "postRollAdCount":3
}
```

Object	Sub-Object	Required	Description
name		Yes	Title of the application for readability
preRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in the slot. By default this value is set to 0, but is required when using direct VAST tags.
midRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in each slot. By default this value is set to 0, but is required when using direct VAST tags.
postRollAdCount		No	Only used for direct VAST campaigns. Its value will set the number of ads to be returned in the slot. By default this value is set to 0, but is required when using direct VAST tags.

Response

```
{
  "id": "a0f7465b-6357-469f-98cc-3f9d8cb7987f",
  "name": "Newest_Application",
  "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
  "preRollAdCount": 1,
  "midRollAdCount": 2,
  "postRollAdCount": 3
}
```

Get Application Ad Configurations

Returns a collection of ad configurations on the indicated applicationId. The response will include individual ad slots "pre-roll", "mid-roll", and/or "post-roll" for direct VAST configurations and a single "ad" server configurations that have ad break information determined by the ad decisioning server (e.g. Freewheel SmartXML, DFP ad rules).

Endpoint: GET `/domains/{domainId}/applications/{applicationId}/adConfig`

Example: GET

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}/adConfig`

Response

```
{
    "preRoll": {
        "id": "6a38cbc1-7257-4a9b-99fa-121d4f971768",
        "name": "pre",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "preroll"
    },
    "midRoll": {
        "id": "9989ad10-62a6-4f82-828d-d63ed464a33e",
        "name": "mid",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "midroll"
    },
    "postRoll": {
        "id": "c244fd41-7188-41a4-ba5f-2f6edd6a67fa",
        "name": "postnew",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "postroll"
    }
}
```

Add or replace adConfig on an Application

Adds or Replaces the currently configured adconfigs. The input adconfig can be individual slots "pre-roll", "mid-roll", and/or "post-roll" for direct VAST campaigns OR a single "ad" campaign to indicate the ad decisioning server will determine the ads breaks (e.g. Freewheel SmartXML, DFP ad rules).

Endpoint: POST `/domains/{domainId}/applications/{applicationId}/adConfig`

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}/adConfig`

Input (direct VAST)

```
{
    "preRoll": {
        "name": "preRollUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d"
    },
    "postRoll": {
        "name": "postRollUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d"
    }
}
```

Input (smartxml or dfp)

```
{
    "ad": {
        "name": "postRollUpdated",
        "adPosition": "smartxml",
        "adServerId": "72b453a5-37b5-5fa1-6deb-2958363ade2"
    }
}
```

Object	Sub-Object	Required	Description
preRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns.
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The adServerId intended to be used with this slot
midRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns.
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The adServerId intended to be used with this slot
postRoll		No	The ad slot intended to be updated or replaced. This should only be used with direct VAST campaigns.
	name	Yes	An arbitrary name for user readability purposes only
	adServerId	Yes	The adServerId intended to be used with this slot
ad		no	Only used for Freewheel SmartXML or DFP ad rules where ad slots are defined by the ad decisioning server. Can not be used with defined prerolls, midrolls, and/or postrolls
	name	Yes	An arbitrary name for user readability purposes only
	adPosition	Yes	Must be "smartxml" or "dfp"
	adServerId	Yes	The adServerId intended to be used with this slot

Response

```
{
    "id": "7aa9fcfe-ea9d-449c-80b6-fd5e7b7a4b72",
    "name": "Player- Single and Multi Ads",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a",
    "adConfig": {
        "postRoll": {
        "id": "b04116aa-5656-45e3-adc3-c2eda4c6127e",
        "name": "prenewUpdated",
        "adServerId": "34b353d8-f7d5-4fa1-9dea-49183236df8d",
        "adServerName": "Fireworks waterfall",
        "adPosition": "postroll"
        }
    }
}
```

Delete Application

Deletes the application from the indicated **`domainId`** .

Endpoint: DELETE **`/domains/{domainId}/applications/{applicationId}`**

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}`

Response

```
{
    "id": "7aa9fcfe-ea9d-449c-80b6-fd5e7b7a4b72"
}
```

Delete AdConfig

Deletes all of the ad configurations from the indicated **`applicationId`** .

Endpoint: DELETE **`/domains/{domainId}/applications/{applicationId}/adConfig`**

Example: DELETE

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/applications/{applicationId}/adConfig`

Response

```
{
    "id": [
        "6a38cbc1-7257-4a9b-99fa-121d4f971768",
        "9989ad10-62a6-4f82-828d-d63ed464a33e",
        "c244fd41-7188-41a4-ba5f-2f6edd6a67fa"
    ]
}
```

Ad Decisioning Server

Get Ad Decisioning Servers

Returns a collection of ad decisioning servers configured for the domainId. This method fetches 20 ad servers per page, returns the **`totalResult`** to indicate the total number of ad servers and provides the previous or next page requests within the body.

Endpoint: GET **/domains/{domainId}/adServers**

Example: GET

https://api.unicornmedia.com/media-management-api/domains/{domainId}/adServers

Response

```
{
    "results": [
        {
            "id": "4df36e4c-f418-403d-b17e-60c99dbe65b4",
            "name": "Newest Ad Server",
            "baseUrl": "test.com",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        },
        {
            "id": "06790067-e7fa-4e90-8d65-6ed4d77b41b7",
            "name": "New Ad Provider",
            "baseUrl": "test.com",
            "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
        }
    ],
    "prev": null,
    "next": null,
    "totalResults": 6
}
```

Get Ad Decisioning Server details

Returns the essential information for the indicated **adServerId** .

Endpoint: GET **/domains/{domainId}/adServers/{adServerId}**

Example: GET

https://api.unicornmedia.com/media-management-api/domains/{domainId}/adServers/{adServerId}

Response

```
{
    "id": "4df36e4c-f418-403d-b17e-60c99dbe65b4",
    "name": "Newest Ad Server",
    "baseUrl": "test.com",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
}
```

Create an Ad Decisioning Server

Creates a new ad decisioning server in the indicated domain by providing an user-defined name and the URL of the server.

Endpoint: POST **/domains/{domainId}/adServers**

Example: POST

https://api.unicornmedia.com/media-management-api/domains/{domainId}/adServers

Input:

```
{
    "name" : "New-ad-server",
    "baseUrl": "www.google.com"
    "variables": {
    "{{placeholder1}} key1": "value1",
        "{{ph2}} key2": "value2",
        "UMADPARAM key3": "value3",
        "key4": "value4"
        }
}
```

Object	Sub-Object	Required	Description
Name		Yes	The name of the ad server for readability purposes
baseURL		Yes	The HTTP request URL of the ad decisioning server
variables		No	Key-value pair mappings for the ad server URL that can utilize page level parameters passed on the query string(see Parameters section) or Dynamic Ad Parameters (see Dynamic Ad Params section). Double curly bracket syntax in the "key" will not be used and defines a placeholder for the insertion of the key-value pair within the baseUrl. See the example and full description below

Response:

```
{
    "id": "40c11e99-8878-4f17-81c8-ca7d16d9ebbe",
    "name": "New-ad-server",
    "baseUrl": "www.google.com",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
    "variables": {
    "{{placeholder1}} key1": "value1",
        "{{ph2}} key2": "value2",
        "{{p3}} UMADPARAM": "UMADPARAMvalue3",
        "key4": "value4"
        }
}
```

The curly brace syntax indicates to the Once system to manipulate the values at runtime. When used within a "key" it indicates where the key-value pair should be placed on the ad server call. The value within the curly braces are user defined and should be matched with the same value configured on the base URL.

When used as a "value," it indicates the actual value should be dynamically generated based on the information collected from the user (see Dynamic Ad Params) or metadata stored on the media item at ingest. When defining page level parameters to be appended, the UMADPARAM should be used as the "key" and the value should include `UMADAPARM{key}` to indicate to the Once system to parse out the key from the query string and use the value passed in.

It is recommended to leave any static variables on the the base Url for readability purposes and use placeholders only when the ordering of the parameters are required. When used without placeholders, the parameters will be appended to the base URL in alphabetical order based on the key.

Example

Base URL for the ad call:

