



v2015-08-17 Property Manager Deprecated Rule Formats

December 21, 2022

Contents

Welcome

[Welcome](#)

PAPI conventions

[API versioning](#)

[Advanced and locked features](#)

v2015-08-17 behaviors

[v2015-08-17 behaviors](#)

[adaptiveImageCompression](#)

[advanced](#)

[akamaizer](#)

[akamaizerTag](#)

[allHttpInCacheHierarchy](#)

[allowCloudletsOrigins](#)

[allowDelete](#)

[allowPatch](#)

[allowPost](#)

[allowPut](#)

[asset_prioritization](#)

[audience_segmentation](#)

[baseDirectory](#)

[breakConnection](#)

[cacheError](#)

[cacheId](#)

[cacheKeyIgnoreCase](#)

[cacheKeyQueryParams](#)

[cacheKeyRewrite](#)

[cachePost](#)

[cacheRedirect](#)

[caching](#)

[centralAuthorization](#)

[chaseRedirects](#)

constructResponse
continuousDeployment
cpCode
deliveryReceipt
denyAccess
deviceCharacteristicCacheId
deviceCharacteristicHeader
dnsAsyncRefresh
dnsPrefetch
downgradeProtocol
downstreamCache
edgeConnect
edgeImageConversion
edgeLoadBalancingAdvanced
edgeLoadBalancingDataCenter
edgeLoadBalancingOrigin
edgeOriginAuthorization
edgeScape
edgeSideIncludes
edge_redirector
enhancedAkamaiProtocol
failAction
forwardRewrite
g2oheader
gzipResponse
hdDataAdvanced
healthDetection
http2
imageManager
inputValidation
instant
instantConfig
largeFileOptimization
limitBitRate
mediaFileRetrievalOptimization
modifyIncomingRequestHeader
modifyIncomingResponseHeader
modifyOutgoingRequestHeader
modifyOutgoingResponseHeader
netSession

networkConditionsHeader
origin
persistentClientConnection
persistentConnection
personallyIdentifiableInformation
predictivePrefetching
prefetch
prefetchable
prefreshCache
randomSeek
readTimeout
realUserMonitoring
redirect
referrerChecking
removeQueryParameter
removeVary
report
requestControl
responseCode
responseCookie
restrictObjectCaching
rewriteUrl
rmaOptimization
saasdefinitions
savePostDcaProcessing
scheduleInvalidation
segmentedContentProtection
segmentedMediaOptimization
shutr
simulateErrorCode
siteShield
spdy
subCustomer
sureRoute
tcpOptimization
tieredDistribution
timeout
validateEntityTag
verifyTokenAuthorization
visitor_prioritization

watermarkUrl
webApplicationFirewall
webdav

v2015-08-17 criteria

v2015-08-17 criteria

bucket
cacheability
clientIp
clientIpVersion
cloudletsOrigin
contentDeliveryNetwork
contentType
deviceCharacteristic
fileExtension
filename
hostname
matchAdvanced
matchCpCode
matchResponseCode
originTimeout
path
queryStringParameter
random
requestCookie
requestHeader
requestMethod
requestProtocol
responseHeader
time
tokenAuthorization
userAgent
userLocation
userNetwork

Notice

Notice

Welcome

Welcome

Akamai often modifies Property Manager API (PAPI) features, each time deploying a new internal version of the feature. By default, the Property Manager interface in [Control Center](#)⁺ uses the latest available feature versions and you may be prompted to upgrade your configuration. In the interest of stability, PAPI does not support this system of selective updates for each feature. Instead, PAPI's rule objects are simply versioned as a whole. These versions, which update infrequently, are known as rule formats.

PAPI supports different dated versions for the set of features available within a property's rule tree. Akamai releases a new stable version of a rule format twice a year on average. As best practice, you should upgrade to the most recent dated rule format available. See [API versioning](#) for details.

This guide provides details for all behaviors and criteria the Property Manager API supports in the v2017-06-19 **deprecated** rule format version. The version available to you is determined by the product and modules assigned to the property. You can get it by running the [List available behaviors for a property](#) operation.

PAPI conventions

API versioning

The API exposes several different versioning systems:

- The version of the API is specified as part of the URL path. The current API version is `v1`.
- The API supports different dated versions for the set of features available within a property's rule tree. You can [freeze](#) and smoothly [update](#) the set of features that a property's rules apply to your content. Each behavior and criteria you invoke within your rules may independently increment versions from time to time, but you can only specify the most recent dated rule format to freeze the set of features. Otherwise, if you assign the `latest` rule format, features update automatically to their most recent version. This may abruptly result in errors if JSON in your rules no longer comply with the most recent feature's set of requirements.



Once you've frozen a rule format in PAPI, that state persists even if you use the Property Manager interface in [Control Center](#). You no longer get any feature upgrade prompts.

- The latest set of features are detailed in the [behavior](#) and [criteria](#) reference.
- PAPI lets you access your own set of property versions. Versions are available as URL resources that you can modify and activate independently, or perform roll-back if needed. This set is the only versioned object under your direct control.
- The API's [Build interface](#) also provides details on the current software release and its accompanying *catalog* of behaviors and criteria. These include version numbers and extraneous commit and build dates, which bear no relation to dated rule format versions. Don't rely on any of the internal version numbers this interface makes available.

Expect internal catalog release versions to update the most frequently, followed by less frequent rule format versions, followed by infrequent new API versions.

Advanced and locked features

In addition to its `name` and component `options`, special types of behavior and criteria objects may feature these additional members:

- A `uuid` string signifies an *advanced* feature. Advanced behaviors and criteria are read-only, and can only be modified by Akamai representatives. They typically deploy metadata customized for you, whose functionality falls outside the predefined guidelines of what other read/write behaviors can do. Such metadata might also cause problems if executed outside of its intended context within the rule tree. Throughout the behavior and criteria reference, advanced features are identified as *read-only*.
- If a `locked` boolean member is `true`, it indicates a behavior or criteria that your Akamai representative has *locked* so that you can't modify it. You typically arrange with your representative to lock certain behaviors to protect sensitive data from erroneous changes. Any kind of behavior or criteria may be locked, including writable ones.

When modifying rule trees, you need to preserve the state of any `uuid` or `locked` members. You receive an error if you try to modify or delete either of these special types of feature. You can reposition regular features relative to these special ones, for example by inserting them within the same rule, but each rule's sequence of special features needs to remain unchanged.

Higher-level rule trees may also indicate the presence of these special features:

- A `uuid` member present on a rule object indicates that at least one of its component behaviors or criteria is advanced and read-only. You need to preserve this `uuid` as well when modifying the rule tree.
- A `criteriaLocked` member enabled on a criteria rule by your Akamai representative means that you may *not* insert additional criteria objects within the sequence. This typically keeps complex logical tests from breaking. Preserve the state of `criteriaLocked` when modifying the rule tree.

v2015-08-17 behaviors

v2015-08-17 behaviors

The following represents all rule behaviors the Property Manager API supports. The set available to you is determined by the product and modules associated with the property. Use the [List Available Behaviors](#) operation to get this information.

This reference specifies behaviors used in the `v2015-08-17 rule format`. See the [most recent feature set](#), which corresponds to the `latest` rule format.

adaptiveImageCompression

The Adaptive Image Compression feature compresses JPEG images depending on the requesting network's performance, thus improving response time. The behavior specifies three performance tiers based on round-trip tests: 1 for excellent, 2 for good, and 3 for poor. It assigns separate performance criteria for mobile (cellular) and non-mobile networks, which the `do_aic_mobile` and `do_aic_nonmobile` options enable independently.

There are six `method` options, one for each tier and type of network. If the `method` is `compress`, choose from among the six corresponding `slider` options to specify a percentage. As an alternative to compression, setting the `method` to `strip` removes unnecessary application-generated metadata from the image. Setting the `method` to `bypass` serves clients the original image.

The behavior serves `ETags` headers as a data signature for each adapted variation. In case of error or if the file size increases, the behavior serves the original image file. Flushing the original image from the edge cache also flushes adapted variants. The behavior applies to the following image file extensions: `jpg`, `jpeg`, `jpe`, `jif`, `jfif`, and `jfi`.

Options

- `do_aic_mobile` (*boolean*): When enabled, adapts images served over cellular mobile networks. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `do_aic_nonmobile` (*boolean*): When enabled, adapts images served over non-cellular networks. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `standard_comp_t1_method` (*enum string*): Specifies tier-1 non-cellular network behavior, either `compress` , `strip` , or `bypass` .
- `standard_comp_t1_slider` (*number within 0-100 range*): With `standard_comp_t1_method` set to `compress` , this specifies the compression percentage.
- `standard_comp_t2_method` (*enum string*): Specifies tier-2 non-cellular network behavior, either `compress` , `strip` , or `bypass` .
- `standard_comp_t2_slider` (*number within 0-100 range*): With `standard_comp_t2_method` set to `compress` , this specifies the compression percentage.
- `standard_comp_t5_method` (*enum string*): Specifies tier-5 non-cellular network behavior, either `compress` , `strip` , or `bypass` .
- `standard_comp_t5_slider` (*number within 0-100 range*): With `standard_comp_t5_method` set to `compress` , this specifies the compression percentage.
- `mobile_comp_t1_method` (*enum string*): Specifies tier-1 behavior, either `compress` , `strip` , or `bypass` .
- `mobile_comp_t1_slider` (*number within 0-100 range*): With `mobile_comp_t1_method` set to `compress` , this specifies the compression percentage.
- `mobile_comp_t2_method` (*enum string*): Specifies tier-2 cellular-network behavior, either `compress` , `strip` , or `bypass` .
- `mobile_comp_t2_slider` (*number within 0-100 range*): With `mobile_comp_t2_method` set to `compress` , this specifies the compression percentage.
- `mobile_comp_t5_method` (*enum string*): Specifies tier-5 cellular-network behavior, either `compress` , `strip` , or `bypass` .
- `mobile_comp_t5_slider` (*number within 0-100 range*): With `mobile_comp_t5_method` set to `compress` , this specifies the compression percentage.

Feature previously named: `aic`

Related behaviors: [redirect](#) , [enhancedAkamaiProtocol](#) , [cacheKeyIgnoreCase](#) , [cacheKeyQueryParams](#) , [cacheError](#) , [removeVary](#)

advanced

A [read-only behavior](#) that specifies Akamai XML metadata. It can only be configured on your behalf by Akamai Professional Services.

Options

- `description` (*string*): Human-readable description of what the XML block does.
- `xml` (*string*): Akamai XML metadata.

Related behaviors: [cacheError](#) , [redirect](#) , [webApplicationFirewall](#) , [cacheKeyQueryParams](#) , [removeVary](#) , [cacheKeyIgnoreCase](#)

akamaizer

This [read-only behavior](#) allows you to run regular expression substitutions over web pages. Contact Akamai Professional Services for help configuring the Akamaizer. See also the [akamaizerTag](#) behavior.

Options

- `status` (*boolean*): Enables the Akamaizer behavior. (In the Beta API, please substitute "on" and "off" string values.)

Related behaviors: [modifyOutgoingResponseHeader](#) , [modifyOutgoingRequestHeader](#) , [removeVary](#) , [redirect](#) , [cachePost](#) , [dnsAsyncRefresh](#)

akamaizerTag

This [read-only behavior](#) specifies HTML tags and replacement rules for hostnames used in conjunction with the [akamaizer](#) behavior. Contact Akamai Professional Services for help configuring the Akamaizer.

Options

- `matchhostname` (*string*): Specifies the hostname to match on as a Perl-compatible regular expression.
- `replacementhostname` (*string*): Specifies the replacement hostname for the tag to use.
- `scope` (*enum string*): Specifies the part of HTML content the `tags_attr` refers to:
 - `attr` for when `tags_attr` refers to a tag/attribute pair, the match only applies to the attribute.
 - `urlattr` is the same as `attr`, but applies when the attribute value is a URL. In that case, it converts to an absolute URL prior to substitution.
 - `block` substitutes within the tag's contents, but not within any nested tags.
 - `page` ignores the `tags_attr` field and performs the substitution on the entire page.
- `tags_attr` (*enum string*): Specifies the tag or tag/attribute combination to operate on, any of the following:

<code>a</code>	<code>form_action</code>	<code>script</code>
<code>a_href</code>	<code>iframe</code>	<code>script_src</code>
<code>area</code>	<code>iframe_src</code>	<code>table</code>
<code>area_href</code>	<code>img</code>	<code>table_background</code>
<code>base</code>	<code>img_src</code>	<code>td</code>
<code>base_href</code>	<code>link</code>	<code>td_background</code>
<code>form</code>	<code>link_href</code>	

- `behavior` (*enum string*): Either `stop` to replace only one match, or `continue` to replace all.
- `type` (*boolean*): Whether to `include` or `exclude` what `tags_attr` specifies. (In the Beta API, please substitute "include" and "exclude" string values.)

Feature previously named: `akamaizertag`

Related behaviors: [modifyOutgoingResponseHeader](#), [modifyOutgoingRequestHeader](#), [redirect](#), [cachePost](#), [removeVary](#), [webApplicationFirewall](#)

allHttpInCacheHierarchy

Allow all HTTP request methods to be used for the edge's parent servers, useful to implement features such as [SiteShield](#), [SureRoute](#), and Tiered Distribution. (See the [siteShield](#) , [sureRoute](#) , and [tieredDistribution](#) behaviors.)

Options

- `allow` (*boolean*): Enables all HTTP requests for parent servers in the cache hierarchy.

(In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `enableallmethodscacheh`

Related behaviors: [allowPut](#) , [allowDelete](#) , [webApplicationFirewall](#) , [redirect](#) , [cacheKeyQueryParams](#) , [removeVary](#)

allowCloudletsOrigins

Allows Cloudlets Origins to determine the criteria, separately from the Property Manager, under which alternate [origin](#) definitions are assigned.

This behavior must appear alone within its own rule. When enabled, it allows any [cloudletsOrigin](#) criteria within sub-rules to override the prevailing origin.

Options

- `enabled` (*boolean*): Allows you to assign custom origin definitions referenced in sub-rules by [cloudletsOrigin](#) labels. If disabled, all sub-rules are ignored.
- `honorBaseDirectory` (*boolean*): If enabled, prefixes any Cloudlet-generated origin path with a path defined by an Origin Base Path behavior. If no path is defined, it has no effect. If another Cloudlet policy already prepends the same Origin Base Path, the path is not duplicated.

Feature previously named: `conditionalOriginBehavior`

allowDelete

Allow HTTP requests using the DELETE method. By default, only GET and HEAD requests are allowed, and all other methods result in a 403 error. Such content does not cache, and any DELETE requests pass to the origin. See also [allowPost](#) , [allowPut](#) , and [allowPatch](#) .

Options

- `allow` (*boolean*): When enabled, allows DELETE requests. Content does *not* cache. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `allowdelete`

Related behaviors: [allowPut](#) , [cacheKeyIgnoreCase](#) , [cacheError](#) , [redirect](#) , [allHttpInCacheHierarchy](#) , [cacheKeyQueryParams](#)

allowPatch

Allow HTTP requests using the PATCH method. By default, only GET and HEAD requests are allowed, and all other methods result in a 403 error. Such content does not cache, and any PATCH requests pass to the origin. See also [allowPut](#) , [allowPost](#) , and [allowDelete](#) .