```
http://www.brightcove.com?{{placeholder1}}&{{ph2}}&staticvariable=123&{{p3}}
```

Configured variables:

```
"{{placeholder1}} key1": "{{ipaddress}}",
"{{ph2}} key2": "value2",
"{{p3}} UMADPARAM": "UMADPARAMkey3",
"key4": "value4"
```

Once URL request:

```
https://once.../content.once?UMADPARAMkey3=value3
```

Resulting ad call:

```
http://www.brightcove.com?key1=192.168.1.1&key2=value2&staticvariable=123&key3=value3&key4=value4
```

Update an Ad Decisioning Server

Updates the name of the ad decisioning server indicated by the adServerId.

Endpoint: POST `/domains/{domainId}/adServers/{adServerId}`

Example: POST

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/adServers/{adServerId}`

Input:

```
{
    "name" : "New-ad-server-updated-name",
    "baseUrl": "www.test.com"
    "variables": {
        "key1": "value1",
        "someKey": "someValue"
        }
}
```

Object	Sub-Object	Required	Description
Name		Yes	The name of the ad server for readability purposes
baseURL		Yes	The HTTP request URL of the ad decisioning server
variables		No	Key-value pair mappings for the ad server URL that can utilize page level parameters passed on the query string(see Parameters section) or Dynamic Ad Parameters (see Dynamic Ad Params section). Double curly bracket syntax in the "key" will not be used and defines a placeholder for the insertion of the key-value pair within the baseUrl. See the example and full description under Create an ad decisioning server above

Response:

```
{
    "id": "40c11e99-8878-4f17-81c8-ca7d16d9ebbe",
    "name": "New-ad-server-updated-name",
    "baseUrl": "www.test.com",
    "domainId": "4eca7ac5-3954-416d-bb23-e65aa511b85a"
    "variables": {
        "key1": "value1",
        "someKey": "someValue"
        }
}
```

Delete an Ad Decisioning Server

Deletes an ad decisioning server in the indicated by the adServerId.

Endpoint: DELETE **`/domains/{domainId}/adServers/domains/{domainId}/adServers/{adServerId}`**

`https://api.unicornmedia.com/media-management-api/domains/{domainId}/adServers/{adServerId}`

Response:

```
{
    "id": "4df36e4c-f418-403d-b17e-60c99dbe65b4",
}
```

Once URL Creation

Brightcove Once VOD uses a proprietary URL syntax. You configure this URL through a programmatic URL generation method that you customize for your particular business needs. This URL employs the Brightcove Once platform to detect viewer devices, distribute content and insert and target ads according to your specifications.

There are two APIs for requesting content to be played back as either VOD or UX. Once UX is an add-on for Once VOD for defining the VOD stream by normalizing Ad Server requests into a VAST 3.0 and embedded the information into a VMAP object. The stream definition includes, stream timeline, stream duration, thidy-party audience measurement information, ad breaks, normalized ad metadata, closed captioning and subtitles.

If content is being delivered from secure pages the request can be made to the same endpoint with HTTPS protocol to ensure secure delivery of the content using a single virtual host.

URL Syntax

The Once VOD URL is constructed by using a programmatic URL syntax. This section will cover the URL structures and the available parameters.

API hostname:

`http://once.unicornmedia.com`

Syntax:

`{Protocol}://{Host}/{Service}/{DeliveryType}/{RequestedFileType}/{domainId}/{applicationId}/{mediaItemId}/{Vi`

Example:

`http://once.unicornmedia.com/now/od/auto/c6589dd5-8f31-4ae3-8a5f-a54ca3d7c973/5291cc2e-c90c-4f05-889e-af8067f8c1ec/31d1ef83-70f4-4de8-9f4e-2f3608d25517/content.once`

URL Variables

Parameter	Type	Description
Protocol	String	HTTP or HTTPS for delivery from secure pages
Host	String	"once.unicornmedia.com" or CNAME pointing to "once.unicornmedia.com"
Service	String	"now" is the only value currently available
DeliveryType	String	Used to determine video delivery behavior. Please see Output Formats for more information.
RequestedFileType	String	Used to determine video delivery behavior. Please see Output Formats for more information.

DomainId	**String**	**The Id of the Customer Domain in the Brightcove system obtained by contacting your Brightcove representative.**
ApplicationId	**String**	**The Id for the application in which to record metrics and enforce business rules obtained through your Brightcove representative.**
MediaItemId or ForeignKey	**String**	**The `MediaItemId` or `ForeignKey` for the base asset being requested.**
VirtualFileName	**String**	**Arbitrary file name for the request.** • **The value is a preference; however, the file extension should accurately represent the requested file type** • **Regardless of the file type, the system will return a media file** • **May help you identify the delivery format and play back properly** Extensions are determined by the output format used in your request. Please see Output Formats for more information.

Output formats

Utilizing various combinations of delivery type, file type, and extension in your Once URLs, the response can be modified to suit various environments as needed. This behavior can either be handled completely by Once's automatic decision logic, or can be be narrowed to respond with only adaptive or progressive formats. The URL variables outlined in the following sections are the only supported combinations for each delivery method.

Auto

To utilize the Once platform's innate device detection to determine proper delivery and file type for the requesting client device, use the combination for Auto delivery. This device detection will determine a number of different playback variables depending on the requesting device including optimal file format, video quality, and for HLS, segment length. The Once platform's Auto detection will prioritize adaptive delivery to a device if supported, otherwise the device will receive an appropriate progressive response.

DeliveryType	FileType	Extension
od	**auto**	**once**

URL map

```
Host/Service/DeliveryType/FileType/DomainId/ApplicationId/MediaItemId/VirtualFileName.Extension
```

Example

```
http://.../od/auto/4989f8db-9187-49ca-86ff-e2cd5342cd4/35217a08-d9db-469f-aec0-
7c88406a0c60/63598604-cb88-44f8-bc47-d207af3b7b4a/content.once
```

Adaptive

The Once platform utilizes the HTTP Live Streaming (HLS) protocol for adaptive, multi-bitrate playback to devices that support this method. Requesting an adaptive response will force an HLS response with discontinuities between content switching and keeps the Once platform's logic in place for determining remaining device specific parameters including the range of profiles to be returned and segment length. If you wish to tailor the profiles returned on a request, you can denote those specific profiles by using the additional query string parameters: `umoprofiles` or `umaprofiles` .

DeliveryType	FileType	Extension
master	playlist	m3u8

URL map

```
Host/Service/DeliveryType/FileType/DomainId/ApplicationId/MediaItemId/VirtualFileName.Extension
```

Example