Options

- `allow` (*boolean*): When enabled, allows PATCH requests. Content does *not* cache. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `allowpatch`

allowPost

Allow HTTP requests using the POST method. By default, only GET and HEAD are allowed, and all other methods result in a 403 error. See also [allowPut](#) , [allowDelete](#) , and [allowPatch](#) .

Options

- `allow` (*boolean*): When enabled, allows POST requests. (In the Beta API, please substitute "on" and "off" string values.)
- `allow_without_content_length` (*boolean*): By default, POST requests also require a `Content-Length` header, or they result in a 411 error. With this option enabled with no specified `Content-Length` , the edge server relies on a `Transfer-Encoding` header to chunk the data. If neither header is present, it assumes the request has no body, and it adds a header with a 0 value to the forward request. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `allowpost`

Related behaviors: [cacheKeyIgnoreCase](#) , [downstreamCache](#) , [cacheError](#) , [removeVary](#) , [cacheRedirect](#) , [modifyOutgoingResponseHeader](#)

allowPut

Allow HTTP requests using the PUT method. By default, only GET and HEAD are allowed, and all other methods result in a 403 error. Such content does not cache, and any PUT requests pass to the origin. See also [allowPost](#) , [allowDelete](#) , and [allowPatch](#) .

Options

- `allow` (*boolean*): When enabled, allows PUT requests. Content does *not* cache. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `allowput`

Related behaviors: [allowDelete](#) , [redirect](#) , [cacheKeyIgnoreCase](#) , [cacheError](#) , [allHttpInCacheHierarchy](#) , [removeVary](#)

asset_prioritization

Enables the API Prioritization Cloudlet, which maintains continuity in user experience by serving an alternate static response when load is too high. You can configure rules using either the Cloudlets Policy Manager application or the [Cloudlets API](#). The feature is designed to serve static API content, such as fallback JSON data.

NOTE: If you want to serve non-API HTML content, see the [visitor_prioritization](#) behavior.

Options

- `status` (*boolean*): Activates the API Prioritization feature. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `nimbus_policy_token` (*string*): Specifies the name of the API Prioritization policy, using alphanumeric and underscore characters.
- `use_throttled_cpcode` (*boolean*): Specifies whether to apply an alternative CP code for requests served the alternate response. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `throttled_cpcode` (*object*): With `use_throttled_cpcode` enabled, this specifies the CP code as an object:

```
"cpcode": {  
  "id" : 12345,  
  "name" : "sent to waiting room"  
}
```

- `netstorage` (*object*): Specify the NetStorage domain that contains the alternate response. For example:

```
"netstorage": {  
  "id" : "id_string",  
  "name" : "Waiting Room",  
}
```

```
"downloadDomainName" : "example.wait.akamai.com",  
"cpCode" : 12345  
}
```

- `sr_file` (*string*): Specify the full NetStorage path for the alternate response, including trailing file name.
- `label` (*string*): A label to distinguish this API Prioritization policy from any others in the same property.

audience_segmentation

Allows you to divide your users into different segments based on a persistent cookie. You can configure rules using either the Cloudlets Policy Manager application or the [Cloudlets API](#).

Options

- `status` (*boolean*): Enables the Audience Segmentation cloudlet feature. (In the Beta API, please substitute "on" and "off" string values.)
- `nimbus_policy_token` (*object*): Specifies the name of the audience segmentation policy, using alphanumeric and underscore characters. The token serves as a string prefix in forming the cookie name.
- `label` (*string*): Specifies a suffix to append to the cookie name. This helps distinguish this audience segmentation policy from any others within the same property.
- `segmentTrackingMethod` (*enum string*): Specifies the method to pass segment information to the origin. The Cloudlet passes the rule applied to a given request either `in_cookie_header` , `in_query_param` , `in_custom_header` , or the default, `never` .
- `segmentTrackingQueryParam` (*string*): With `segmentTrackingMethod` set to `in_query_param` , this query parameter specifies the name of the segmentation rule.
- `segmentTrackingCookieName` (*string*): With `segmentTrackingMethod` set to `in_cookie_header` , this cookie name specifies the name of the segmentation rule.

- `segmentTrackingCustomHeader` (*string*): With `segmentTrackingMethod` set to `in_custom_header`, this custom HTTP header specifies the name of the segmentation rule.
- `segment_cookie_expiry_type` (*enum string*): Specifies when the segmentation cookie expires, either `on_browser_close`, `never`, or based on a specific `duration`.
- `segment_cookie_duration` (*duration string*): With `segment_cookie_expiry_type` set to `duration`, specifies the lifetime of the segmentation cookie.
- `rolling_segment` (*boolean*): If disabled, sets the expiration time only if the cookie is not yet present in the request. (In the Beta API, please substitute "1" and "0" string values.)

baseDirectory

Prefix URLs sent to the origin with a base path.

For example, with an origin of `example.com`, setting the `basedir` to `/images` sets the origin's base path to `example.com/images`. Any request for a `my_pics/home.jpg` file resolves on the origin server to `example.com/images/my_pics/home.jpg`.

Note that changing the origin's base path also causes a change to the cache key. Until that resolves, it may cause a traffic spike to your origin server.

Options

- `basedir` (*string*): Specifies the base path of content on your origin server. The value must begin and end with a slash (/) character, for example `/parent/child/`.

Feature previously named: `basedir`

Related behaviors: [cacheError](#), [cacheKeyQueryParams](#), [redirect](#), [modifyOutgoingResponseHeader](#), [cacheKeyIgnoreCase](#), [rewriteUrl](#)

breakConnection

A [read-only behavior](#) that simulates an origin connection problem, typically to test an accompanying [failAction](#) policy.

Options

- `status` (*boolean*): Enables the break connection behavior. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `breakconnect`

Related behaviors: [failAction](#) , [cacheKeyQueryParams](#) , [redirect](#) , [cacheError](#) , [healthDetection](#) , [webApplicationFirewall](#)

cacheError

Caches the origin's error responses to decrease server load. Applies for 10 seconds by default to the following HTTP codes: `204` , `305` , `400` , `404` , `405` , `501` , `502` , `503` , `504` , and `505` .

Options

- `enabled` (*boolean*): When enabled, activates the error-caching behavior. (In the Beta API, please substitute "true" and "false" string values.)
- `ttl` (*duration string*): Overrides the default caching duration of `10s` . Note that if set to `0` , it is equivalent to `no-cache` , which forces revalidation and may cause a traffic spike. This can be counterproductive when, for example, the origin is producing an error code of `500` .
- `preservestale` (*boolean*): When enabled, the edge server preserves stale cached objects when the origin returns `400` , `500` , `502` , `503` , and `504` error codes. This avoids re-fetching and re-caching content after transient errors. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `negativettl`

Related behaviors: `cacheRedirect` , `cacheKeyQueryParams` , `cacheKeyIgnoreCase` , `removeVary` , `redirect` , `denyAccess`

cacheId

Controls which query parameters, headers, and cookies are included in or excluded from the cache identifier.

Options

- `rule` (*enum string*): Specifies how to modify the cache ID:
 - `includehd` includes specified HTTP headers in the cache ID.
 - `includeck` includes specified cookies in the cache ID.
 - `includeallqs` includes all query parameters when forming a cache ID.
 - `includeqs` includes the specified set of query parameters when forming a cache ID.
 - `excludeqs` excludes the specified set of query parameters when forming a cache ID.
 - `includeurl` includes the full URL, the same as the default without the `cacheid` behavior.
- `value` (*enum string*): If set to `nameandvalue` , includes the value of the specified elements in the cache ID. Otherwise if set to `nameonly` , only their names are included.
- `optional` (*boolean*): When enabled, requires the behavior's specified elements to be present for content to cache. When disabled, requests that lack the specified elements are still cached. This option only applies when the `type` is `includeck` , `includeqs` , `includehd` , or `excludeqs` . (In the Beta API, please substitute "on" and "off" string values.)
- `elementlist` (*array of string values*): Specifies the names of the query parameters, cookies, or headers to include or exclude from the cache ID. (Only applies when the `rule` is `includeck` , `includehd` , `includeqs` , or `excludeqs` .)

Feature previously named: `cacheid`

Related behaviors: `redirect` , `cacheKeyQueryParams` , `removeVary` , `cacheError` , `webApplicationFirewall` , `modifyOutgoingResponseHeader`

cacheKeyIgnoreCase

By default, cache keys are generated under the assumption that path and filename components are case-sensitive, so that `File.html` and `file.html` use separate cache keys. Enabling this behavior forces URL components whose case varies to resolve to the same cache key. Enable this behavior if your origin server is already case-insensitive, such as those based on Microsoft IIS.

With this behavior enabled, make sure any child rules do not match case-sensitive path components, or you may apply different settings to the same cached object.

Note that if already enabled, disabling this behavior potentially results in new sets of cache keys. Until these new caches are built, your origin server may experience traffic spikes as requests pass through. It may also result in *cache pollution*, excess cache space taken up with redundant content.

If you're using [NetStorage](#) in conjunction with this behavior, also set NetStorage's **Force Case** option to match it, and make sure you name the original files consistently as either upper- or lowercase.

Options

- `ignore_case` (*boolean*): When enabled, ignores case when forming cache keys. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `cachekeyignorecase`

Related behaviors: `cacheError` , `cacheKeyQueryParams` , `cacheRedirect` , `removeVary` , `dnsAsyncRefresh` , `redirect`

cacheKeyQueryParams

By default, cache keys are formed as URLs with full query strings. This behavior allows you to consolidate cached objects based on specified sets of query parameters.

Note also that whenever you apply behavior that generates new cache keys, your origin server may experience traffic spikes before the new cache starts to serve out.

Options

- `behavior` (*enum string*): Configures how sets of query string parameters translate to cache keys:
 - `ignore-all` causes query string parameters to be ignored when forming cache keys.
 - `ignore` or `include` makes the key depend on the sequence of values in the `parameters` field.
 - `include-all-preserve-order` forms a separate key for the entire set of query parameters, but sensitive to the order in which they appear. (For example, `?q=akamai&state=ma` and `?state=ma&q=akamai` cache separately.)
 - `include-all-alphabetize-order` forms keys for the entire set of parameters, but the order doesn't matter. The examples above both use the same cache key.

Be careful when applying `behavior` not to ignore any parameters that result in substantially different content, as it is *not* reflected in the cached object.

- `parameters` (*array of string values*): With `behavior` set to `include` or `ignore`, `parameters` specifies the set of parameter field names to include in or exclude from the cache key. By default, these match the field names as string prefixes.
- `exact_match` (*boolean*): When enabled, `parameters` must match exactly. Keep disabled to match string prefixes. (In the Beta API, please substitute `"yes"` and `"no"` string values.)

Feature previously named: `cachekeyqueryparams`

Related behaviors: [cacheError](#), [cacheKeyIgnoreCase](#), [redirect](#), [removeVary](#), [cacheRedirect](#), [denyAccess](#)

cacheKeyRewrite

This [read-only behavior](#) rewrites a default cache key's path. Contact Akamai Professional Services for help configuring it.

WARNING: This feature is in Beta, so please test thoroughly.

Options

- `purgekey` (*string*): Specifies the new cache key path as an alphanumeric value.

Feature previously named: `cachekeyrewrite`

cachePost

By default, POST requests are passed to the origin. This behavior overrides the default, and allows you to cache POST responses.

Options

- `enabled` (*boolean*): Enables caching of POST responses. (In the Beta API, please substitute "on" and "off" string values.)
- `usebody` (*enum string*): Define how and whether to use the POST message body as a cache key:
 - `ignore` uses only the URL to cache the response.
 - `MD5` adds a string digest of the data as a query parameter to the cache URL.
 - `query` adds the raw request body as a query parameter to the cache key, but only if the POST request's `Content-Type` is `application/x-www-form-urlencoded`. (Use this in conjunction with [cacheId](#) to define relevant query parameters.)

Feature previously named: `postcaching`

Related behaviors: [modifyOutgoingResponseHeader](#) , [removeVary](#) , [redirect](#) , [cacheKeyQueryParams](#) , [cacheError](#) , [dnsAsyncRefresh](#)

cacheRedirect

Caches HTTP 302 redirect responses. By default, Akamai edge servers cache HTTP 302 redirects depending on their `Cache-Control` or `Expires` Headers. Enabling this behavior instructs edge servers to cache 302 redirects the same as they would for HTTP 200 responses.

Options

- `enabled` (*boolean*): Enables the redirect caching behavior. (In the Beta API, please substitute `"true"` and `"false"` string values.)

Feature previously named: `cache302`

Related behaviors: [cacheError](#) , [cacheKeyIgnoreCase](#) , [cacheKeyQueryParams](#) , [removeVary](#) , [redirect](#) , [dnsAsyncRefresh](#)

caching

Control content caching on edge servers: whether or not to cache, whether to honor the origin's caching headers, and for how long to cache.

NOTE: Any `no-store` or `bypass-cache` HTTP headers set on the origin's content overrides this behavior.

Options

- `behavior` (*enum string*): Specify the caching option:
 - `no-store` clears the cache and serves from the origin.

- `bypass-cache` retains the cache but serves from the origin.
- Honor the origin's `max-age` header
- Honor the origin's `cc` header
- Honor the origin's `expires` header
- Honor `both` the origin's `cc` and `expires` header, whichever comes last.
- `mustrevalidate` (*boolean*): Determines what to do once the cached content has expired, by which time the Akamai platform should have re-fetched and validated content from the origin. If enabled, only allows the re-fetched content to be served. If disabled, may serve stale content if the origin is unavailable. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `ttl` (*duration string*): The maximum time content may remain cached. Setting the value to `0` is the same as setting a `no-cache` header, which forces content to revalidate.
- `defaultttl` (*duration string*): Set the `max-age` header for the cached content.

centralAuthorization

Forward client requests to the origin server for authorization, along with optional `Set-Cookie` headers, useful when you need to maintain tight access control. The edge server forwards an `If-Modified-Since` header, to which the origin needs to respond with a `304` (Not-Modified) HTTP status when authorization succeeds. If so, the edge server responds to the client with the cached object, since it does not need to be re-acquired from the origin.

Options

- `status` (*boolean*): Enables the centralized authorization behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `centralauth`

Related behaviors: [cacheKeyIgnoreCase](#) , [prefreshCache](#) , [cacheError](#) , [cacheKeyQueryParams](#) , [redirect](#) , [validateEntityTag](#)

chaseRedirects

Controls whether the edge server chases any redirects served from the origin.

Options

- `status` (*boolean*): When enabled, allows edge servers to chase redirects. (In the Beta API, please substitute "true" and "false" string values.)
- `limit` (*string*): Specifies, as a string, the maximum number of redirects to follow.
- `serve404` (*boolean*): Once the redirect `limit` is reached, enabling this option serves an HTTP 404 (Not Found) error instead of the last redirect. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `chaseredirects`

Related behaviors: [cacheError](#) , [cacheRedirect](#) , [dnsAsyncRefresh](#) , [rewriteUrl](#) , [prefreshCache](#) , [cacheKeyQueryParams](#)

constructResponse

A [read-only behavior](#) that constructs an HTTP response, complete with HTTP status code and body, to serve from the edge independently of your origin. Contact Akamai Professional Services for help configuring it.

Options

- `status` (*boolean*): When enabled, serves the custom response. Enabling the behavior evicts the cached object associated with the request, since it is not being served. (In

the Beta API, please substitute "on" and "off" string values.)

- `body` (*string*): HTML response of up to 2000 characters to send to the end-user client.
- `responsecode` (*enum string*): The HTTP response code to send to the end-user client, either 200 or 404 .