```
http://.../master/playlist/4989f8db-9187-49ca-86ff-e2cd5342cd4/35217a08-d9db-469f-aec0-7c88406a0c60/63598604-cb88-44f8-bc47-d207af3b7b4a/content.m3u8
```

Continuous Adaptive (ctAdaptive)

Similar to an adaptive request, the `ctadaptive` response will force a HLS response with the discontinuities removed to create a single linear stream playable on even the most simple implementations of HLS devices (e.g. native Android players). We recommend using a regular adaptive stream where possible for the best cache efficiency, but `ctadaptive` will solve difficult platforms and the majority of devices or player implementations for HLS. If you wish to tailor the profiles returned on a request, you can denote those specific profiles by using the additional query string parameters: `umoprofiles` or `umaprofiles` .

DeliveryType	FileType	Extension
master	ct-playlist	m3u8

URL map

```
Host/Service/DeliveryType/FileType/DomainId/ApplicationId/MediaItemId/VirtualFileName.Extension
```

Example

```
http://.../master/ct-playlist/4989f8db-9187-49ca-86ff-e2cd5342cd4/35217a08-d9db-469f-aec0-7c88406a0c60/63598604-cb88-44f8-bc47-d207af3b7b4a/content.m3u8
```

Progressive

If the desired or required video output is a progressive download in an mp4 format, the stitched combination should be used. When requesting a progressive playback experience, a `renditionId` of the requested quality must be included in the URL for successful playback. The placement of the `renditionId` in the Once URL is immediately following the ApplicationId, as shown in the example below.

DeliveryType	FileType	Extension
media	progressive	mp4

URL map

```
Host/Service/DeliveryType/FileType/DomainId/ApplicationId/RenditionId/MediaItemId/VirtualFileName.Extension
```

Example

```
http://.../media/progressive/4989f8db-9187-49ca-86ff-e2cd5342cd4/35217a08-d9db-469f-aec0-7c88406a0c60/468fb310-a585-11e4-bfdb-005056837bc7/63598604-cb88-44f8-bc47-d207af3b7b4a/content.mp4
```

Foreign key use

The Brightcove Once platform allows for the use of foreign key values to be used instead of system generated MediaItemIds within the URL construct to identify the video for playback. This allows for the flexibility of using existing unique ID values associated to customer videos to be reused. **`Foreignkey`** values must be unique to each video item and cannot be used multiple times in the same Domain.

Example MediaItemId

```
63598604-cb88-44f8-bc47-d207af3b7b4a
```

Example Url using MediaItemId

```
http://once.unicornmedia.com/.../63598604-cb88-44f8-bc47-d207af3b7b4a/content.once
```

Example Foreign Key

```
video-co-ca-2113
```

Example URL using Foreign Key

```
http://once.unicornmedia.com/.../video-co-ca-2113/content.once
```

Note: for foreign keys that have special characters such as colons, please see the **`UMFKEY`** (http://docs.brightcove.com/en/once/guides/getting-started-once.html#UMFKEY) query string parameter.

Parameters

Query string

Description

Query string parameters can be used to modify the standard Once URL to append page level data for ad or beacon requests, modify response behavior, or other special use cases as needed.

Ad params

`UMADPARAM` - Specify the Ads Served Back by Ad Provider

```
UMADPARAM{Key}={Value}
```

Use this parameter to pass along page level and client side data to the Once platform to be appended to ad calls made to your ad provider, enabling this data to be used in their ad targeting/decisioning.

Example Key/Value pairs:

- `cc=US`
- `state=Arizona`

Customer Created URL:

```
http://once.unicornmedia.com/.../content.m3u8?UMADPARAMcc=US&UMADPARAMstate=Arizona
```

Brightcove consumes this Once VOD URL, removes the **`UMADPARAM`** key designator, and serves the resulting key-value pair to the ad provider who in turn can utilize that data to respond with the appropriate ad creative based on their decisioning and logic.

URL Sent by Brightcove Once to ad provider:

```
http://ad.provider.url?cc=US&state=Arizona
```

The Once VOD URL example above would make the ad call and return ads specific to the country and state values provided.

UMPTPARAM - Pass Values Through with No System Manipulation

```
UMPTPARAM{Key}={Value}
```

This is used to pass key/value pairs into a call to an ad provider when the value should not be URL decoded by the Once service. In contrast to **`UMADPARAM`** , this parameter lets you pass a string directly through the system without manipulation by Brightcove. For example, you have the ability to pass a string value of comma delimited key-value pairs that require URL Encoding.

Example Key/Value pairs:

- **`Cust_params=Denver%Android_4_0%26tile%Video`**
- **`Video_param=26iu%3D%252F7646`**

Customer Created URL:

```
http://once.unicornmedia.com/.../content.once?&UMPTPARAMCust_Params=Denver%Android_4_0%26
tile%Video&UMPTPARAMVideo_param=26iu%3D%252F7646
```

URL Sent by Brightcove to ad provider:

```
http://ad.provider.url?Cust_Params=Denver%Android_4_0%26tile%Video&Video_param=26iu%3D%252F7646
```

`UMLITERAL` - Pass Set Number of Characters Right of '&' as One String Value

```
UMLITERAL{Key},{Integer}&{Value}
```

This is used to collectively set a defined number of characters right of a specified '&' query string delimiter as one value for inclusion in the Ad Provider URL. **`UMLITERAL`** can be utilized to include ad targeting data that may otherwise be unsafe to include in a URL such as semi-colons. The example below assumes you are passing an 81-character string through as one value.

Example of string value:

```
slau=midroll&tpos=180;;ptgt=a&slau=midroll&tpos=360;&ptgt=a&slau=midroll&tpos=540
```

Number of Characters in String:

81

Customer Created URL:

```
http://once.unicornmedia.com.../VirtualFileName?
UMLITERALmidrolls,81&slau=midroll&tpos=180;;ptgt=a&slau=midroll&tpos=360;;ptgt=a&slau=midroll&tpos=540
```

URL Sent by Brightcove Once to ad provider:

```
http://ad.provider.url?
slau=midroll&tpos=180;;ptgt=a&slau=midroll&tpos=360;;ptgt=a&slau=midroll&tpos=540
```

Beacon parameters

`UMBEPARAM`

Similar to the use of `UMADPARAM` , the `UMBEPARAM` parameter can be utilized to pass along page level and client side data to the Once platform to instead be appended to third party audience measurement URLs made to an external analytics provider, enabling this data to be used in the tracking of a number of playback events. For more information please see the Beaconing section (http://docs.brightcove.com/en/once/guides/getting-started-once.html#beacons).

Example Key/Value pairs:

- `Localadid=55`
- `Uniqueid=country`

Customer Created URL:

`http://once.unicornmedia.com/.../content.once?UMBEPARAMLocaladid=55&UMBEPARAMuniqueid=country`

Parameter Examples:

- `UMBEPARAMLocaladid=55`
- `UMBEPARAMuniqueid=country`

Fired Beaconing URL:

`http://tracking.exampleurl?Localadid=55&uniqueid=$country`

UMXBEACON

`UMXBEACON` enables the specification of a standalone URL to be called to at play start that can be used independently or in addition to pre-configured beacons. The URL to beacon to must be encoded and if other query strings are utilized, `UMXBEACON` must be the last appended to the URL.

Independent Tracking

`UMXBEACON={encodedURL}`

Example:

`http://once.unicornmedia.com/.../content.once?UMXBEACON=http%3A%2F%2Ftracking.exampleurl%3Fkeyone%3D111`

Fired beaconing URL:

`http://tracking.exampleurl?keyone=111`

Utility params

`UMFKEY` - Alternative method of requesting videos by Foreign Key

`UMFKEY={URLEncodedForeignKey}`

`UKFKEY` allows for requesting Once URLs utilizing the Foreign Key as a query string parameter rather than in the URL path. This function allows for the inclusion of characters that would otherwise be disallowed in the URL path, including potentially unsafe characters. Any value used with `UMFKEY` must be appended to the Once URL encoded. If present, `UMFKEY` will take priority over the standard `MediaItemId` in the URL path and a static value should be used in place of the `MediaItemId` .

Example Key/Value pairs:

- `ForeignKey=uma:video:movie.com:480337`

Example Static MediaItemId

`00000000-0000-0000-0000-000000000000`

Customer Created URL:

`http://once.unicornmedia.com/.../00000000-0000-0000-0000-000000000000/content.once?UMFKEY=uma%3Avideo%3Amovie.com%3A480337`

NOTE: The **`UMFKEY`** value must be URL-encoded.

Modifying Delivery Renditions

The robust Once device detection logic caters to specific devices based on device capabilities and hardware, but also allows users to bypass the decisioning logic. There are two methods used to override or add standard renditions at run-time per the query string. When appended as a query string to the Once URL, the associated profile(s) will be dynamically inserted in the returned m3u8 to temporarily override or add the standard rendition list for adaptive playback. The value can be a single RenditionId or BrevityId, a comma-delimited list of Ids, or utilize pre-defined profile setIds (see Default profile sets for umo/umaprofiles). All responses containing overridden profiles will continue to utilize the Once platform's device detection logic to determine the order of renditions and segment size that is most suitable for the requesting device.

Temporarily Override the Standard Rendition List (umoprofiles)

Using **`UMOPROFILES={RenditionId}`** or **`UMOPROFILES={BrevityId}`** or **`UMOPROFILES={profile}`** query string parameter will override any renditions that have been selected by the Once Device Detection logic.

URL Map:

```
/.../MediaItemId/VirtualFileName?umoprofiles=Id1,Id2,...,Id(n)
```

Examples:

```
http://once.unicornmedia.com/.../content.m3u8?umoprofiles=74ba4d0b-a347-11e4-bfdb-005056837bc7,sd1200
```

```
http://once.unicornmedia.com/.../content.m3u8?umoprofiles=720p,1080p
```

```
http://once.unicornmedia.com/.../content.m3u8?umoprofiles=hd6500,hd4400
```

Temporarily Add Renditions to the Standard Rendition List (umaprofiles)

Using **`UMAPROFILES={RenditionId}`** or **`UMAPROFILES={BrevityId}`** or **`UMAPROFILES={profile}`** query string parameter will add the defined renditions to the rendition set selected by the Once Device Detection logic.

URL Map:

```
/.../MediaItemId/VirtualFileName?umaprofiles=Id1,Id2,...,Id(n)
```

Example:

```
http://once.unicornmedia.com/.../content.m3u8?umaprofiles=74ba4d0b-a347-11e4-bfdb-005056837bc7,sd2000
```

```
http://once.unicornmedia.com/.../content.m3u8?umaprofiles=1080p
```

Default rendition sets for umo/umaprofiles:

Default rendition sets are tailored to cover a broad set of profile configurations for the most commonly used settings. Custom rendition sets and profiles are configurable, please contact your Brightcove representative to define your own requirement.

Note: When defining your own set of renditions on delivery, any duplicate renditions will be ignored (e.g. defining **`sdmed,sdhigh`** will discard the duplicate sd1200)

RenditionId	BrevityId	Default Profiles			
825b7f2a-a31b-11e4-bfdb-005056837bc7	audio				
5ff484d6-a33d-11e4-bfdb-005056837bc7	sd264	mobile		sdlow	
ac2e7f0b-a345-11e4-bfdb-005056837bc7	sd512				

74ba4d0b-a347-11e4-bfdb-005056837bc7	sd764		sd		sdmed	
076ea1a2-a35b-11e4-bfdb-005056837bc7	sd1200			tablet		sdhigh
225bd8bb-a577-11e4-bfdb-005056837bc7	sd2000					
9adf3bc3-a578-11e4-bfdb-005056837bc7	hd3000	720p	hd			
468fb310-a585-11e4-bfdb-005056837bc7	hd4400					
8efbb9ca-a586-11e4-bfdb-005056837bc7	hd6500	1080p				

Dynamic ad params

DAP key/value pairs are functional throughout the system; preconfigured and mapped. The Once platform will dynamically pull known client device information from the mapping during runtime and include that data as part of the tracking or targeting parameters expected by the audience measurement systems or Ad Servers, respectively. See Server Side Beaconing (http://docs.brightcove.com/en/once/guides/getting-started-once.html#serverSideBeaconing) and UMADPARAM (http://docs.brightcove.com/en/once/guides/getting-started-once.html#UMADPARAM) for URL examples.

Available Dynamic Ad Parameters (DAP's)	Ad Request	Content Beacon	Ad Beacon
mediaitem.title	✓	✓	✓
• **String value of mediaitem defined by the title ingest object. Spaces will be encoded as %2B.**			
mediaitem.tremor.title	✓	✓	✓✓
• **String value of mediaitem.tremor.title (necessary when using Tremor as an ad provider) Media Item Title with space URL encoded using %20.**			
mediaitem.guid	✓	✓	✓
• **String value of the system generated mediaitemId**			
mediaitem.foreignkey	✓	✓	✓
• **String value of foreignkey defined by the foreignkey ingest object.**			
mediaitem.duration	✓	✓	✓
• **Content file duration in seconds based on the total length of content**			
mediaitem.adposition	✓		
• **Integer value for the ad position, e.g., 1, 2, 3 for pre, mid, post**			
mediaitem.adpositiontime	✓		
• **Time in seconds relative to the content where ad is being played**			

randomnumber32	✓	✓	✓
• **A unique system generated random number**			
timestamputc	✓	✓	✓
• **A 32 bit representation of the servers current time in epoch time (seconds) (UTC)**			
useragent	✓	✓	✓
• **String of the useragent of the requesting client device**			
ipaddress	✓	✓	✓
• **User's IP address**			
dma	✓	✓	✓
• **Designated Market Area - 3 digit numeric code**			
state	✓	✓	✓
• **Abbreviated State code**			
region	✓	✓	✓
• **Name of the region**			
referringHost	✓	✓	✓
• **String of the referring host from which the content was requested**			
referringURL	✓	✓	✓
• **URL encoded representation of the HTTP Header: Referrer**			
countrycode	✓	✓	✓
• **String of the ISO country code retrieved through ip intelligence.**			
postalcode	✓	✓	✓
• **String of the user's postal code (if available) from ip intelligence.**			
xm."key"	✓	✓	✓
• **Used to extract extended metadata defined at ingest by the "metadata" object for the key and its set value.**			
latitude	✓		
• **To accommodate ad providers that require lat/long to be sent with ad tag request. Value based on ip intelligence.**			

longitude	✓		
• **To accommodate ad providers that require lat/long to be sent with ad tag request. Value based on ip intelligence.**			
beacon.quartile.text		✓	✓
• **The quartile text, e.g. Start, FirstQuartile, Midpoint, ThirdQuartile, Complete.**			
beacon.quartile.value		✓	✓
• **The quartile time in seconds relative to the content or ad.**			
beacon.interval.value		✓	✓
• **The time in seconds relative to the content or ad based for heartbeat intervals.**			
mediaitem.segmentcount		✓	
• **Used only for play or end segment beacons. Programmatic way to provide an incremental count for play and/or end beacons.**			
beacon.smartxml.adcreativeid			✓
• **(For Freewheel Smart XML). The creative Id pulled from the Ad Server response. Xpath: adResponse/siteSection/videoPlayer/videoAsset/adSlots/temporalAdSlot/selectedAds/adReference@creativeId**			
beacon.smartxml.adpodid			✓
• **(For Freewheel Smart XML). The customId pulled from the Ad Server response. Xpath: adResponse/siteSection/videoPlayer/videoAsset/adSlots/temporalAdSlot:customId**			
beacon.smartxml.adpodposition			✓
• **(For Freewheel Smart XML). Programmatically generated based on the position sequence of the adCreativeId within a temporal pod.**			
xfp.correlator	✓		
• **(For DFP) Random number correlator specific to using the DFP ad server. Unique per page view. The value will persist to each standard and optimized pod call.**			
xfp.pod	✓		
• **(For DFP) Represents a pod within a video. Pass &pod=1 for first pod, &pod=2 for second pod, and so on.**			
xfp.ppos	✓		
• **(For DFP) Position within a pod. Pass &ppos=1 for the first position, &ppos=2 for the second position, and so on.**			
xfp.vpos	✓		
• **(For DFP) Whether ad request is being sent for preroll, midroll or postroll.**			
xfp.scor	✓		
• **(For DFP) Integer generated for each video stream that reflects unique streams. The value will persist to each standard and optimized pod call.**			
xfp.ipaddress	✓		

• **(For DFP) To carry the IP value into each standard and optimized pod call.**	
xfp.ss_req	✓
• **(For DFP) Required for all initial and subsequent ad tags for the Brightcove-to-XFP session.**	

Note: all xfp.{value} params are for DFP.

Advertising

Description

Once serves up and distributes ads through ad campaigns associated with your ApplicationId to dynamically request the payload from your Ad Server(s). Advertising calls are made server-side on behalf of the client to the ad server configured within your Brightcove account. Our platform supports VAST XML, SmartXML, DFP XFP, and nearly any VAST-like XML format.

Types supported

- **VAST**
- **VAST-Wrapper**
- **DFP**
-
  - **VAST**
  - **VAST-Wrapper**
  - **Ad Pods**
- **Freewheel**
-
  - **VAST**
  - **VAST-Wrapper**
  - **SmartXML**

Value generation

Populating the values necessary to correctly call to an Ad Provider can be be accomplished through three methods: via the query string parameter UMADPARAM, statically mapped and inserted within the Once system, dynamically inserted via DAP (dynamic ad parameters) or any combination of the three.

Static

Static values for ad tag parameters will be preconfigured and set in place in the Brightcove Once database to be fired in each call to your ad provider.

Contextual

The Once platform can utilize Dynamic Ad Parameters to generate some contextual information based off of known client device information and include that data as part of the targeting parameters expected by your ad server. Please see the full list of dynamic ad parameters (http://docs.brightcove.com/en/once/guides/getting-started-once.html#dapTable) for more information.

Page level

For values that are not static or contextually generated, targeting information can be appended to the initial Once URL via the query string values like UMADPARAM. Please see the advertising query string section (http://docs.brightcove.com/en/once/guides/getting-started-once.html#queryStringAd) for more information.

Other features

Ad Cache

If an ApplicationId configured to make ad calls is used in a request for video, Once will send a request to the associated Ad Server to retrieve an ad payload. If the Ad Server returns ad creative that has not been previously returned, the Ad Cache service will automatically ingest that ad creative video within your ad catalog. Video requests that receive ad responses for a particular creative that is currently being ingested will be served with that slot unmonetized. After ingestion of the ad creative has completed, Once will monetize subsequent responses.

Ad Unit Waterfall Method