Feature previously named: `construct_response`

continuousDeployment

The Phased Release Cloudlet provides gradual and granular traffic management to an alternate origin in near real time. Use the [Cloudlets API](#) or the Cloudlets Policy Manager application within Control Center to set up your Cloudlets policies.

Options

- `enabled` (*boolean*): Enables the Phased Release Cloudlet.
- `cloudletPolicy` (*object*): Specifies the Cloudlet policy as an object containing two members: a descriptive `name` string and an integer `id` set by the Cloudlet application.
- `label` (*string*): A label to distinguish this Phased Release policy from any others within the same property.
- `populationCookieType` (*enum string*): For the population of users the Cloudlet defines, select whether to assign a cookie to them, or else `NONE` . Other option values specify when the cookie expires: after a specific `DURATION` , at a `FIXED_DATE` , once the browser session ends (`ON_BROWSER_CLOSE`), or `NEVER` . If you select the Cloudlet's *random* membership option, it overrides the option value so that it is effectively `NONE` .
- `populationExpirationDate` (*ISO 8601 format date/time string*): With the `populationCookieType` set to `FIXED_DATE` , this specifies the date and time when membership expires, and the browser no longer sends the cookie. Subsequent requests re-evaluate based on current membership settings.

- `populationDuration` (*duration string*): With the `populationCookieType` set to `DURATION`, this sets the lifetime of the cookie from the initial request. Subsequent requests re-evaluate based on current membership settings.
- `populationRefresh` (*boolean*): With the `populationCookieType` set to `DURATION`, enabling this option resets the original duration of the cookie if the browser refreshes before the cookie expires.
- `failoverEnabled` (*boolean*): Allows failure responses at the origin defined by the Cloudlet to fail over to the prevailing origin defined by the property.
- `failoverResponseCode` (*array of string values*): With `failoverEnabled` on, this defines the set of failure codes that initiate the failover response.
- `failoverDuration` (*number within 0-100 range*): Specifies the number of seconds to wait until the client tries to access the failover origin after the initial failure is detected. Set the value to `0` to immediately request the alternate origin upon failure.

cpCode

Content Provider Codes (CP codes) allow you to distinguish various reporting and billing segments. You receive a CP code when purchasing Akamai service, and you need it to access properties. This behavior allows you to apply any valid CP code, including additional ones you may request from Akamai Professional Services. For a CP code to be valid, it must belong to the same contract and be associated with the same product as the property, and the group must have access to it.

Options

- `cpcode` (*object*): Specifies a `cpcode` object, which includes an `id` key and a descriptive `name`:

```
"cpcode": {
  "id" : 12345,
  "name" : "my cpcode"
}
```

Feature previously named: `cpcode`

deliveryReceipt

A static behavior that's required when specifying the Cloud Monitor module's (`edgeConnect`) behavior.

Options

- `status` (*read-only string*): When `on` , enables the behavior.

NOTE: You can only apply this behavior if the property is marked as secure. See [Secure Property Requirements](#) for guidance.

Feature previously named: `receiptdelivery`

denyAccess

Denies access assuming a condition in the rule matches. For example, a `userLocation` match paired with the `denyaccess` behavior would deny requests from a specified part of the world.

NOTE: By keying on the value of the `reason` option, `denyaccess` behaviors can override each other when called from nested rules. For example, a parent rule might deny access to a certain geographic area, citing "location" as the `reason` , but another nested rule can then allow access for a set of IPs within that area, so long as the `reason` matches.

Options

- `behavior` (*boolean*): Either `allow` or `deny` access. (In the Beta API, please substitute "deny" and "allow" string values.)

- `reason` (*string*): Text message that keys why access is denied. Any subsequent `denyaccess` behaviors within the rule tree may refer to the same `reason` key to override the current `behavior`.

Feature previously named: `denyaccess`

Related behaviors: `redirect`, `cacheError`, `cacheKeyQueryParams`, `webApplicationFirewall`, `modifyOutgoingResponseHeader`, `removeVary`

deviceCharacteristicCacheId

By default, source URLs serve as cache IDs on edge servers. Electronic Data Capture allows you to specify an additional set of device characteristics to generate separate cache keys. Use this in conjunction with the `deviceCharacteristicHeader` behavior.

Options

- `edc_elementlist` (*array of string values*): Specifies a set of information about the device with which to generate a separate cache key, any of the following values:

<code>accept_third_party_cookie</code>	<code>mobile_browser</code>
<code>ajax_preferred_geoloc_api</code>	<code>mobile_browser_version</code>
<code>ajax_support_javascript</code>	<code>model_name</code>
<code>brand_name</code>	<code>pdf_support</code>
<code>cookie_support</code>	<code>physical_screen_height</code>
<code>device_os</code>	<code>physical_screen_width</code>
<code>device_os_version</code>	<code>png</code>
<code>dual_orientation</code>	<code>preferred_markup</code>
<code>flash_lite_version</code>	<code>resolution_height</code>
<code>full_flash_support</code>	<code>resolution_width</code>
<code>gif_animated</code>	<code>viewport_initial_scale</code>
<code>html_preferred_dtd</code>	<code>viewport_width</code>
<code>is_mobile</code>	<code>xhtml_file_upload</code>
<code>is_tablet</code>	<code>xhtml_preferred_charset</code>
<code>is_wireless_device</code>	<code>xhtml_support_level</code>
<code>jpg</code>	<code>xhtml_supports_iframe</code>
<code>marketing_name</code>	<code>xhtml_supports_table_for_layout</code>
<code>max_image_height</code>	<code>xhtml_table_support</code>
<code>max_image_width</code>	<code>xhtmlmp_preferred_mime_type</code>

Feature previously named: `edccacheid`

Related behaviors: [deviceCharacteristicHeader](#) , [redirect](#) , [modifyOutgoingResponseHeader](#) , [removeVary](#) , [cached](#) , [frontEndOptimization](#)

deviceCharacteristicHeader

Sends selected information about requesting devices to the origin server, in the form of an `X-Akamai-Device-Characteristics` HTTP header. Use in conjunction with the [deviceCharacteristicCached](#) behavior.

Options

- `edc_elementlist` (*array of string values*): Specifies the set of information about the requesting device to send to the origin server, any of the following:

<code>accept_third_party_cookie</code>	<code>mobile_browser</code>
<code>ajax_preferred_geoloc_api</code>	<code>mobile_browser_version</code>
<code>ajax_support_javascript</code>	<code>model_name</code>
<code>brand_name</code>	<code>pdf_support</code>
<code>cookie_support</code>	<code>physical_screen_height</code>
<code>device_os</code>	<code>physical_screen_width</code>
<code>device_os_version</code>	<code>png</code>
<code>dual_orientation</code>	<code>preferred_markup</code>
<code>flash_lite_version</code>	<code>resolution_height</code>
<code>full_flash_support</code>	<code>resolution_width</code>
<code>gif_animated</code>	<code>viewport_initial_scale</code>
<code>html_preferred_dtd</code>	<code>viewport_width</code>
<code>is_mobile</code>	<code>xhtml_file_upload</code>
<code>is_tablet</code>	<code>xhtml_preferred_charset</code>
<code>is_wireless_device</code>	<code>xhtml_support_level</code>
<code>jpg</code>	<code>xhtml_supports_iframe</code>
<code>marketing_name</code>	<code>xhtml_supports_table_for_layout</code>
<code>max_image_height</code>	<code>xhtml_table_support</code>
<code>max_image_width</code>	<code>xhtmlmp_preferred_mime_type</code>

Feature previously named: `edcheader`

Related behaviors: [dnsAsyncRefresh](#) , [cacheKeyIgnoreCase](#) , [cacheError](#) , [deviceCharacteristicCached](#) , [redirect](#) , [enhancedAkamaiProtocol](#)

dnsAsyncRefresh

Allow an edge server to use an expired DNS record when forwarding a request to your origin. The *type A* DNS record refreshes *after* content is served to the end user, so there is no wait for the DNS resolution. Avoid this behavior if you want to be able to disable a server immediately after its DNS record expires.

Options

- `status` (*boolean*): When enabled, allows edge servers to refresh an expired DNS record after serving content. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `timeout` (*duration string*): Set the maximum allowed time an expired DNS record may be active.

Feature previously named: `dnsasyncrefresh`

Related behaviors: `cacheKeyIgnoreCase` , `cacheError` , `removeVary` , `cacheKeyQueryParams` , `cacheRedirect` , `redirect`

dnsPrefresh

A [read-only behavior](#) that allows edge servers to refresh your origin's DNS record independently from end-user requests. The *type A* DNS record refreshes before the origin's DNS record expires.

Options

- `status` (*boolean*): When enabled, allows edge servers to refresh DNS records before they expire. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `delay` (*duration string*): Specifies the amount of time following a DNS record's expiration to asynchronously prefetch it.

- `timeout` (*duration string*): Specifies the amount of time to prefetch a DNS entry if there have been no requests to the domain name.

Feature previously named: `dnsprefresh`

Related behaviors: `dnsAsyncRefresh` , `removeVary` , `redirect` , `webApplicationFirewall` , `cacheError` , `cacheKeyIgnoreCase`

downgradeProtocol

Serve static objects to the end-user client over HTTPS, but fetch them from the origin via HTTP.

Options

- `status` (*boolean*): Enables the protocol downgrading behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `protocoldowngrade`

downstreamCache

Specify the caching instructions the edge server sends to the end user's client or client proxies. By default, the cache's duration is whichever is less: the remaining lifetime of the edge cache, or what the origin's header specifies. If the origin is set to `no-store` or `bypass-cache` , edge servers send *cache-busting* headers downstream to prevent downstream caching.

Options

- `behavior` (*enum string*): Specify the caching instructions the edge server sends to the end user's client. It accepts the following values:

- `allow` : The value of `allow_behavior` chooses the caching method and headers to send to the client.
- `must-revalidate` : This equates to a `Cache-Control: no-cache` header, which allows caching but forces the client browser to send an `if-modified-since` request each time it requests the object.
- `bust` : Sends cache-busting headers downstream.
- `tunnel-origin` : This passes `Cache-Control` and `Expires` headers from the origin to the downstream client.
- `none` : Don't send any caching headers. Allow client browsers to cache content according to their own default settings.
- `allow_behavior` (*enum string*): Specify how the edge server calculates the downstream cache by setting the value of the `Expires` header:
 - `from-value` sends the value of the edge cache's duration.
 - `from-max-age` sends the `cache:max-age` value applied to the object, without evaluating the cache's duration.
 - `lesser` sends the lesser value of what the origin specifies and the edge cache's remaining duration. This is the default behavior.
 - `greater` sends the greater value of what the origin specifies and the edge cache's remaining duration.
 - `remaining-lifetime` sends the value of the edge cache's remaining duration, without comparing it to the origin's headers.
 - `pass-origin` sends the value of the origin's header, without evaluating the edge cache's duration.
- `ttl` (*duration string*): Set the duration of the cache. Setting the value to `0` equates to a `no-cache` header that forces revalidation.
- `headers_sent` (*enum string*): Specifies the HTTP headers to include in the response to the client:
 - `pass-origin` sends the same set of `Cache-Control` and `Expires` headers received from the origin.
 - `cc-only` sends only the origin's `Cache-Control` header.
 - `expires-only` sends only the origin's `Expires` header.

- `both` sends `Cache-Control` and `Expires` header.
- `none` strips both headers.
- `send_private` (*boolean*): When enabled, adds a `Cache-Control: private` header to prevent objects from being cached in a shared caching proxy. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `downstreamcaching`

Related behaviors: `allowPost` , `removeVary` , `cacheKeyIgnoreCase` , `cacheError` , `dnsAsyncRefresh` , `gzipResponse`

edgeConnect

Configures traffic logs for the Cloud Monitor push API.

Options

- `publish` (*boolean*): Enables Cloud Monitor's log-publishing behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `api_connector` (*enum string*): Describes the API connector type, either `default` , `bmc_apm` , or `siem_json` .
- `api_data_elements` (*array of string values*): Specifies the data set to log, any of the following values:

<code>apm</code>	<code>networkv1</code>	<code>seccratedenyv2</code>
<code>geo</code>	<code>reqheader</code>	<code>seccratewarnv2</code>
<code>http</code>	<code>resheader</code>	
<code>networkperformance</code>	<code>secappv2</code>	

- `dest_host` (*string*): Specifies the target hostname accepting push API requests.
- `dest_path` (*string*): Specifies the push API's endpoint.
- `override_aggregate_settings` (*boolean*): When enabled, overrides default log settings. (In the Beta API, please substitute `"on"` and `"off"` string values.)

- `aggregate_time` (*duration string*): With `override_aggregate_settings` enabled, specifies how often logs generate.
- `aggregate_lines` (*string*): With `override_aggregate_settings` enabled, specifies the maximum number of lines to include in each log.
- `aggregate_size` (*string*): With `override_aggregate_settings` enabled, specifies the log's maximum size.

Feature previously named: `edgeconnect`

Related behaviors: `webApplicationFirewall` , `redirect` , `modifyOutgoingResponseHeader` , `cacheKeyQueryParams` , `cacheError` , `allowPut`

edgeImageConversion

The Edge Image Converter offloads various image manipulation tasks to edge servers. This behavior specifies various predefined policies to apply.

Options

- `eic_bool` (*boolean*): Enables the edge image conversion behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `failover_option` (*boolean*): If the request results in a server error, enabling this forwards it to the origin. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `policy_bool` (*boolean*): Enables a specified set of image conversion policies. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `op_policy_table` (*array*): With `policy_bool` enabled, specifies commands that when present or not in the query string release an error code.
- `op_policy_violation` (*enum string*): Specifies the HTTP error code if any `op_policy_table` conditions match, either `400` , `403` , `404` , or `409` .

Feature previously named: `edge_image_converter`

Related behaviors: `cacheError` , `rewriteUrl` , `edgeScape` , `failAction` , `enhancedAkamaiProtocol` , `redirect`

edgeLoadBalancingAdvanced

A [read-only behavior](#) that implements customized Edge Load Balancing features. Contact Akamai Professional Services for help configuring it.

Options

- `description` (*string*): A description of what the `xml` block does.
- `xml` (*string*): A block of Akamai XML metadata.

Feature previously named: `elb_advanced`

edgeLoadBalancingDataCenter

The Edge Load Balancing module allows you to specify groups of data centers that implement load balancing, session persistence, and real-time dynamic failover. Enabling ELB routes requests contextually based on location, device, or network, along with optional rules you specify.

This behavior specifies details about a data center, and must be paired in the same rule with an [edgeLoadBalancingOrigin](#) behavior, which specifies its origin. An *origin* is an abstraction that helps group a logical set of a website or application. It potentially includes information about many data centers and cloud providers, as well as many end points or IP addresses for each data center. More than one data center can thus refer to the same origin.

Options

- `origin_id` (*string*): Corresponds to the `id` specified by the [edgeLoadBalancingOrigin](#) behavior associated with this data center.
- `alias` (*string*): Provides a description for the ELB data center, for your own reference.
- `host` (*string*): Specifies the data center's hostname.
- `name` (*string*): If using session persistence, this specifies the value of the cookie named in the corresponding [edgeLoadBalancingOrigin](#) behavior's `cookie_name` option.
- `enable_dc_failover` (*boolean*): When enabled, allows you to specify failover rules. (In the Beta API, please substitute "ON" and "OFF" string values.)
- `ip_address` (*string*): With `enable_dc_failover` enabled, specifies this data center's IP address.
- `failover_rules` (*array*): With `enable_dc_failover` enabled, provides up to four failover rules to apply in the specified order. These rules appear as objects with the following fields:
 - `modify_request` (*boolean*): When enabled, allows you to modify the request's hostname or path.
 - `override_hostname` (*boolean*): When enabled, overrides the request's hostname with the `failover_hostname`.
 - `failover_hostname`: The hostname of the data center to fail over to.
 - `context_root`: Specifies the path to use in the forwarding request, typically the root (/) when failing over to a different data center, or a full path such as `/static/error.html` when failing over to an error page.
 - `absolute_path` (*boolean*): If enabled, the path specified by `context_root` is interpreted as an absolute server path, for example to reference a site-down page. If disabled, the path is appended to the request.

(In the API's beta version, please substitute boolean values with "true" or "false" string values.)

Feature previously named: `elb_data_center`

Related behaviors: [edgeLoadBalancingOrigin](#), [cacheKeyQueryParams](#), [redirect](#), [modifyIncomingRequestHeader](#), [personallyIdentifiableInformation](#), [modifyIncomingResponseHeader](#)

edgeLoadBalancingOrigin

The Edge Load Balancing module allows you to implement groups of data centers featuring load balancing, session persistence, and real-time dynamic failover. Enabling ELB routes requests contextually based on location, device, or network, along with optional rules you specify.

This behavior specifies the data center's origin, and must be paired in the same rule with at least one [edgeLoadBalancingDataCenter](#) behavior, which provides details about a particular data center. An *origin* is an abstraction that helps group a logical set of a website or application. It potentially includes information about many data centers and cloud providers, as well as many end points or IP addresses for each data center. To specify an ELB origin, you must have configured an [origin](#) behavior whose `type` is set to `elb_origin_group`.

Options

- `id` (*string*): Specifies a unique descriptive string for this ELB origin. The value must match the `origin_id` specified by the [edgeLoadBalancingDataCenter](#) behavior associated with this origin.
- `name` (*string*): Provides a description for the ELB origin, for your own reference.
- `host` (*string*): Specifies the hostname associated with the ELB rule.
- `enable_session_persistence` (*boolean*): When enabled, allows you to specify a cookie to pin the user's browser session to one data center. When disabled, ELB's default load balancing may send users to various data centers within the same session. (In the Beta API, please substitute "ON" and "OFF" string values.)
- `cookie_name` (*string*): With `enable_session_persistence` enabled, this specifies the name of the cookie that marks users' persistent sessions. (The accompanying [edgeLoadBalancingDataCenter](#) behavior's `name` option specifies the cookie's value.)

Feature previously named: `elb_origin`

Related behaviors: [edgeLoadBalancingDataCenter](#), [cacheKeyQueryParams](#), [redirect](#), [modifyIncomingRequestHeader](#), [personallyIdentifiableInformation](#), [modifyIncomingResponseHeader](#)

edgeOriginAuthorization

Allows the origin server to use a cookie to ensure requests from Akamai servers are genuine.

NOTE: This behavior requires that you specify the cookie's domain name, so it is best to deploy within a match of the hostname. It does not work properly when the origin server accepts more than one hostname (for example, using virtual servers) that do not share the same top-level domain.


Options

- `enabled` (*boolean*): Enables the cookie-authorization behavior. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `name` (*string*): Specifies the name of the cookie to use for authorization.
- `value` (*string*): Specifies the value of the authorization cookie.
- `domain` (*string*): Specify the cookie's domain, which must match the top-level domain of the `Host` header the origin server receives.

Feature previously named: `edgeoriginauth`

Related behaviors: [cacheError](#) , [removeVary](#) , [cacheKeyIgnoreCase](#) , [dnsAsyncRefresh](#) , [cacheRedirect](#) , [denyAccess](#)

edgeScape

[EdgeScape](#)  allows you to customize content based on the end user's geographic location or connection speed. When enabled, the edge server sends a special `X-Akamai-Edgescape` header to the origin server encoding relevant details about the end-user client as key-value pairs.

Options

- `status` (*boolean*): When enabled, sends the `X-Akamai-Edgescape` request header to the origin. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `edgescape`

Related behaviors: `redirect` , `cacheError` , `denyAccess` , `cacheKeyQueryParams` , `removeVary` , `cacheRedirect`

edgeSideIncludes

Allows edge servers to process edge side include (ESI) code to generate dynamic content. Since this behavior requires more parsing time, you should not apply it to pages with no ESI code, or to any non-HTML content.

Options

- `status` (*boolean*): Enables ESI processing. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `responseheaders` (*boolean*): Enable ESI only for content featuring the following HTTP response header: `Edge-control: dca=esi` (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `passsetcookie` (*boolean*): When enabled, allows edge servers to pass your origin server's cookies to the ESI processor. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `passclientip` (*boolean*): When enabled, allows edge servers to pass the client IP header to the ESI processor. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `i18nstatus` (*boolean*): When enabled, provides internationalization support for ESI. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `i18ncharset` (*array of string values*): With `i18nstatus` enabled, specifies the character sets to use when transcoding the ESI language, `UTF-8` and `ISO-8859-1` for example.

Feature previously named: `esi`

Related behaviors: [cacheKeyQueryParams](#) , [redirect](#) , [modifyOutgoingResponseHeader](#) , [rewriteUrl](#) , [cacheError](#) , [denyAccess](#)

edge_redirector

This behavior enables the [Edge Redirector Cloudlet](#) application, which is designed to help you manage large numbers of redirects. Assuming cloudlets are available on your contract, choose **Configure⇒Cloudlets** to control the Edge Redirector within Control Center. Otherwise use the [Cloudlets API](#) to configure it programmatically.

Options

- `status` (*boolean*): Enables the Edge Redirector Cloudlet. (In the Beta API, please substitute "on" and "off" string values.)
- `nimbus_policy_token` (*string*): Specifies the name of the redirect policy, using alphanumeric and underscore characters.

Related behaviors: [rewriteUrl](#) , [cacheRedirect](#) , [cacheError](#) , [prefreshCache](#) , [edgeScape](#) , [modifyOutgoingResponseHeader](#)

enhancedAkamaiProtocol

Enables the Enhanced Akamai Protocol, a suite of advanced routing and transport optimizations that increase your website's performance and reliability. It is only available to specific applications, and requires a special routing from edge to origin.

WARNING: Disabling this behavior may significantly reduce a property's performance.

This behavior does not include any options. Specifying the behavior itself enables it.

Feature previously named: `enhancedakamaiprotocol`

Related behaviors: `cacheError` , `webApplicationFirewall` , `cacheKeyIgnoreCase` , `redirect` , `cacheKeyQueryParams` , `dnsAsyncRefresh`

failAction

Specifies how to respond when the origin is not available: by serving stale content, by serving an error page, or by redirecting.

Options

- `status` (*boolean*): When enabled in case of a failure to contact the origin, the current behavior applies. (In the Beta API, please substitute "on" and "off" string values.)
- `type` (*enum string*): Specifies the basic action to take when there is a failure to contact the origin:
 - `servestale` serves content that is already in the cache.
 - `redirect` specifies a redirect action. (Use with these options: `redirecthostname` , `redirecthostname` , `redirectcustompath` , `redirectpath` , `redirectmethod` , `modifyproto` , and `proto` .)
 - `recreatedcex` serves alternate content from an external network. (Use with these options: `cexhostname` , `cexcustompath` , and `cexpath` .)
 - `recreatedco` serves alternate content from your network. (Use with these options: `cohostname` , `cocustompath` , and `copath` .)
 - `recreatedns` serves [NetStorage](#) content. (Use with these options: `nshostname` , `nspath` , and `cpcode` .)
 - `dynamic` allows you to serve dynamic SaaS content. (Use with these options: `dynamic_method` , `dynamiccustompath` , `saas_type` , `saas_suffix` , `saas_regex` , and `saas_replace` .)
- `cpcode` (*object*): When `type` is `recreatedns` , this specifies a *cpcode* for which to log errors for the NetStorage location. It appears as an object that includes an `id` key and a descriptive `name` :

```
"cpcode": {
  "id" : 12345,
  "name" : "my cpcode"
}
```

- `nshostname` (*object*): When the `type` is `recreatedns` , specifies the [NetStorage](#) origin to serve the alternate content. For example:

```
"nshostname": {
  "id" : "id_string",
  "name" : "Example Downloads",
  "downloadDomainName" : "example.download.akamai.com",
  "cpCode" : 12345
}
```

NOTE: Contact Akamai Professional Services for your NetStorage origin's `id` .

- `nspath` (*string*): When the `type` is `recreatedns` , specifies the path for the [NetStorage](#) request.
- `redirectmethod` (*enum string*): When the `type` is `redirect` , specifies the HTTP response code, either `301` or `302` .
- `redirecthostnametype` (*enum string*): When the `type` is `redirect` , this specifies whether to preserve the `original` hostname in the redirect, or whether to use an `alternate` hostname specified with `redirecthostname` .
- `redirecthostname` (*string*): When the `type` is `redirect` and the `redirecthostnametype` is `alternate` , this specifies the hostname for the redirect.
- `redirectcustompath` (*boolean*): When the `type` is `redirect` , enabling this allows you use `redirectpath` to customize a new path. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `redirectpath` (*string*): When the `type` is `redirect` , this specifies a new path.
- `modifyproto` (*boolean*): When the `type` is `redirect` , enabling this allows you to modify the redirect's protocol using the value of the `proto` field. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `proto` (*enum string*): When the `type` is `redirect` and `modifyproto` is enabled, this specifies the redirect's protocol, either `HTTP` or `HTTPS` .
- `preservequerystring` (*boolean*): When using `cocustompath` , `cexcustompath` , or `redirectcustompath` to specify a custom path, enabling this passes in the original

request's query string as part of the path. (In the Beta API, please substitute "on" and "off" string values.)

- `cexhostname` (*string*): When `type` is `recreatedcex`, this specifies a hostname.
- `cexcustompath` (*boolean*): When `type` is `recreatedcex`, enabling this allows you to specify a custom path. (In the Beta API, please substitute "on" and "off" string values.)
- `cexpath` (*string*): When `type` is `recreatedcex` and `cexcustompath` is enabled, this specifies a custom path.
- `cohostname` (*string*): When the `type` is `recreatedco`, specifies the static hostname for the alternate redirect.
- `cocustompath` (*boolean*): When the `type` is `recreatedco`, enabling this allows you to specify a custom redirect path. (In the Beta API, please substitute "on" and "off" string values.)
- `copath` (*string*): When the `type` is `recreatedco` and `cocustompath` is enabled, this specifies a custom redirect path.
- `dynamic_method` (*enum string*): With the `type` set to `dynamic`, specifies the redirect method, either `serve-301`, `serve-302`, or `serve-alternate`.
- `dynamiccustompath` (*boolean*): When enabled, allows you to modify the original requested path. (In the Beta API, please substitute "on" and "off" string values.)
- `dynamicpath` (*string*): With `dynamiccustompath` enabled, specifies the new path.
- `saas_type` (*enum string*): Identifies the component of the request that identifies the SaaS dynamic failaction: either `cookie`, `hostname`, `path`, or `query_string`.
- `saas_suffix` (*string*): With `type` set to `dynamic`, specifies the static portion of the SaaS dynamic failaction.
- `saas_regex` (*string*): With `type` set to `dynamic`, specifies the substitution pattern (a Perl-compatible regular expression) that defines the SaaS dynamic failaction.
- `saas_replace` (*string*): With `type` set to `dynamic`, specifies the replacement pattern that defines the SaaS dynamic failaction.
- `saas_cname_enabled` (*boolean*): With the `saas_type` set to `hostname`, specifies whether to use a CNAME chain to determine the hostname for the SaaS dynamic failaction. (In the Beta API, please substitute "on" and "off" string values.)
- `saas_cname_level` (*number*): With `saas_cname_enabled` on, specifies the number of elements in the CNAME chain backwards from the edge hostname that determines

the hostname for the SaaS dynamic failaction.

- `saas_cookie` (*string*): With `saas_type` set to `cookie` , specifies the name of the cookie that identifies this SaaS dynamic failaction.
- `saas_query_string` (*string*): With `saas_type` set to `query_string` , specifies the name of the query parameter that identifies this SaaS dynamic failaction.

Feature previously named: `failaction`

Related behaviors: `cacheKeyQueryParams` , `redirect` , `cacheError` , `rewriteUrl` , `webApplicationFirewall` , `cacheKeyIgnoreCase`

forwardRewrite

The Forward Rewrite Cloudlet allows you to conditionally modify the forward path in edge content without affecting the URL that displays in the user's address bar. If cloudlets are available on your contract, choose **Configure⇒Cloudlets** to control how this feature works, or use the [Cloudlets API](#) to configure it programmatically.

Options

- `status` (*boolean*): Enables the Forward Rewrite Cloudlet behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `nimbus_policy_token` (*string*): Specifies the name of the forward rewrite policy, using alphanumeric and underscore characters.

g2oheader

The *signature header authentication* (g2o) security feature provides header-based verification of outgoing origin requests. Edge servers encrypt request data in a pre-

defined header, which the origin uses to verify that the edge server processed the request. This behavior configures the request data, header names, encryption algorithm, and shared secret to use for verification.

Options

- `status` (*boolean*): Enables the g2o verification behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `data_header` (*string*): Specifies the name of the header that contains the request data that needs encryption.
- `signed_header` (*string*): Specifies the name of the header containing encrypted request data.
- `encoding_version` (*enum string*): Specifies the version of the encryption algorithm as an integer string value from `1` through `5`.
- `sign_string_status` (*enum string*): If set to `default`, the encrypted string is based on the forwarded URL. If set to `custom`, you can use `sign_string` to customize the set of data to encrypt.
- `sign_string` (*array of string values*): If the `sign_string_status` is set to `custom`, specifies the set of data to be encrypted as a combination of concatenated strings, any of the following values:

<code>AK_CLIENT_REAL_IP</code>	<code>AK_HOSTHEADER</code>	<code>AK_SCHEME</code>
<code>AK_DOMAIN</code>	<code>AK_METHOD</code>	<code>AK_URL</code>
<code>AK_EXTENSION</code>	<code>AK_PATH</code>	
<code>AK_FILENAME</code>	<code>AK_QUERY</code>	

- `secret_key` (*string*): Specifies the shared secret key.
- `nonce` (*string*): Specifies the cryptographic *nonce* string.

Related behaviors: [cacheRedirect](#), [verifyTokenAuthorization](#), [removeVary](#)

gzipResponse

Apply *gzip* compression to speed transfer time. The behavior applies best to text content such as HTML, CSS, and JavaScript, especially once it exceeds about 10KB. Do not apply

it to already compressed image formats, or to small files that would add more time to uncompress.

Options

- `behavior` (*enum string*): Specify when to compress responses, either `always` , `never` , or `origin_response` to clients that send an `Accept-Encoding: gzip` header.

Feature previously named: `gzipresponse`

Related behaviors: [cacheKeyIgnoreCase](#) , [downstreamCache](#) , [dnsAsyncRefresh](#) , [cacheKeyQueryParams](#) , [prefetchable](#) , [prefreshCache](#)

hdDataAdvanced

A [read-only behavior](#) that specifies Akamai XML metadata that can only be configured on your behalf by Akamai Professional Services. Unlike the [advanced](#) behavior, this may apply a different set of overriding metadata that executes in a post-processing phase.