Brightcove Once Supports ad creative waterfall behavior for VAST XML, Freewheel SmartXML, and DFP XFP. Once makes an ad request to the first node in sequence and, if an ad unit is returned, the ad is rendered with appropriate tracking. If an ad unit is not returned, Once will try the next node in sequence and continue until all nodes have been tried.

With this methodology, an ad creative in the sequence will be used if the ad server response is not empty and includes a valid media file.

Beacons

Beacons are a service within the Once platform that can fire predetermined events to a third-party analytics provider when your content is played. Beacons can be set up for base content or ads and can fire at a number of different times depending on what configuration is requested. You may configure a single play beacon to be fired, quartile events (Start, First Quartile, etc.), or at set intervals of time such as every 5 seconds. You may also have multiple beaconing URLs per application or event.

	Beacon Type	
Event Type	**Single Video**	**Ad**
Play	**Yes**	**No**
• **Content has been requested and first segment returned**		
Start	**Yes**	**Yes**
• **Content or Ad has been requested and first segment returned.**		
First Quartile	**Yes**	**Yes**
• **Content or Ad has reached 25% of its total duration.**		
Midpoint	**Yes**	**Yes**
• **Content or Ad has reached 50% of its total duration.**		
Third Quartile	**Yes**	**Yes**
• **Content or Ad has reached 75% of its total duration**		

Complete	**Yes**	**Yes**
• **Content or Ad has completed**		
Quartiles	**Yes**	**Yes**
• **Creates all the quartiles (first/mid/third/complete)**		
Interval	**Yes**	**No**
• **Heartbeat beacon that can ping on any defined interval.**		
segmentPlay	**Yes**	**No**
• **Indicate a segment has started. A segment is defined by content separated by ad breaks.**		
segmentEnd	**Yes**	**No**
• **Indicate a segment has ended. A segment is defined by content separated by ad breaks.**		

Value generation

Each variable in a beaconing URL can be specified according to preferences and passed dynamically to the beacon call via the query string parameter **`UMBEPARAM`** , statically mapped and inserted within the Once system, dynamically inserted via DAP (dynamic ad parameters) or any combination of these 3 options. Beaconing URLs do not require additional variables. However, the variables must be included if you want to use the **`UMBEPARAM`** query string parameter.

Static

Static values for beacons can be preconfigured and set in place in the Brightcove Once database to always be fired with the associated beacon server base URL.

DAP

The Once platform can automatically generate some contextual information based off of known client device information from the mapping in the runtime database and include that data as part of the tracking or targeting parameters expected by the Beaconing Server or Ad Server. See the full list of dynamic ad parameters (http://docs.brightcove.com/en/once/guides/getting-started-once.html#dapTable).

Page level

For values that are not static or DAP generated, beacon key value pairs can be appended to the initial Once URL request to be utilized as well. See the Beacon query string section (http://docs.brightcove.com/en/once/guides/getting-started-once.html#serverSideBeaconing) for more information.

Your Brightcove representative can assist in the process of configuring beacons. The information required is below:

- **Example URL to be utilized**
- **Explanation of the various query string parameters included on the beacon URL and what their values should be on each request.**
- **Information as to what events should be reported against.**

CDN

Brightcove Once utilizes existing customer CDN relationships to provide the delivery of content to end users. All requests will be brokered by the Once platform, with Once acting as origin, and your CDN delivering the content. This section outlines support and the steps your Brightcove representative will go through with you to configure the Once service along with your CDN provider. Note: If the use of token authentication is desired please see the list of supported hashing algorithms and configuration details here.

Supported providers

- **Akamai**
- **CloudFront**
- **Limelight**
- **Level3**

Details

- **Brightcove will generate and provide a host name for you to use. (Example: customername.dpl.unicornmedia.com)**
- **Your CDN provider will need to set up an *HTTP Large Object* configuration utilizing the provided hostname as the Origin.**
- **Your CDN will need to enable or verify that the following settings are enabled:**
- - **Cache on Query String or Unique-Cache**
  - **If there is a tiered caching option available, this should be enabled as well.**
  - **"Keep alive" should be on**
  - **Cache control should be set to honor Origin headers**

Content Restriction

Publication rules

Publication Rules are a set of **optional** business logic parameters that may be applied to the content you upload into the Brightcove Once platform to restrict playback in a number of ways.

Source

Inherited from domain

By default, a domain is created with a publication rule is set to a 10 year window with no other restrictions. Any catalog created within the domain will inherit the publication rule from the domain. This rule can be modified through the Media Management API to include other default restrictions.

Inherited from catalog

By default, media items created will inherit the publication rule that is assigned to the catalog that the asset is created in. This rule can be edited through the Media Management API to include other default restrictions.

Ingest

At ingest, publication rules can be defined on a per asset basis via the Brightcove Once ingestion methods. Note* If the catalog contains default catalog publication rules, defining rules at ingest will be additive to the media item.

Please see the ingest schema for more information on how to include publication rules into your ingest request.

Start date / end date

Playback is only allowed within the start and end dates.

Geo restrictions

Once supports geo-restriction at the country level both by inclusion and exclusion. Countries are specified by ISO 3166-1 country codes.

- **Allow country or countries: when you allow one or more countries, viewing in all other countries is automatically disallowed.**
- **Deny country or countries: when you disallow viewing in one or more countries, viewing is automatically allowed in all other countries.**

Client restrictions

IpAddress

Playback is allowed or denied based on the IP address of the request.

UserAgent

Playback is allowed or denied based on the value of the useragent from the requesting device.

Referring host

Playback is allowed or denied based on the value of the referring host retrieved from the HTTP header.

Multiple publication rules

Multiple rules can be used if a particular behavior is desired. For example, using two publication rules, you could limit playback to a particular country for a set amount of time before offering the video globally. In practice, creating this implementation would go as follows: Publication rule 1 would allow video playback for a particular specified date range to a particular single country only. Publication rule 2 would start at the end time of rule 1 with a 10 year time limit and no country restriction.

Content Security

AES-128

AES-128 is the Advanced Encryption Standard, which utilizes 128 bit keys for additional protection of HLS content. The Once platform will handle key generation and authorization with the device. When a request for encrypted content is made, authorization information is passed to the requesting device within the HLS manifest as the attribute EXT-X-KEY and must be successfully validated in order for playback to begin.

Configuration of AES within a domain may be done across the entire domain or may be configured to provide multi-mode delivery. Security can be specified on a catalog by and must use an application with the same setting or playback will be denied. Videos ingested into a *secure* catalog will inherit the assigned security settings whereas videos ingested into an *unsecured*catalog will not. Note* When AES-128 is configured, delivery through progressive download will be restricted.

For example, a domain may contain a catalog for secure content such as feature films, episodic, or other premium types as well as a catalog for unsecured content such as clips or trailers. Additionally, that domain would have ApplicationIds for secure content and unsecured content. Secure or unsecured content would only be accessible via the appropriate secure and unsecured ApplicationIds, respectively. Your Brightcove representative can assist in the configuration of AES-128 for your domain.

There are two modes offered for AES-128.

- **Mode A - Unencrypted ads with encrypted content**
- **Mode B - The entire stream will be encoded (content + ads)**

Note: Mode A is recommended when delivering both secure and unsecure content to maximize caching efficiency of ad creatives.

Apple FairPlay DRM

FairPlay is an Apple DRM technology that is exclusively used to protect streamed and downloaded content on Apple devices. Apple licenses this technology to enable content owners to deliver content to apps running on Apple TV.