Options

- `description` (*string*): Human-readable description of what the XML block does.
- `xml` (*string*): A block of Akamai XML metadata.

Feature previously named: `hddata_advanced`

healthDetection

Monitors the health of your origin server by tracking unsuccessful attempts to contact it. Use this behavior to keep end users from having to wait several seconds before a forwarded request times out, or to reduce requests on the origin server when it is unavailable.

When client requests are forwarded to the origin, the edge server tracks the number of attempts to connect to each IP address. It cycles through IP addresses in least-recently-tested order to avoid hitting the same one twice in a row. If the number of consecutive unsuccessful tests reaches a threshold you specify, the behavior identifies the address as faulty and stops sending requests. The edge server returns an error message to the end user or else triggers any [failAction](#) behavior you specify.

Options

- `ip_retrycount` (*number*): The number of consecutive connection failures that mark an IP address as faulty.
- `ip_retry_interval` (*duration string*): Specifies the amount of time the edge server waits before trying to reconnect to an IP address it has already identified as faulty.
- `max_reconnects` (*number*): Specifies the maximum number of times the edge server contacts your origin server. If your origin is associated with several IP addresses, `max_reconnects` effectively overrides the value of `ip_retrycount`.

Feature previously named: `healthdetect`

Related behaviors: [failAction](#), [cacheKeyQueryParams](#), [redirect](#), [cacheError](#), [webApplicationFirewall](#), [cacheKeyIgnoreCase](#)

http2

Enables the HTTP/2 protocol, which reduces latency and improves efficiency.

This behavior does not include any options. Specifying the behavior itself enables it.

NOTE: You can only apply this behavior if the property is marked as secure. See [Secure Property Requirements](#) for guidance.)

imageManager

Optimizes images' size or file type for the requesting device. You can also use this behavior to generate API tokens to apply your own policies to matching images using the [Image and Video Manager API](#).

Options

- `image_management_enabled` (*boolean*): Enable image management capabilities and generate a corresponding API token. (In the Beta API, please substitute "on" and "off" string values.)
- `resize_enabled` (*boolean*): Specify whether to scale down images to the maximum screen resolution, as determined by the rendering device's user agent. Note that enabling this may affect screen layout in unexpected ways. (In the Beta API, please substitute "on" and "off" string values.)
- `best_file_type_enabled` (*boolean*): Specify whether to convert images to the best file type for the requesting device, based on its user agent and the initial image file. This produces the smallest file size possible that retains image quality. (In the Beta API, please substitute "on" and "off" string values.)
- `advanced_settings_enabled` (*boolean*): When enabled, allows you to generate a custom [Image and Video Manager API](#) token to apply a corresponding policy to this set of images. The token consists of a descriptive label (the `api_key`) concatenated with a property-specific identifier that's generated when you save the property. The API registers the token when you activate the property. (In the Beta API, please substitute "on" and "off" string values.)
- `api_key` (*string*): With the `advanced_settings_enabled` option activated, assign a prefix label to help match the policy token to this set of images, limited to 32 alphanumeric or underscore characters. If you don't specify a label, *default* becomes the prefix.
- `api_key_default` (*string*): Specify the default policy identifier, which is registered with the [Image and Video Manager API](#) once you activate this property. (The `advanced_settings_enabled` option must be inactive.)

Feature previously named: `imagemanagement`

inputValidation

The Input Validation Cloudlet detects anomalous edge requests and helps mitigate repeated invalid requests. You can configure it using either the Cloudlets Policy Manager application, available within Control Center in **Configure⇒Cloudlets**, or the [Cloudlets API](#).

Use this behavior to specify criteria that identifies each unique end user, and optionally supplement the Input Validation policy with additional criteria your origin uses to identify invalid requests. Specify the threshold number of invalid requests that triggers a penalty, and the subsequent response. Also specify an ordinary failure response for those who have not yet met the threshold, which should not conflict with any other behavior that defines a failure response.

Options

- `enabled` (*boolean*): Applies the Input Validation Cloudlet behavior. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `cloudletPolicy` (*object*): Identifies the Cloudlet policy in an object featuring unique numeric `id` and descriptive string `name` members.
- `label` (*string*): Distinguishes this Input Validation policy from any others within the same property.
- `userIdentificationKeyType` (*enum string*): Sets the method to identify unique end users, either sets of query `PARAMS`, sets of HTTP `HEADERS`, or `BOTH`. Series of requests that share the same set of header or parameter values identify each unique user. Blank values must remain consistently blank across the series of requests for the user to be identified.
- `userIdentificationKeyHeaders` (*array of string values*): With `userIdentificationKeyType` set to `HEADERS` or `BOTH`, this specifies the HTTP headers whose combined set of values identify each end user.
- `userIdentificationKeyParams` (*array of string values*): With `userIdentificationKeyType` set to `PARAMS` or `BOTH`, this specifies the query parameters whose combined set of values identify each end user.
- `resetOnValid` (*boolean*): Upon receiving a valid request, enabling this resets the `penaltyThreshold` counter to zero. Otherwise, even those series of invalid requests that are interrupted by valid requests may trigger the `penaltyAction`. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `validateOnOriginWith` (*enum string*): For any validation that edge servers can't perform alone, this specifies additional validation steps based on how the origin identifies an invalid request. If a request is invalid, the origin can indicate this to the edge server

either with a `RESPONSE_CODE` or `RESPONSE_CODE_AND_HEADER` . If no additional validation is necessary, specify `DISABLED` .

- `validateOnOriginHeaderName` (*string*): If `validateOnOriginWith` is set to `RESPONSE_CODE_AND_HEADER` , this specifies the header name for a request that the origin identifies as invalid.
- `validateOnOriginHeaderValue` (*string*): If `validateOnOriginWith` is set to `RESPONSE_CODE_AND_HEADER` , this specifies an invalid request's header value that corresponds to the `validateOnOriginHeaderName` .
- `validateOnOriginResponseCode` (*number*): Unless `validateOnOriginWith` is `DISABLED` , this identifies the integer response code for requests the origin identifies as invalid.
- `failureAction` (*enum string*): For invalid requests that do not yet exceed the `penaltyThreshold` , this specifies the response, either `DENY_403` or `REDIRECT_302` .
- `failure302Uri` (*string*): With `failureAction` set to `REDIRECT_302` , this specifies the redirect link for invalid requests that have not yet triggered a penalty.
- `failureNetStorage` (*object*): With `failureAction` set to `DENY_403` , this specifies the NetStorage account that serves out the failure's static 403 response content. Details appear in an object featuring a `downloadDomainName` string member that identifies the NetStorage hostname, and an integer `cpCode` to track the traffic.
- `failure403NetStoragePath` (*string*): With `failureAction` set to `DENY_403` , this specifies the full path to the static 403 response content relative to the `downloadDomainName` in the `failureNetStorage` object.
- `penaltyThreshold` (*number*): Specifies the number of invalid requests permitted before executing the `penaltyAction` .
- `penaltyAction` (*enum string*): Once the `penaltyThreshold` of invalid requests is met, this specifies the response, either `DENY_403` or `REDIRECT_302` .
- `penalty302Uri` (*string*): With `penaltyAction` set to `REDIRECT_302` , this specifies the redirect link for end users who trigger the penalty.
- `penaltyNetStorage` (*object*): With `penaltyAction` set to `DENY_403` , this specifies the NetStorage account that serves out the penalty's static 403 response content. Details appear in an object featuring a `downloadDomainName` string member that identifies the NetStorage hostname, and an integer `cpCode` to track the traffic.
- `penalty403NetStoragePath` (*string*): With `penaltyAction` set to `DENY_403` , this specifies the full path to the static 403 response content relative to the `downloadDomainName` in the `penaltyNetStorage` object.

instant

The Instant feature allows you to prefetch content to the edge cache by adding link relation attributes to markup. For example:

```
<a href="page2.html" rel="Akamai-prefetch">Page 2</a>
```

Default link relation values are `prefetch` and `Akamai-prefetch`. Applies only to HTML elements that may specify an external file: `<a>`, `<base>`, ``, `<script>`, `<input>`, `<link>`, `<table>`, `<td>`, or `<th>`. (For the latter three, some legacy browsers support a nonstandard `background` image attribute.)

NOTE: This behavior provides an alternative to the `prefetch` and `prefetchable` behaviors, which allow you to configure more general prefetching behavior outside of markup.

Options

- `prefetch_cacheable` (*boolean*): When enabled, applies prefetching only to objects already set to be cacheable, for example using the `caching` behavior. Only applies to content with the `tieredDistribution` behavior enabled. (In the Beta API, please substitute "on" and "off" string values.)
- `prefetch_no_store` (*boolean*): Allows otherwise non-cacheable `no-store` content to prefetch if the URL path ends with `/` to indicate a request for a default file, or if the extension matches the value of the `prefetch_no_store_ext` option. Only applies to content with the `sureRoute` behavior enabled. (In the Beta API, please substitute "on" and "off" string values.)
- `prefetch_no_store_ext` (*array of string values*): Specifies a set of file extensions for which the `prefetch_no_store` option is allowed.
- `prefetch_html` (*boolean*): When enabled, allows edge servers to prefetch additional HTML pages while pages that link to them are being delivered. This only applies to links from `<a>` or `<link>` tags with the appropriate link relation attribute. (In the Beta API, please substitute "on" and "off" string values.)

- `customize_rel_attr` (*array of string values*): Specify link relation values that activate the prefetching behavior. For example, specifying `fetch` allows you to use shorter `rel="fetch"` markup.

Related behaviors: [enhancedAkamaiProtocol](#) , [dnsAsyncRefresh](#) , [redirect](#) , [cacheError](#) , [webApplicationFirewall](#) , [cacheRedirect](#)

InstantConfig

Multi-Domain Configuration, also known as *InstantConfig*, allows you to apply property settings to all incoming hostnames based on a DNS lookup, without explicitly listing them among the property's hostnames.

Options

- `status` (*boolean*): Enables the InstantConfig behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `mdc`

Related behaviors: [allowPost](#) , [allHttpInCacheHierarchy](#) , [allowPut](#) , [webApplicationFirewall](#) , [edgeOriginAuthorization](#) , [denyAccess](#)

LargeFileOptimization

The [Large File Optimization](#) feature improves performance and reliability when delivering large files. This behavior is required for objects larger than 1.8GB, and recommended for anything over 100MB. You should apply it only to the specific content that needs optimization, such as a download directory's `.gz` files, and enable the `filenames_changed` option while enforcing your own filename versioning policy.

NOTE: It is best to use [NetStorage](#) for objects larger than 1.8GB.

Options

- `status` (*boolean*): Enables the file optimization behavior. (In the Beta API, please substitute "on" and "off" string values.)
- `lfo_type` (*enum string*): Caches entire objects if set to `nonpoc`. Otherwise when set to `poc`, allows *partial-object caching*, which always applies to large objects served from [NetStorage](#). To enable this, the origin must support byte range requests.
- `min_size` (*string*): Optimization only applies to files larger than this, expressed as a number suffixed with a unit string such as `MB` or `GB`.
- `max_size` (*string*): Optimization does not apply to files larger than this, expressed as a number suffixed with a unit string such as `MB` or `GB`.
- `filenames_changed` (*boolean*): When `lfo_type` is set to `poc`, enabling this option signals your intention to vary filenames by version, strongly recommended to avoid serving corrupt content when chunks come from different versions of the same file. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `largefileoptimizations`

Related behaviors: [cacheKeyQueryParams](#), [cacheError](#), [cacheRedirect](#), [dnsAsyncRefresh](#), [removeVary](#), [cacheKeyIgnoreCase](#)

limitBitRate

Control the rate at which content is served to end users, optionally varying the speed depending on the file size or elapsed download time. Each bit rate specified in the `bitrateTable` array corresponds to a `thresholdTable` entry that activates it. You can use this behavior to prevent media downloads from progressing faster than they are viewed, for example, or to differentiate various tiers of end-user experience.

Options

- `option` (*boolean*): When enabled, activates the bit rate limiting behavior. (In the Beta API, please substitute "on" and "off" string values.)
- `bitrateTable` (*array*): Specifies a download rate that corresponds to a `thresholdTable` entry. The bit rate appears as a two-member object consisting of a numeric

`bitrateValue` and a `bitrateUnit` string, with allowed values of `Kbps` , `Mbps` , and `Gbps` . For example:

```
"bitrateTable": [
  {
    "bitrateValue": 1,
    "bitrateUnit": "Kbps"
  }
]
```

- `thresholdTable` (*array*): Specifies the minimum size of the file or the amount of elapsed download time before applying the bit rate limit from the corresponding `bitrateTable` entry. The threshold appears as a two-member object consisting of a numeric `thresholdValue` and `thresholdUnit` string, with allowed values of `s` or `B` . This example throttles a download that lasts more than 5 seconds:

```
"thresholdTable": [
  {
    "thresholdValue": 5,
    "thresholdUnit": "s"
  }
]
```

Feature previously named: `bitratelimiting`

Related behaviors: `cacheError` , `downstreamCache` , `cacheKeyIgnoreCase` , `modifyOutgoingRequestHeader` , `allowPost` , `rewriteUrl`

mediaFileRetrievalOptimization

Media File Retrieval Optimization (MFRO) speeds the delivery of large media files by relying on caches of partial objects. It is recommended for files larger than 100 MB, is required for files larger than 1.8 GB, and works best with [NetStorage](#).

Options

- `status` (*boolean*): Enables the partial-object caching behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)

modifyIncomingRequestHeader

Modify, add, remove, or pass along specific request headers coming upstream from the client.

Depending on the type of `action` you want to perform, specify the corresponding *standard* header name, or a `custom_header_name` if the standard name is set to `other`. The `header_value` serves as a match condition when the action is `delete` or `modify`, and the `new_header_value` applies when the action is `add` or `modify`.

See also [modifyIncomingResponseHeader](#), [modifyOutgoingRequestHeader](#), and [modifyOutgoingResponseHeader](#).

Options

- `action` (*enum string*): Either `add`, `delete`, `modify`, or `pass` incoming HTTP request headers.
- `standard_add_header_name` (*enum string*): If the value of `action` is `add`, this specifies the name of the field to add, either `accept-encoding`, `accept-language`, or `other`.
- `standard_delete_header_name` (*enum string*): If the value of `action` is `delete`, this specifies the name of the field to remove, either `if-modified-since`, `via`, or `other`.
- `standard_modify_header_name` (*enum string*): If the value of `action` is `modify`, this specifies the name of the field to modify, either `accept-encoding`, `accept-language`, or `other`.
- `standard_pass_header_name` (*enum string*): If the value of `action` is `pass`, this specifies the name of the field to pass through, either `accept-encoding`, `accept-language`, or `other`.
- `custom_header_name` (*string*): Specifies a custom field name that applies when the relevant *standard* header name is set to `other`.
- `header_value` (*string*): With the `action` set to `add`, specifies the new header value.

- `new_header_value` (*string*): With the `action` set to `modify` , supplies an HTTP header replacement value.
- `multi_headers_avoidance` (*boolean*): When enabled with the `action` set to `modify` , prevents creation of more than one instance of a header. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `modincomingreqheader`

Related behaviors: [modifyOutgoingResponseHeader](#) , [modifyOutgoingRequestHeader](#) , [cacheKeyQueryParams](#) , [redirect](#) , [denyAccess](#) , [cacheError](#)

modifyIncomingResponseHeader

Modify, add, remove, or pass along specific response headers coming downstream from the origin.

Depending on the type of `action` you want to perform, specify the corresponding *standard* header name, or a `custom_header_name` if the standard name is set to `other` . The `header_value` serves as a match condition when the action is `delete` or `modify` , and the `new_header_value` applies when the action is `add` or `modify` .

See also [modifyIncomingRequestHeader](#) , [modifyOutgoingRequestHeader](#) , and [modifyOutgoingResponseHeader](#) .

Options

- `action` (*enum string*): Either `add` , `delete` , `modify` , or `pass` incoming HTTP response headers.
- `standard_add_header_name` (*enum string*): If the value of `action` is `add` , this specifies the name of the field to add, any of the following values:

<code>cache-control</code>	<code>edge-control</code>	<code>last-modified</code>
<code>content-type</code>	<code>expires</code>	<code>other</code>

- `standard_delete_header_name` (*enum string*): If the value of `action` is `delete` , this specifies the name of the field to remove, either `cache-control` , `content-type` , `vary` , or `other` .

- `standard_modify_header_name` (*enum string*): If the value of `action` is `modify`, this specifies the name of the field to modify, either `cache-control`, `content-type`, `edge-control`, or `other`.
- `standard_pass_header_name` (*enum string*): If the value of `action` is `pass`, this specifies the name of the field to pass through, either `cache-control`, `expires`, `pragma`, or `other`.
- `custom_header_name` (*string*): Specifies a custom field name that applies when the relevant *standard* header name is set to `other`.
- `header_value` (*string*): With the `action` set to `add`, specifies the header's new value.
- `new_header_value` (*string*): With the `action` set to `modify`, specifies an HTTP header replacement value.
- `multi_headers_avoidance` (*boolean*): When enabled with the `action` set to `modify`, prevents creation of more than one instance of a header. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `modincomingrespheader`

Related behaviors: [cacheKeyQueryParams](#), [modifyOutgoingResponseHeader](#), [cacheError](#), [redirect](#), [removeVary](#), [denyAccess](#)

modifyOutgoingRequestHeader

Modify, add, remove, or pass along specific request headers going upstream towards the origin.

Depending on the type of `action` you want to perform, specify the corresponding *standard* header name, or a `custom_header_name` if the standard name is set to `other`. The `header_value` serves as a match condition when the action is `delete` or `modify`, and the `new_header_value` applies when the action is `add` or `modify`. Whole-text replacements apply when the action is `modify`, and substitutions apply when set to `regex`.

See also [modifyIncomingRequestHeader](#), [modifyIncomingResponseHeader](#), and [modifyOutgoingResponseHeader](#).

Options

- `action` (*enum string*): Either `add` or `delete` outgoing HTTP request headers, `modify` their fixed values, or specify a `regex` pattern to transform them.
- `standard_add_header_name` (*enum string*): If the value of `action` is `add`, this specifies the name of the field to add, either `user-agent` or `other`.
- `standard_delete_header_name` (*enum string*): If the value of `action` is `delete`, this specifies the name of the field to remove, either `pragma`, `user-agent`, `via`, or `other`.
- `standard_modify_header_name` (*enum string*): If the value of `action` is `modify` or `regex`, this specifies the name of the field to modify, either `user-agent` or `other`.
- `custom_header_name` (*string*): Specifies a custom field name that applies when the relevant *standard* header name is set to `other`.
- `header_value` (*string*): With the `action` set to `add`, specifies the new header value.
- `new_header_value` (*string*): With the `action` set to `modify`, specifies an HTTP header replacement value.
- `regex_header_match` (*string*): When the `action` is `regex`, specifies a Perl-compatible regular expression to match within the header value.
- `regex_header_replace` (*string*): When the `action` is `regex`, specifies text that replaces the `regex_header_match` pattern within the header value.
- `match_multiple` (*boolean*): When enabled with the `action` set to `regex`, replaces all occurrences of the matched regular expression, otherwise only the first match if disabled. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `multi_headers_avoidance` (*boolean*): When enabled with the `action` set to `modify`, prevents creation of more than one instance of a header. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `modoutgoingreqheader`

Related behaviors: [modifyOutgoingResponseHeader](#), [denyAccess](#), [redirect](#), [cacheError](#), [removeVary](#), [rewriteUrl](#)

modifyOutgoingResponseHeader

Modify, add, remove, or pass along specific response headers going downstream towards the client.

Depending on the type of `action` you want to perform, specify the corresponding *standard* header name, or a `custom_header_name` if the standard name is set to `other` . The `header_value` serves as a match condition when the action is `delete` or `modify` , and the `new_header_value` applies when the action is `add` or `modify` . Whole-text replacements apply when the action is `modify` , and substitutions apply when set to `regex` .

See also [modifyIncomingRequestHeader](#) , [modifyIncomingResponseHeader](#) , and [modifyOutgoingRequestHeader](#)

Options

- `action` (*enum string*): Either `add` or `delete` outgoing HTTP response headers, `modify` their fixed values, or specify a `regex` pattern to transform them.
- `standard_add_header_name` (*enum string*): If the value of `action` is `add` , this specifies the name of the field to add, any of the following values:

<code>access-control-allow-credentials</code>	<code>content-disposition</code>
<code>access-control-allow-headers</code>	<code>content-type</code>
<code>access-control-allow-methods</code>	<code>edge-control</code>
<code>access-control-allow-origin</code>	<code>pragma</code>
<code>access-control-expose-headers</code>	<code>P3P</code>
<code>access-control-max-age</code>	<code>other</code>
<code>cache-control</code>	

- `standard_delete_header_name` (*enum string*): If the value of `action` is `delete` , this specifies the name of the field to remove, any of the following values:

<code>access-control-allow-credentials</code>	<code>content-disposition</code>
<code>access-control-allow-headers</code>	<code>content-type</code>
<code>access-control-allow-methods</code>	<code>expires</code>
<code>access-control-allow-origin</code>	<code>pragma</code>
<code>access-control-expose-headers</code>	<code>P3P</code>
<code>access-control-max-age</code>	<code>other</code>
<code>cache-control</code>	

- `standard_modify_header_name` (*enum string*): If the value of `action` is `modify` or `regex` , this specifies the name of the field to modify, any of the following values:

<code>access-control-allow-credentials</code>	<code>access-control-expose-headers</code>
<code>access-control-allow-headers</code>	<code>access-control-max-age</code>
<code>access-control-allow-methods</code>	<code>cache-control</code>
<code>access-control-allow-origin</code>	<code>content-disposition</code>

content-type	P3P
pragma	other

- `custom_header_name` (*string*): Specifies a custom field name that applies when the relevant *standard* header name is set to `other`.
- `header_value` (*string*): Specifies the existing value of the header to match.
- `new_header_value` (*string*): With the `action` set to `modify`, specifies the new HTTP header replacement value.
- `regex_header_match` (*string*): When the `action` is `regex`, specifies a Perl-compatible regular expression to match within the header value.
- `regex_header_replace` (*string*): When the `action` is `regex`, specifies text that replaces the `regex_header_match` pattern within the header value.
- `match_multiple` (*boolean*): When enabled with the `action` set to `regex`, replaces all occurrences of the matched regular expression, otherwise only the first match if disabled. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `multi_headers_avoidance` (*boolean*): When enabled with the `action` set to `modify`, prevents creation of more than one instance of a header. The last header clobbers others with the same name. This option affects the entire set of outgoing headers, and is not confined to the subset of regular expression matches. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `modoutgoingrespheader`

Related behaviors: [cacheError](#), [redirect](#), [cacheKeyQueryParams](#), [removeVary](#), [denyAccess](#), [modifyOutgoingRequestHeader](#)

netSession

This behavior enables various features of NetSession, a client-side download manager application that's especially appropriate for large file downloads. For the feature to work, the end user must download the DLM client.

Options

- `netsession` (*boolean*): Enables NetSession DLM capabilities for this content. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `enable_domain` (*boolean*): ... (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `enable_dlm` (*boolean*): When enabled, launches files once they fully download. For example, specify this option to run an executable application. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `enable_download_clients` (*boolean*): When enabled, allows download clients to form a peer-to-peer network to reduce transmission time. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `disable_reporting` (*boolean*): Disable download state reporting via HTTP beacon messages. Otherwise when enabled, you can view the state of each download by choosing **Monitor⇒Download Analytics** on the DLM client. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `resume_url` (*array of string values*): Specify an alternate domain from which to resume a paused download. This generates a corresponding shortcut link on the end user's desktop that disappears after the download is complete.
- `org_name` (*string*): The name of the organization that displays in the NetSession client DLM interface.
- `support_url` (*string*): A supporting link to the `org_name` that displays in the NetSession client DLM interface.

Feature previously named: `netsession`

networkConditionsHeader

Instructs edge servers to send an `X-Akamai-Network-Condition` header to the origin assessing the quality of the network.

Options

- `behavior` (*enum string*): If set to `2tier`, assessment is either `Excellent` or `Poor`. If set to `3tier`, the assessment can also be `Fair`.

Feature previously named: `networkconditionsheader`

Related behaviors: `tcpOptimization` , `adaptiveImageCompression` , `deviceCharacteristicCached` , `shutr`

origin

Specify the hostname and settings used to contact the origin once service begins. You can use your own origin, [NetStorage](#), an Edge Load Balancing origin, or a SaaS dynamic origin.

Options

- `type` (*enum string*): Choose whether to fetch your content from your own server (`customer`), your [NetStorage](#) account (`netstorage`), any available Edge Load Balancing origin (`elb_origin_group`), or a SaaS dynamic origin (`saas_dynamic_origin`). NetStorage is most appropriate for static content.
- `netstorage` (*object*): If the `type` is `netstorage` , this option specifies the details of the netstorage server. For example:

```
"netstorage": {
  "id" : "id_string",
  "name" : "Example Downloads",
  "downloadDomainName" : "example.download.akamai.com",
  "cpCode" : 12345
}
```

- `origin_id` (*string*): With the `origin type` set to `elb_origin_group` , identifies the Edge Load Balancing origin. This must correspond to an [edgeLoadBalancingOrigin](#) behavior's `id` attribute within the same property.
- `hostname` (*string*): With the `origin type` set to `customer` , this specifies the hostname or IPv4 address of your origin server, from which edge servers can retrieve your content.
- `saas_type` (*enum string*): With `type` set to `saas_dynamic_origin` , specifies the part of the request that identifies this SaaS dynamic origin, either `path` , `cookie` , `query_string` , or `hostname` .

- `saas_cname_enabled` (*boolean*): With `saas_type` set to `hostname` , enabling this allows you to use a *CNAME chain* to determine the hostname for this SaaS dynamic origin. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `saas_cname_level` (*number*): With `saas_type` set to `hostname` and `saas_cname_enabled` on, specifies the desired number of hostnames to use in the *CNAME chain*, starting backwards from the edge server.
- `saas_cookie` (*string*): With the `type` set to `saas_dynamic_origin` and the `saas_type` set to `cookie` , this specifies the name of the cookie that identifies this SaaS dynamic origin.
- `saas_query_string` (*string*): With the `type` set to `saas_dynamic_origin` and the `saas_type` set to `query_string` , this specifies the name of the query parameter that identifies this SaaS dynamic origin.
- `saas_regex` (*string*): With the `type` set to `saas_dynamic_origin` , this specifies the Perl-compatible regular expression match that identifies this SaaS dynamic origin.
- `saas_replace` (*string*): Specifies replacement text for what `saas_regex` matches.
- `saas_suffix` (*string*): With the `type` set to `saas_dynamic_origin` , specifies the static part of the SaaS dynamic origin.
- `forwardhostheader` (*enum string*): When the `type` is set to either `customer` or `saas_dynamic_origin` , this specifies which `Host` header to pass to the origin:
 - `requesthostheader` passes the original request's header.
 - `originhostname` passes the current origin's `hostname` .
 - `custom` passes the value of `customforwardhostheader` . Use this option if you want requests handled by different properties to converge on the same cached object.
- `customforwardhostheader` (*string*): With `forwardhostheader` set to `custom` , this specifies the name of the custom host header the edge server should pass to the origin.
- `cachekeyhostname` (*enum string*): With the origin `type` set to `custom` , this specifies the hostname to use when forming a cache key. Specify `originhostname` if your origin server's responses do not depend on the hostname, otherwise specify `requesthostheader` when using a virtual server.
- `compression` (*boolean*): Enables *gzip* compression for non-NetStorage origins. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `tcip_enabled` (*boolean*): When enabled on non-NetStorage origins, allows you to send a custom header (the `tcip_header`) identifying the IP address of the immediate client

connecting to the edge server. This may provide more useful information than the standard `X-Forward-For` header, which proxies may modify. (In the Beta API, please substitute `"on"` and `"off"` string values.)

- `tcip_header` (*string*): With `tcip_enabled` on, this specifies the name of the field identifying the end client's IP address, for example `True-Client-IP`.
- `tcip_allow_clients_to_set` (*boolean*): With `tcip_enabled` on along with this option, if a client sets the `True-Client-IP` header, the edge server allows it and passes the value to the origin. Otherwise the edge server removes it and sets the value itself. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `origin_cert_delegate` (*enum string*): For non-NetStorage origins, maximize security by controlling which certificates edge servers should trust, either `delegate`, `warn`, `custom`, or `no_selection`. (This option only applies if the property is marked as secure. See [Secure Property Requirements](#) for guidance.)
- `origin_cert_valid_cn_other` (*array of string values*): With `origin_cert_delegate` set to `custom`, specifies values to look for in the origin certificate's `Subject Alternate Name` or `Common Name` fields. Specify `{{Origin Hostname}}` and `{{Forward Host Header}}` as variables in the order you want them to be evaluated.
- `origin_certs_to_honor` (*enum string*): With `origin_cert_delegate` set to `custom`, specifies whether to trust pinned origin server certificates (`custom_certs`), any certificate signed by an Akamai-managed authority set (`standard_cas`), or any certificate signed by a custom authority set you manage (`custom_cas`). If set to `combo`, may rely on all three inputs.
- `origin_cert_standard_cas` (*object*): With `origin_certs_to_honor` set to either `standard_cas` or `combo`, specifies an array of Akamai-managed certificate names. Currently, the only allowed value is `akamai-permissive`.
- `origin_cert_custom_cas` (*object*): With `origin_certs_to_honor` set to either `custom_cas` or `combo`, specifies an array of certification objects. Contact your Akamai representative for details on this object's requirements.
- `origin_cert_custom_certs` (*object*): With `origin_certs_to_honor` set to either `custom_certs` or `combo`, specifies an array of certification objects. Contact your Akamai representative for details on this object's requirements.
- `http_port` (*string*): Specifies the port on your origin server to which edge servers should connect for HTTP requests, customarily `80`.
- `https_port` (*string*): Specifies the port on your origin server to which edge servers should connect for secure HTTPS requests, customarily `443`. (This option only

applies if the property is marked as secure. See [Secure Property Requirements](#) for guidance.)

persistentClientConnection

This [read-only behavior](#) activates *persistent connections* between edge servers and clients, which allow for better performance and more efficient use of resources. Compare with the [persistentConnection](#) behavior, which configures persistent connections for the entire journey from origin to edge to client. Contact Akamai Professional Services for help configuring either.

WARNING: Disabling or removing this behavior may negatively affect performance.

Options

- `status` (*boolean*): Enables the persistent connections behavior. (In the Beta API, please substitute "on" and "off" string values.)
- `timeout` (*duration string*): Specifies the timeout period after which edge server closes the persistent connection with the client, 500 seconds by default.