Current limitations:

- **One license server configuration per Once domain**
- **FairPlay packaged content will not be playable outside of iOS or AppleTV approved applications**
- **The Brightcove License server configuration and storing your content key must be done by Brightcove.**

Due to complexities with Fairplay configurations please contact your Brightcove representative to appropriately configure your domain and setup of your application to communicate with the license server. Brightcove requires the following information:

1. **Application Certificate (AC) (Provided by Apple in the Phase 2 SDK)**
2. **Application Secret Key (ASK) (Provided by Apple in the Phase 2 SDK)**
3. **Private key used to sign the CSR. (Generated by publisher)**
4. **Password for the private key (if private key is encrypted) (See 3)**

The CSR is not security sensitive, but the private key is. The CSR Private Key is generated by the publisher and would have been passed to Apple during the CSR signing process. Due to the secure nature of the private key, we ask that the Private Key is sent in a secure manner, preferably wrapped and encrypted in some manner.

CDN Token Authentication

As an added layer of security, CDN token based authentication and hashing algorithms are supported for content delivery through Brightcove Once. The authorization tokens to the CDN will be generated by the Once service and requires the hashing and appendage of specific parameters on the Once URL to pass the Once authorization.

Currently Once supports Token Authentication for the following CDNs:

- **Akamai EdgeAuth 1.0/2.0**
- **Level 3**
- **Amazon Cloudfront**

Contact your Brightcove representative for assistance on configurations.

Shared Secrets

The Once token authentication system utilizes two shared secrets that are required; the CDN shared secret and a customer defined shared secret. The customer shared secret is used by Once as a gateway to authentication through the Once API and should not be confused by the CDN shared secret for resource access. By default, the same shared secret is used for both layers of authentication, but if desired can be different on an application basis.

Appending token authentication parameters

This section assumes knowledge of the Once URL Creation, please see creating a Once URL Creation (http://docs.brightcove.com/en/once/guides/getting-started-once.html#onceURLcreation) for the URL structure. The following parameters are necessary for utilizing token authentication.

Parameter	Required	Description
umsstime	Yes	Start time of authorized content. (UNIX epoch time (seconds))
`umsetime`	Yes, if `umsttl` is not used	End time of authorized content. (UNIX epoch time (seconds))
`umsttl`	Yes, if `umsttl` is not used	Number in seconds after the start time to keep the content authorized.
`umshash`	Yes	HMAC-SHA1 hash of URL path and query - using the common shared secret

Akamai EdgeAuth

This section covers the required information for Brightcove Once to handle token authentication through Akamai, but assumes all the configurations between the customer and CDN have been complete.

Customer Requirements:

- **Akamai shared secret**
- **Customer common shared secret**
- **CDN hostname**
- **Hashing requirements for full or partial**

- **Full (host/path/resource?key=value)**
- **Partial (path/resource?key=value)**

- **Failover URL for unauthenticated requests (optional)**

Level 3

This section covers the required information for Brightcove Once to handle token authentication through Level 3, but assumes all the configurations between the customer and CDN have been complete.

Customer Requirements:

- **Level 3 shared secret(s) (Up to 10 active keys)**
- **Customer common shared secret**
- **CDN hostname**
- **Hashing requirements for full or partial**
- **Full (host/path/resource?key=value)**
- **Partial (path/resource?key=value)**
- **Failover URL for unauthenticated requests (optional)**

Shared Secret Configuration:

It is important that the key and entry associations be the same between the CDN and the Once system. If multiple tokens are available in the same window, the Once system will choose the first available to utilize for the hashing the authentication request.

Entry	Secret (up to 64 chars)	Not Valid Before	Not Valid After
0	sharedsecret0	20071113050000	20081225080000
1	sharedsecret1	20081225080000	20090211120000

Note: The chosen parameters to be passed to the CDN must be set between the customer and the CDN. By default the following are passed:

- **Hash token : token**
- **Not valid before token : nvb**
- **Not valid after token : nva**

Error Events

Error Code	Explanation	Suggested Steps
`DownloadAccessDeniedError`	Once attempted to download the submitted media file at the specified address, but received a permission error. If the source URL is hosted on S3, read access has likely not been granted for the file or the bucket in which it's contained. The simplest way to grant read permissions is to utilize an S3 client, like Amazon's AWS Console. In technical speak, Once received a 403 Access Denied response from the hosting server.	Check your source content and retry ingest.
`DownloadFailureError`	This error occurs when Once is unable to download a file for a reason other than Access Denied or File Not Found. If the server returns a response code, Once will display it in the error message so that you can determine what happened and make any necessary updates to the file.	Check your source content and retry ingest.
`DownloadTimeoutError`	Once attempted to download the submitted media file, but did not receive a response from the hosting server in a reasonable amount of time..	Check your source content and host and retry ingest.
`FileNotFoundError`	When attempting to download the submitted file at the specified address, Once was unsuccessful. Most commonly this could indicate that there was a typo in the source url, the file has been removed, a redirect loop was encountered, or another cause preventing successful download. In the case of HTTP this means that Once received an HTTP response code >= 300 when attempting to download the file. (403 errors will be classified separately as DownloadAccessDenied errors.)	Check your source content and host and retry ingest.
`FilePermanentlyMovedError`	Once attempted to download a file at the specified address, but the source server returned a 301 code.	Check your source content and retry ingest.
`InaccurateDurationError`	The output generated for this file has a duration that is significantly longer than the input file's duration. Generally this is the result of invalid/inaccurate metadata or timestamps that cause the encoder to create a long file.	Check your source content and retry ingest.
`InvalidDownloadedFileTypeError`		Check your source content

	Once tried to download a file at the specified address, but instead of a video or audio file, got back something else, like a HTML or XML document.	and retry ingest.
`NoMediaError`	The input file is not an audio or video file. It could be a PDF, an image, an .exe, or something else, but it does not appear to be a video or a sound file.	Check your source content.
`QuicktimeReferenceError`	The input file references audio or video tracks that aren't actually present in the file. The file is effectively acting as a placeholder that links to outside media tracks.	Check your source content.
`S3BucketNotFoundError`	No bucket was found on S3 with the specified name. If you are using the s3:// protocol in your URLs, make sure that you only include the bucket name - "s3://my_bucket" - and not the "s3.amazonaws.com" domain.	Check your source content url and retry ingest.
`TimedTextValidationError`	The timed text file did not pass validation, or is not a supported format. Check the message of the error and compare it against your input.	Check your source content and retry ingest.
`TranscodeError`	Indicates that transcoding the submitted file was not successful.	Contact Brightcove Support.
`TruncatedFileError`	The input file is smaller than the headers say it should be. Generally this means the file got cut off at some point previous to being transferred.	Check your source content.
`UnexpectedResultError`	An unexpected error occurred. Contact Brightcove Support for more information.	Contact Brightcove Support.
`UnreadableFileError`	Indicates that a media file was unable to be decoded, typically due to the source being corrupt.	Check your source content.
`UnsupportedCodecError`	**The source content is using a codec that Once currently does not support. See Supported Video Formats (./supported-formats.html) for a list of formats that are not supported.**	**Contact Brightcove Support.**
`UnsupportedEncodingError`	**This error occurs when the submitted file was encoded in a way that Once does not currently support.**	**Contact Brightcove Support.**
`UnsupportedEncryptionError`	**This error occurs because the submitted file is encrypted, preventing us from opening it.**	**Contact Brightcove Support.**

Ingest Schema