Feature previously named: `clientpconns`

Related behaviors: [removeVary](#) , [cacheKeyQueryParams](#) , [dnsAsyncRefresh](#) , [denyAccess](#) , [redirect](#) , [prefreshCache](#)

persistentConnection

A [read-only behavior](#) that enables more efficient *persistent connections* from origin to edge server to client. Compare with the [persistentClientConnection](#) behavior, which customizes persistent connections from edge to client. Contact Akamai Professional Services for help configuring either.

WARNING: Disabling this behavior wastes valuable browser resources. Leaving connections open too long makes them vulnerable to attack. Avoid both of these scenarios.

Options

- `status` (*boolean*): Enables persistent connections. (In the Beta API, please substitute "on" and "off" string values.)
- `timeout` (*duration string*): Specifies the timeout period after which edge server closes a persistent connection.

Feature previously named: `pconns`

Related behaviors: [cacheKeyIgnoreCase](#) , [webApplicationFirewall](#) , [cacheKeyQueryParams](#) , [redirect](#) , [cacheError](#) , [dnsAsyncRefresh](#)

personallyIdentifiableInformation

Marks content covered by the current rule as sensitive *personally identifiable information* that needs to be treated as secure and private. That includes anything involving personal information: name, social security number, date and place of birth, mother's maiden name, biometric data, or any other data linked to an individual. If you attempt to save a property with such a rule that also caches or logs sensitive content, the added behavior results in a validation error.

WARNING: This feature only identifies some vulnerabilities. For example, it does not prevent you from including secure information in a query string or writing it to an origin folder. It also can't tell whether the SSL protocol is in effect.

Options

- `treat_as_pii` (*boolean*): When enabled, marks content as personally identifiable information (PII). (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `pii`

Related behaviors: [cacheKeyQueryParams](#) , [webApplicationFirewall](#) , [redirect](#) , [cacheKeyIgnoreCase](#) , [cacheError](#) , [allowDelete](#)

predictivePrefetching

This behavior potentially reduces the client's page load time by pre-caching objects based on historical data for the page, not just its current set of referenced objects. It also detects second-level dependencies, such as objects retrieved by JavaScript.

Options

- `status` (*boolean*): Enables the predictive prefetching behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `accuracy-target` (*enum string*): The level of prefetching, either `low`, `med`, or `high`. A higher level results in better client performance, but potentially greater load on the origin.

Feature previously named: `predictiveprefetching`

prefetch

Instructs edge servers to retrieve content linked from requested pages as they load, rather than waiting for separate requests for the linked content. This behavior applies depending on the rule's set of matching conditions. Use in conjunction with the `prefetchable` behavior, which specifies the set of objects to prefetch.

Options

- `enabled` (*boolean*): Applies prefetching behavior when enabled. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `prefetching`

prefetchable

Allow matching objects to prefetch into the edge cache as the parent page that links to them loads, rather than waiting for a direct request. This behavior applies depending on the rule's set of matching conditions. Use `prefetch` to enable the overall behavior for parent pages that contain links to the object.

Options

- `enabled` (*boolean*): When enabled, allows matching content to prefetch when referenced on a requested parent page. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `prefetchableobject`

Related behaviors: `gzipResponse` , `prefreshCache` , `report` , `adaptiveImageCompression` , `dnsAsyncRefresh` , `rewriteUrl`

prefreshCache

Refresh cached content before its time-to-live (TTL) expires, to keep end users from having to wait for the origin to provide fresh content.

Prefreshing starts asynchronously based on a percentage of remaining TTL. The edge serves the prefreshed content only after the TTL expires. If the percentage is set too high, and there is not enough time to retrieve the object, the end user waits for it to refresh from the origin, as is true by default without this prefresh behavior enabled. The edge does not serve stale content.

Options

- `enabled` (*boolean*): Enables the cache prefreshing behavior. (In the Beta API, please substitute `"true"` and `"false"` string values.)

- `prefreshval` (*number within 0-100 range*): Specifies when the prefetch occurs as a percentage of the TTL. For example, for an object whose cache has 10 minutes left to live, and an origin response that is routinely less than 30 seconds, a percentage of 95 prefetches the content without unnecessarily increasing load on the origin.

Feature previously named: `cacheprefresh`

Related behaviors: [removeVary](#) , [cacheKeyIgnoreCase](#) , [cacheError](#) , [cacheKeyQueryParams](#) , [tieredDistribution](#) , [cacheRedirect](#)

randomSeek

Optimizes `.flv` and `.mp4` files to allow random jump-point navigation.

Options

- `flv` (*boolean*): Enables random seek optimization in FLV files. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `mp4` (*boolean*): Enables random seek optimization in MP4 files. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `max_size` (*string*): With the `mp4` option enabled, sets the maximum size of the MP4 file to optimize, expressed as a number suffixed with a unit string such as `MB` or `GB` .

Feature previously named: `randomseek`

readTimeout

A [read-only behavior](#) that specifies how long the edge server should wait for a response from the requesting forward server after a connection has already been established. Any

failure to read aborts the request and sends a `504` Gateway Timeout error to the client. Contact Akamai Professional Services for help configuring this behavior.

Options


- `timeout` (*duration string*): Specifies the read timeout necessary before failing with a `504` error. This value should never be zero.


Feature previously named: `readtimeout`

Related behaviors: `removeVary` , `cacheError` , `redirect` , `cacheKeyQueryParams` , `dnsAsyncRefresh` , `webApplicationFirewall`

realUserMonitoring

[Real User Monitoring](#)

[\(RUM\)](#)  injects JavaScript into HTML pages served to end-user clients that monitors page-load performance and reports on various data, such as browser type and geographic location. See [report](#) for information on how to configure logs.

Akamai customers can consult the documentation for [Real User Monitoring](#)  for more information on this behavior.

Options

- `status` (*boolean*): When enabled, activates real-use monitoring. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `rum`

redirect

Respond to the client request with a redirect without contacting the origin. Specify the redirect as a path expression starting with a `/` character relative to the current root, or as a fully qualified URL. This behavior relies primarily on `destination_host` and `destination_path` to manipulate the hostname and path independently.

Options

- `mobile_default_choice` (*enum string*): When set to `mobile`, allows only a `302` response code. When set to `default`, allows all other `responseCode` values.
- `destination_protocol` (*enum string*): Choose the protocol for the redirect URL, either `http`, `https`, or `same_as_request` to pass through the original protocol.
- `destination_host` (*enum string*): Specify how to change the requested hostname, independently from the pathname:
 - `subdomain` prepends a subdomain from the `destination_host_subdomain` field.
 - `sibling` replaces the leftmost subdomain with the `destination_host_sibling` field.
 - `other` specifies a static domain in the `destination_host_other` field.
 - `same_as_request` preserves the hostname unchanged.
- `destination_host_subdomain` (*string*): If `destination_host` is set to `subdomain`, this specifies a subdomain to prepend to the current hostname. For example, a value of `m` changes `www.example.com` to `m.www.example.com`.
- `destination_host_sibling` (*string*): If `destination_host` is set to `sibling`, this specifies the subdomain with which to replace to the current hostname's leftmost subdomain. For example, a value of `m` changes `www.example.com` to `m.example.com`.
- `destination_host_other` (*string*): If `destination_host` is set to `other`, this specifies the full hostname with which to replace the current hostname.
- `destination_path` (*enum string*): Specify how to change the requested pathname, independently from the hostname:
 - `prefix_request` prepends a path with the `destination_path_prefix` field. You also have the option to specify a suffix using `destination_path_suffix` and `destination_path_suffix_status`.
 - `other` replaces the current path with the `destination_path_other` field.
 - `same_as_request` preserves the current path unchanged.
- `destination_path_prefix` (*string*): When `destination_path` is set to `prefix_request`, this prepends the current path. For example, a value of `/prefix/path` changes

`/example/index.html` to `/prefix/path/example/index.html` .

- `destination_path_suffix_status` (*enum string*): When `destination_path` is set to `prefix_request` , this gives you the option of adding a suffix. Specify `no_suffix` if you want to preserve the end of the path unchanged. If you specify `suffix` , the `destination_path_suffix` provides the value.
- `destination_path_suffix` (*string*): When `destination_path` is set to `prefix_request` and `destination_path_suffix_status` is set to `suffix` , this specifies the suffix to append to the path.
- `destination_path_other` (*string*): When `destination_path` is set to `prefix_request` , this replaces the current path.
- `queryString` (*boolean*): When enabled, passes incoming query string parameters as part of the redirect URL. (In the Beta API, please substitute "ignore" and "append" string values.)
- `responseCode` (*enum string*): Specify the redirect's response code: `301` (Moved Permanently), `302` (Found), `303` (See Other), or `307` (Temporary Redirect).

Related behaviors: [cacheKeyQueryParams](#) , [cacheError](#) , [removeVary](#) , [webApplicationFirewall](#) , [denyAccess](#) , [modifyOutgoingResponseHeader](#)

referrerChecking

Limits allowed requests to a set of domains you specify.

Options

- `status` (*boolean*): Enables the referer-checking behavior. (In the Beta API, please substitute "on" and "off" string values.)
- `strict` (*boolean*): When enabled, excludes requests whose `Referer` header include a relative path, or that are missing a `Referer` . When disabled, only excludes requests whose `Referer` hostname is not part of the `domain` set. (In the Beta API, please substitute "on" and "off" string values.)
- `domain` (*array of string values*): Specifies the set of allowed domains. With `allowchildren` disabled, prefixing values with `*`. specifies domains for which

subdomains are allowed.

- `allowchildren` (*boolean*): When enabled, allows all subdomains for the `domain` set, just like adding a `*` prefix to each. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `refererchecking`

Related behaviors: `cacheError` , `dnsAsyncRefresh` , `removeVary` , `tieredDistribution` , `cacheRedirect` , `denyAccess`

removeQueryParameter

Remove named query parameters before forwarding the request to the origin.

Options

- `removelist` (*array of string values*): Specifies parameters to remove from the request.

Feature previously named: `removeqsbyname`

Related behaviors: `cacheKeyQueryParams` , `removeVary` , `cacheError` , `downstreamCache` , `redirect` , `modifyOutgoingResponseHeader`

removeVary

By default, responses that feature a `Vary` header value of anything other than `Accept-Encoding` and a corresponding `Content-Encoding: gzip` header aren't cached on edge servers. `Vary` headers indicate when a URL's content varies depending on some variable, such as which `User-Agent` requests it. This behavior simply removes the `Vary` header to make responses cacheable.

WARNING: If your site relies on `Vary: User-Agent` to customize content, removing the header may lead the edge to serve content inappropriate for specific devices.

Options

- `remove_vary` (*boolean*): When enabled, removes the `Vary` header to ensure objects can be cached. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `removevary`

Related behaviors: [cacheError](#) , [cacheKeyIgnoreCase](#) , [redirect](#) , [cacheKeyQueryParams](#) , [cacheRedirect](#) , [dnsAsyncRefresh](#)

report

Specify the HTTP request headers or cookie names to log in your Akamai Log Delivery service reports.

Options

- `host_enabled` (*boolean*): Log the `Host` header. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `referer_enabled` (*boolean*): Log the `Referer` header. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `useragent_enabled` (*boolean*): Log the `User-Agent` header. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `acceptlang_enabled` (*boolean*): Log the `Accept-Language` header. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `cookie_mode` (*enum string*): With a set of defined `cookie` names, specifies whether you want to log `all` or `some` cookies, or to turn `off` the feature to stop logging cookies.
- `cookies` (*array of string values*): With `cookie_mode` set to `some` , this specifies the set of cookies names whose values you want to log.

Feature previously named: `reporting`

Related behaviors: `gzipResponse` , `prefetchable` , `prefreshCache` , `dnsAsyncRefresh` , `rewriteUrl` , `redirect`

requestControl

The Request Control Cloudlet allows you to control access to your web content based on the incoming request's IP or geographic location. Assuming cloudlets are available on your contract, choose **Configure⇒Cloudlets** to control how the feature works, or use the [Cloudlets API](#) to configure it programmatically.

Options

- `status` (*boolean*): Enables the Request Control Cloudlet. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `nimbus_policy_token` (*string*): Specifies the name of the Request Control policy, using alphanumeric and underscore characters.
- `enable_branded_403` (*boolean*): If enabled, serves a branded 403 page for this Cloudlet instance. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `netstorage` (*object*): Specifies the NetStorage domain that contains the branded 403 page.
- `branded_403_file` (*string*): Specifies the full path of the branded 403 page, including the filename, but excluding the NetStorage CP code path component.

responseCode

Change the existing response code. For example, if your origin sends a `301` permanent redirect, this behavior can change it on the edge to a temporary `302` redirect.

Options

- `statuscode` (*enum string*): The HTTP status code to replace the existing one, any of the following: 100 , 101 , 102 , 103 , 122 , 200 , 201 , 202 , 203 , 204 , 205 , 206 , 207 , 226 , 300 , 301, 302 , 303 , 304 , 305 , 306 , 307 , 308 , 400 , 401 , 402 , 403 , 404 , 405 , 406, 407 , 408 , 409 , 410 , 411 , 412 , 413 , 414 , 415 , 416 , 417 , 422 , 423 , 424, 425 , 426 , 428 , 429 , 431 , 444 , 449 , 450 , 499 , 500 , 501 , 502 , 503 , 504, 505 , 506 , 507 , 509 , 510 , 511 , 598 , 599 .
- `enable206override` (*boolean*): When enabled, allows any specified 200 success code to override a 206 partial-content code, in which case the response's content length matches the requested range length. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `setresponsecode`

Related behaviors: [redirect](#) , [cacheError](#) , [modifyOutgoingResponseHeader](#) , [denyAccess](#) , [cacheKeyQueryParams](#) , [rewriteUrl](#)

responseCookie

Set a cookie to send downstream to the client, either with a fixed value or a unique stamp.

Options

- `status` (*boolean*): When enabled, allows you to set a response cookie. (In the Beta API, please substitute "on" and "off" string values.)
- `name` (*string*): Specifies the name of the cookie, which serves as a key to determine if the cookie is set.
- `type` (*enum string*): Assign either a `unique` value, or a `fixed` one based on the `value` field.
- `value` (*string*): If the cookie `type` is `fixed` , this specifies the cookie value.
- `format` (*enum string*): When the `type` of cookie is set to `unique` , set this to either `Apache` or `Akamai` format. The latter format adds milliseconds to the date stamp.

- `defaultDomain` (*enum string*): Either use the `default` domain, or a `specific` one from the `domain` fields.
- `defaultPath` (*enum string*): Either use the `default` path value, or a `specific` one from the `path` field.
- `domain` (*string*): If the `defaultdomainpath` is set to `specific`, this sets the domain for which the cookie is valid. For example, `example.com` makes the cookie valid for that hostname and all subdomains.
- `path` (*string*): If the `defaultdomainpath` is set to `specific`, sets the path component for which the cookie is valid.
- `expires` (*enum string*): Sets various ways to specify when the cookie expires:
 - A value of `never` lets the cookie persist indefinitely.
 - A value of `on_browser_close` limits the cookie to the duration of the session.
 - A value of `duration` requires a corresponding `duration` field value.
 - A value of `fixed_date` requires a corresponding `expiration_date` field value.
- `expiration_date` (*ISO 8601 format date/time string*): If `expires` is set to `fixed_date`, this sets when the cookie expires as a UTC date and time.
- `duration` (*duration string*): If `expires` is set to `duration`, this sets the cookie's lifetime.
- `secure` (*boolean*): When enabled, sets the cookie's `Secure` flag to transmit it with HTTPS. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `setresponsecookie`

Related behaviors: `redirect`, `cacheKeyQueryParams`, `modifyOutgoingResponseHeader`, `cacheError`, `removeVary`, `webApplicationFirewall`

restrictObjectCaching

A [read-only behavior](#) that is required to deploy the Object Caching product. It disables serving HTML content and limits the maximum object size to 100MB. Contact Akamai Professional Services for help configuring it.