```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "type": "object",
    "title": "root",
    "description": "Ingest API Container",
    "additionalProperties": false,
    "properties": {
        "title": {
            "type": "string",
            "description": "The title of the asset",
            "minLength": 0,
            "maxLength": 255,
            "additionalProperties": false
        },
        "foreignKey": {
            "type": "string",
            "description": "The unique identifier for the asset",
            "minLength": 1,
            "maxLength": 255,
            "additionalProperties": false
        },
        "metadata": {
            "type": "object",
            "description": "An array of key value pairs for Extended Metadata",
            "additionalProperties": true,
            "properties": {
                "key": {
                    "type": "string",
                    "description": "The key of an Extended Metadata key value pair",
                    "minLength": 0,
                    "additionalProperties": false
                }
            }
        },
        "media": {
            "type": "object",
            "description": "Container for the source URL of the asset being ingested",
            "additionalProperties": false,
            "properties": {
                "sourceURL": {
                    "type": "string",
                    "description": "The URL string to the source asset",
                    "minLength": 1,
                    "maxLength": 1000,
                    "additionalProperties": false
                }
            },
            "required": [
                "sourceURL"
            ]
        },
        "timedText": {
            "type": "array",
            "description": "Anarrayoftimedtextassociationsforthemediaitem",
            "minItems": 1,
            "uniqueItems": false,
            "additionalProperties": false,
            "items": {
                "media": {
                    "type": "object",
                    "description": "ContainerforthesourceURLoftheassetbeingingested",
                    "sourceURL": {
                        "type": "string",
                        "description": "TheURLstringtothesourceasset",
                        "minLength": 1,
                        "maxLength": 1000,
                        "additionalProperties": false
                    },
                    "passThrough": {
                        "type": "boolean",
"description": "Onlyrequiredwheninformingtheencoderofembeddedcaptionstobeextracted",
"additionalProperties": false"timedText": {
    "type": "array",
    "description": "An array of Timed Text items for the asset",
    "minItems": 1,
    "uniqueItems": true,
    "additionalProperties": false,
    "items": [
        {
            "type": "object",
            "description": "Container describing a timed text item and its contents",
            "additionalProperties": false,
            "properties": {
                "media": {
                    "type": "object",
                    "description": "Container for the source URL of the timed text file
                    "additionalProperties": false,
                    "properties": {
                        "sourceURL": {
                            "type": "string",
                            "description": "The URL string to the source asset",
                            "minLength": 1,
                            "maxLength": 1000,
                            "additionalProperties": false
                        }
                    },
                    "required": [
                        "sourceURL"
                    ]
                },
                "timedTextType": {
                    "type": "string",
                    "description": "The type to categorize the timed text item",
                    "additionalProperties": false,
                    "enum": [
                        "Subtitle",
                        "Caption",
                        "Embedded"
                    ]
                },
                "languages": {
                    "type": "array",
                    "description": "An array of languages contained in the timed text as
                    "additionalProperties": false,
                    "items": [
                        {
                            "type": "string",
                            "description": "A ISO-639 language code to be found in the a
                            "minLength": 2,
                            "maxLength": 255,
                            "additionalProperties": false
                        }
                    ],
                    "required": [
                        "1"
                    ]
                },
                "alternateId": {
                    "type": "string",
                    "description": "The optional id to associate with the timed text ite
                    "minLength": 1,
                    "maxLength": 255,
                    "additionalProperties": false
                }
            },
            "required": [
                "media",
                "timedTextType",
                "languages"
```

```
                                    ]
                                }
                            ],
                            "required": [
                                "0"
                            ]
                        }
                    },
                    "required": [
                        "foreignKey",
                        "media"
                    ]
                },
                "priority": {
                    "type": "string",
                    "description": "Denotes the transcode priority of the asset",
                    "minLength": 0,
                    "additionalProperties": false,
                    "enum": [
                        "Normal"
                    ]
                },
                "publicationRules": {
                    "type": "array",
                    "description": "An array of Publication Rules for the asset",
                    "minItems": 1,
                    "uniqueItems": false,
                    "additionalProperties": false,
                    "items": {
                        "type": "object",
                        "description": "Container for Client Filters and Country Rules for the Publication Rules
                        "additionalProperties": false,
                        "properties": {
                            "channel": {
                                "type": "string",
                                "description": "The Channel Id for the Publication Rule",
                                "minLength": 36,
                                "maxLength": 36,
                                "additionalProperties": false
                            },
                            "startDate": {
                                "type": "integer",
                                "description": "The start date for the Publication Rule",
                                "minimum": 0,
                                "maximum": 2147483647,
                                "additionalProperties": false
                            },
                            "endDate": {
                                "type": "integer",
                                "description": "The end date for the Publication Rule",
                                "minimum": 0,
                                "maximum": 2147483647,
                                "additionalProperties": false
                            },
                            "clientFilters": {
                                "type": "array",
                                "description": "And array of Client Filters for the Publication Rule",
                                "minItems": 1,
                                "uniqueItems": false,
                                "additionalProperties": false,
                                "items": {
                                    "type": "object",
                                    "description": "Container for a Client Filter",
                                    "additionalProperties": false,
                                    "properties": {
                                        "variableName": {
                                            "type": "string",
                                            "description": "The variable name that the Client Filter will key of
                                            "additionalProperties": false,
                                            "enum": [
                                                "IpAddress",
                                                "UserAgent",
                                                "ReferringHost"
                                            ]
                                        },
                                        "value": {
                                            "type": "string",
                                            "description": "The value that the Client Filter will key off of",
                                            "minLength": 0,
                                            "maxLength": 255,
                                            "additionalProperties": false
                                        },
                                        "filterType": {
                                            "type": "string",
                                            "description": "The type of filtering used to compare the value",
                                            "additionalProperties": false,
                                            "enum": [
                                                "Equals",
                                                "NotEquals",
                                                "In",
                                                "NotIn",
                                                "Contains",
                                                "NotContains",
                                                "StartsWith",
                                                "NotStartsWith",
                                                "EndsWith",
                                                "NotEndsWith"
                                            ]
                                        },
                                        "isDenied": {
                                            "type": "boolean",
                                            "description": "Denotes whether a successful comparison of the Clien
                                            "additionalProperties": false
                                        }
                                    },
                                    "required": [
                                        "variableName",
                                        "value",
                                        "filterType",
                                        "isDenied"
                                    ]
                                },
                                "required": [
                                    "0"
                                ]
                            },
                            "countryRules": {
                                "type": "array",
                                "description": "An array of Country Rules for the asset",
                                "minItems": 1,
                                "uniqueItems": false,
                                "additionalProperties": false,
                                "items": {
                                    "type": "object",
                                    "description": "Container for the Country Rule",
                                    "additionalProperties": false,
                                    "properties": {
                                        "countryCode": {
                                            "type": "string",
                                            "description": "The Country Code defined by ISO 3166-1",
                                            "minLength": 2,
                                            "maxLength": 2,
                                            "additionalProperties": false,
                                            "enum": [
                                                "AF",
                                                "AL",
                                                "DZ",
                                                "AS",
        "AD",
        "AO",
        "AI",
        "AQ",
        "AG",
        "AR",
        "AM",
        "AW",
        "AU",
        "AT",
        "AZ",
        "BS",
        "BH",
        "BD",
        "BB",
        "BY",
        "BE",
        "BZ",
        "BJ",
        "BM",
        "BT",
        "BO",
        "BA",
        "BW",
        "BR",
        "IO",
        "VG",
        "BN",
        "BG",
        "BF",
        "MM",
        "BI",
        "KH",
        "CM",
        "CA",
        "CV",
        "CF",
        "TD",
        "CL",
        "CN",
        "CX",
        "CC",
        "CO",
        "KM",
        "CK",
        "CR",
        "HR",
        "CU",
        "CY",
        "CZ",
        "CD",
        "DK",
        "DJ",
        "DM",
        "DO",
        "EC",
        "EG",
        "SV",
        "GQ",
        "ER",
        "EE",
        "ET",
        "FK",
        "FO",
        "FJ",
        "FI",
        "FR",
        "PF",
        "GA",
"GM",
"GE",
"DE",
"GH",
"GI",
"GR",
"GL",
"GD",
"GU",
"GT",
"GN",
"GW",
"GY",
"HT",
"VA",
"HN",
"HK",
"HU",
"IS",
"IN",
"ID",
"IR",
"IQ",
"IE",
"IM",
"IL",
"IT",
"CI",
"JM",
"JP",
"JE",
"JO",
"KZ",
"KE",
"KI",
"KW",
"KG",
"LA",
"LV",
"LB",
"LS",
"LR",
"LY",
"LI",
"LT",
"LU",
"MO",
"MK",
"MG",
"MW",
"MY",
"MV",
"ML",
"MT",
"MH",
"MR",
"MU",
"YT",
"MX",
"FM",
"MD",
"MC",
"MN",
"ME",
"MS",
"MA",
"MZ",
"NA",
"NR",
"NP",
"NL",
"AN",
"NC",
"NZ",
"NI",
"NE",
"NG",
"NU",
"KP",
"MP",
"NO",
"OM",
"PK",
"PW",
"PA",
"PG",
"PY",
"PE",
"PH",
"PN",
"PL",
"PT",
"PR",
"QA",
"CG",
"RO",
"RU",
"RW",
"BL",
"SH",
"KN",
"LC",
"MF",
"PM",
"VC",
"WS",
"SM",
"ST",
"SA",
"SN",
"RS",
"SC",
"SL",
"SG",
"SK",
"SI",
"SB",
"SO",
"ZA",
"KR",
"ES",
"LK",
"SD",
"SR",
"SJ",
"SZ",
"SE",
"CH",
"SY",
"TW",
"TJ",
"TZ",
"TH",
"TL",
"TG",
"TK",
"TO",
"TT",
                                                "TN",
                                                "TR",
                                                "TM",
                                                "TC",
                                                "TV",
                                                "UG",
                                                "UA",
                                                "AE",
                                                "GB",
                                                "US",
                                                "UY",
                                                "VI",
                                                "UZ",
                                                "VU",
                                                "VE",
                                                "VN",
                                                "WF",
                                                "EH",
                                                "YE",
                                                "ZM",
                                                "ZW"
                                            ]
                                        },
                                        "isDenied": {
                                            "type": "boolean",
                                            "description": "Denotes whether a successful comparison of the Clien
                                            "additionalProperties": false
                                        }
                                    },
                                    "required": [
                                        "countryCode",
                                        "isDenied"
                                    ]
                                },
                                "required": [
                                    "0"
                                ]
                            }
                        },
                        "required": [
                            "channel",
                            "startDate",
                            "endDate"
                        ]
                    },
                    "required": [
                        "0"
                    ]
                },
                "cuePoints": {
                    "type": "array",
                    "description": "An array of Cue Points for the asset",
                    "minItems": 1,
                    "uniqueItems": true,
                    "additionalProperties": false,
                    "items": [
                        {
                            "type": "object",
                            "description": "A key value pair of the time in which the Cue Point and unit type of
                            "additionalProperties": false,
                            "properties": {
                                "valueIn": {
                                    "type": "integer",
                                    "description": "The time in which the Cue Point will be inserted",
                                    "minimum": 0,
                                    "maximum": 2147483647,
                                    "additionalProperties": false
                                },
                                "unit": {
```

```
                            "type": "string",
                            "description": "The type of unit the time value",
                            "minLength": 0,
                            "additionalProperties": false,
                            "enum": [
                                "Seconds"
                            ]
                        }
                    },
                    "required": [
                        "valueIn",
                        "unit"
                    ]
                }
            ],
            "required": [
                "0"
            ]
        }
    },
    "required": [
        "foreignKey",
        "media"
    ]
}
```