Options

- `html_disabled` (*read-only string*): Disables edge caching for HTML content.
- `maximum_size` (*read-only string*): Specifies a fixed maximum size of non-HTML content to cache.

Feature previously named: `objectcachingrestrictions`

rewriteUrl

Modifies the path of incoming requests to forward to the origin. This helps you offload URL-rewriting tasks to the edge to increase the origin server's performance, allows you to redirect links to different targets without changing markup, and hides your original directory structure.

Except for regular expression replacements, this behavior manipulates *path expressions*, which start and end with a `/` character.

Options

- `behavior` (*enum string*): The action to perform on the path:
 - If set to `prepend`, specify the `targetpathprepend`. For example, if set to `/prefix/`, `/path1/page.html` changes to `/prefix/path1/page.html`.
 - If set to `replace`, specify the `trigger` and `targetpath`. For example, a `trigger` of `/path1/` and a `targetpath` of `/path1/path2/` changes `/path1/page.html` to `/path1/path2/page.html`.
 - If set to `regexreplace`, specify the `triggerregex` and `targetregex`. For example, specifying `logo\\.(png|gif|jpe?g)` and `brand\\$1` changes `logo.png` to `brand.png`.
 - If set to `rewrite`, specify the `targeturl`. For example, you can direct traffic to `/error/restricted.html`.
 - If set to `remove`, specify the `trigger`. For example, a `trigger` of `/path2/` changes `/path1/path2/page.html` to `/path1/page.html`.

- `trigger` (*string*): When `behavior` is `remove` or `replace` , specifies the part of the incoming path you'd like to remove or modify.
- `triggerregex` (*string*): When `behavior` is set to `regexreplace` , specifies the Perl-compatible regular expression to replace with `targetregex` .
- `targetregex` (*string*): When `behavior` is set to `regexreplace` , this replaces whatever the `triggerregex` field matches, along with any captured sequences from `\$1` through `\$9` .
- `targetpath` (*string*): When `behavior` is set to `replace` , this path replaces whatever the `trigger` field matches in the incoming request's path.
- `targetpathprepend` (*string*): When `behavior` is set to `prepend` , specifies a path to prepend to the incoming request's URL.
- `targeturl` (*string*): When `behavior` is set to `rewrite` , specifies the full path to request from the origin.
- `match_multiple` (*boolean*): When enabled, replaces all potential matches rather than only the first. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `keepqs` (*boolean*): When enabled, retains the original path's query parameters. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `urlrewrite`

Related behaviors: `cacheError` , `cacheKeyQueryParams` , `redirect` , `denyAccess` , `removeVary` , `modifyOutgoingResponseHeader`

rmaOptimization

This behavior is deprecated. Do not add it to any properties.

This behavior does not include any options. Specifying the behavior itself enables it.

Feature previously named: `rmaoptimizations`

Related behaviors: `cacheRedirect` , `cacheKeyIgnoreCase` , `cacheError` , `redirect` , `webApplicationFirewall` , `cacheKeyQueryParams`

saasdefinitions

Configures how the Software as a Service feature identifies *customers*, *applications*, and *users*. A different set of options is available for each type of targeted request, each enabled with the `action`-suffixed option. In each case, you can use `path`, `cookie`, `query_string`, or `hostname` components as identifiers, or `disable` the SaaS behavior for certain targets. If you rely on a `hostname`, you also have the option of specifying a *CNAME chain* rather than an individual hostname. The various options suffixed `regex` and `replace` subsequently remove the identifier from the request.

NOTE: This behavior needs a sibling `origin` behavior whose `type` option is set to `saas_dynamic_origin`.

Options

- `application_action` (*enum string*): Specifies the request component that identifies a SaaS application, either `cookie`, `hostname`, `path`, or `query_string`. Setting it to `disabled` effectively ignores applications.
- `application_cookie` (*string*): With `application_action` set to `cookie`, this specifies the name of the cookie that identifies the application.
- `application_query_string` (*string*): With `application_action` set to `query_string`, this names the query parameter that identifies the application.
- `application_cname_enabled` (*boolean*): With `application_action` set to `hostname`, enabling this allows you to identify applications using a *CNAME chain* rather than a single hostname. (In the Beta API, please substitute "on" and "off" string values.)
- `application_cname_level` (*number*): With `application_cname_enabled` on, this specifies the number of CNAMEs to use in the chain.
- `application_regex` (*string*): Specifies a Perl-compatible regular expression with which to substitute the request's application ID.
- `application_replace` (*string*): Specifies a string to replace the request's application ID matched by `application_regex`.
- `customer_action` (*enum string*): Specifies the request component that identifies a SaaS customer, either `cookie`, `hostname`, `path`, or `query_string`. Setting it to `disabled`

effectively ignores customers.

- `customer_cookie` (*string*): With `customer_action` set to `cookie` , this specifies the name of the cookie that identifies the customer.
- `customer_query_string` (*string*): With `customer_action` set to `query_string` , this names the query parameter that identifies the customer.
- `customer_cname_enabled` (*boolean*): With `customer_action` set to `hostname` , enabling this allows you to identify customers using a *CNAME chain* rather than a single hostname. (In the Beta API, please substitute "on" and "off" string values.)
- `customer_cname_level` (*number*): With `customer_cname_enabled` on, this specifies the number of CNAMEs to use in the chain.
- `customer_regex` (*string*): Specifies a Perl-compatible regular expression with which to substitute the request's customer ID.
- `customer_replace` (*string*): Specifies a string to replace the request's customer ID matched by `customer_regex` .
- `users_action` (*enum string*): Specifies the request component that identifies a SaaS user, either `cookie` , `hostname` , `path` , or `query_string` . Setting it to `disabled` effectively ignores users.
- `users_cookie` (*string*): With `users_action` set to `cookie` , this specifies the name of the cookie that identifies the user.
- `users_query_string` (*string*): With `users_action` set to `query_string` , this names the query parameter that identifies the user.
- `users_cname_enabled` (*boolean*): With `users_action` set to `hostname` , enabling this allows you to identify users using a *CNAME chain* rather than a single hostname. (In the Beta API, please substitute "on" and "off" string values.)
- `users_cname_level` (*number*): With `user_cname_enabled` on, this specifies the number of CNAMEs to use in the chain.
- `users_regex` (*string*): Specifies a Perl-compatible regular expression with which to substitute the request's user ID.
- `users_replace` (*string*): Specifies a string to replace the request's user ID matched by `users_regex` .

savePostDcaProcessing

A [read-only behavior](#), used in conjunction with the [cachePost](#) behavior, that allows the body of POST requests to be processed through Dynamic Content Assembly. Contact Akamai Professional Services for help configuring it.

Options

- `enabled` (*boolean*): Enables processing of POST requests. (In the Beta API, please substitute "on" and "off" string values.)

Feature previously named: `save_post_dca_processing`

Related behaviors: [cachePost](#) , [modifyOutgoingResponseHeader](#) , [modifyOutgoingRequestHeader](#) , [dnsAsyncRefresh](#) , [redirect](#) , [denyAccess](#)

scheduleInvalidation

Specifies when cached content that satisfies a rule's criteria expires, optionally at repeating intervals. In addition to periodic cache flushes, you can use this behavior to minimize potential conflicts when related objects expire at different times.

WARNING: scheduled invalidations can significantly increase origin servers' load when matching content expires simultaneously across all edge servers. As best practice, schedule expirations during periods of lowest traffic.

Options

- `start` (*ISO 8601 format date/time string*): The UTC date and time when matching cached content is to expire.
- `repetition_enabled` (*boolean*): When enabled, invalidation recurs periodically from the `start` time based on the `repeatinterval` time. (In the Beta API, please substitute "unchecked" and "checked" string values.)

- `repeatinterval` (*duration string*): With `repetition_enabled` on, specifies how often to invalidate content from the `start` time, expressed in seconds. For example, an expiration set to midnight and an interval of `86400` seconds invalidates content once a day.

NOTE: Repeating intervals of less than 5 minutes are not allowed for [NetStorage](#) origins.

- `refresh_method` (*enum string*): Specifies how to invalidate the content. Setting it to `invalidate` sends an `If-Modified-Since` request to the origin, re-caching the content only if it is fresher. Setting it to `purge` re-caches content regardless of its freshness, potentially creating more traffic at the origin.

Feature previously named: `scheduledinvalidation`

Related behaviors: [redirect](#) , [removeVary](#) , [modifyOutgoingResponseHeader](#) , [cacheKeyQueryParams](#) , [cacheError](#) , [dnsAsyncRefresh](#)

segmentedContentProtection

Validates authorization tokens at the edge server to prevent unauthorized link sharing.

Options

- `segmentedcontentprotections witch` (*boolean*): Enables the segmented content protection behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `key` (*string*): Specifies the encryption key to use as a shared secret to validate tokens.
- `show_advanced` (*boolean*): When enabled, allows you to specify advanced `transitionKey` and `salt` options. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `transitionKey` (*string*): An alternate encryption key to match along with the `key` field, allowing you to rotate keys with no down time.
- `salt` (*string*): Specifies a salt as input into the token for added security. This value must match the salt used in the token generation code.

- `failureResponse` (*enum string*): Either `verifyanddeny` or `verifyonly` .

Feature previously named: `segmentedcontentprotection`

segmentedMediaOptimization

Optimizes segmented media for live or streaming delivery contexts.

Options

- `behavior` (*enum string*): Set to either `cached` `on_demand` , or streaming `live` . (Only `on_demand` is allowed for [NetStorage](#) origins.)

Feature previously named: `segmentedmediaoptimization`

shutr

The SHUTR protocol extends HTTP to reduce the amount of header data necessary for web transactions with mobile devices.

This behavior does not include any options. Specifying the behavior itself enables it.

Related behaviors: [enhancedAkamaiProtocol](#) , [redirect](#) , [adaptiveImageCompression](#) , [cacheRedirect](#) , [cacheError](#) , [removeVary](#)

simulateErrorCode

A [read-only behavior](#) that simulates various error response codes. Contact Akamai Professional Services for help configuring it.

Options

- `error_type` (*enum string*): Specifies the type of error, any of the following:

<code>err_connect_fail</code>	<code>err_no_good_fwd_ip</code>
<code>err_connect_timeout</code>	<code>err_read_error</code>
<code>err_dns_fail</code>	<code>err_read_timeout</code>
<code>err_dns_in_region</code>	<code>err_surerroute_dns_fail</code>
<code>err_dns_timeout</code>	<code>err_write_error</code>
- `timeout` (*duration string*): When the `error_type` is `err_connect_timeout`, `err_dns_timeout`, `err_surerroute_dns_fail`, or `err_read_timeout`, generates an error after the specified amount of time from the initial request.

Feature previously named: `sim_error_codes`

Related behaviors: [failAction](#), [timeout](#), [responseCookie](#), [modifyIncomingRequestHeader](#), [cacheRedirect](#), [cacheError](#)

siteShield

A [read-only behavior](#) implementing the [SiteShield](#) feature, which helps prevent non-Akamai machines from contacting your origin. Your service representative periodically sends you a list of Akamai servers allowed to contact your origin, with which you establish an Access Control List on your firewall to prevent any other requests.

Options

- `ssmap` (*string*): The hostname that identifies the SiteShield map, available from your Akamai representative.

Feature previously named: `siteshield`

Related behaviors: [webApplicationFirewall](#), [redirect](#), [cacheError](#), [denyAccess](#), [cacheKeyQueryParams](#), [cacheKeyIgnoreCase](#)

spdy

The SPDY protocol enhances HTTPS traffic by using many concurrent connections to download objects within one TCP connection.

This behavior does not include any options. Specifying the behavior itself enables it.

NOTE: You can only apply this behavior if the property is marked as secure. See [Secure Property Requirements](#) for guidance.)

Related behaviors: [redirect](#) , [modifyIncomingRequestHeader](#) , [cacheKeyQueryParams](#) , [modifyOutgoingResponseHeader](#) , [edgeSideIncludes](#) , [modifyIncomingResponseHeader](#)

subCustomer

Enables various features of the Content Policy API, which allows Akamai partners to specify dynamic content policies for their customers' content, supplementing the rules defined within the current property.

Options

- `dynamicpolicy` (*boolean*): Allows the Content Policy API to dynamically modify content. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `origin` (*boolean*): Allows you to set the origin host. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `caching` (*boolean*): Allows you to modify content caching rules. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `referrer` (*boolean*): Allows you to set the referrer whitelist or blacklist. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `ip` (*boolean*): Allows you to specify an IP whitelist or blacklist. (In the Beta API, please substitute `"true"` and `"false"` string values.)

- `geo` (*boolean*): Allows you to specify a location-based whitelist or blacklist. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `contentrefresh` (*boolean*): Allows you to schedule when custom content is to revalidate. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `modifypath` (*boolean*): Allows you to modify the request path. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `cachekey` (*boolean*): Allows you to set which query parameters are included in the cache key. (In the Beta API, please substitute `"true"` and `"false"` string values.)

Feature previously named: `subcustomerenable`

sureRoute

The [SureRoute](#) feature continually tests different routes between origin and edge servers to identify the optimal path. By default, it conducts *races* to identify alternative paths to use in case of a transmission failure. These races increase origin traffic slightly.

This behavior allows you to configure SureRoute along with a test object to improve delivery of non-cacheable `no-store` or `bypass-cache` content. Since edge servers are already positioned as close as possible to requesting clients, the behavior does not apply to cacheable content.

Options

- `sr_enabled` (*boolean*): Enables the SureRoute behavior, to optimize delivery of non-cached content. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `sr_type` (*enum string*): Specifies the set of edge servers used to test routes, either the default `performance` or a `custommap` that must be provided to you by Akamai Professional Services.
- `custom_map` (*string*): If `sr_type` is `custommap`, this specifies the map string provided to you by Akamai Professional Services, or included as part of the [SiteShield](#) product.
- `sr_test_object_url` (*string*): Specifies the path and filename for your origin's test object to use in races to test routes.

Akamai provides sample test objects for the [Dynamic Site Accelerator](#) and Web Application Accelerator products. If you want to use your own test object, it needs to be on the same origin server as the traffic being served through SureRoute. Make sure it returns a 200 HTTP response and does not require authentication. The file should be an average-sized static HTML file (Content-Type: text/html) that is no smaller than 8KB, with no back-end processing.

NOTE: If you have more than one origin server deployed behind a load balancer, you can configure it to serve the test object directly on behalf of the origin, or route requests to the same origin server to avoid deploying the test object on each origin server.

- `sr_to_host_status` (*enum string*): If set to `incoming_hh` , uses the incoming `Host` header when requesting the SureRoute test object. If set to `other` , use `sr_to_host` to specify a custom `Host` header.
- `sr_to_host` (*string*): If `sr_to_host_status` is `other` , this specifies the custom `Host` header to use when requesting the SureRoute test object.
- `sr_race_stat_ttl` (*duration string*): Specifies the time-to-live to preserve SureRoute race results, typically `30m` . If traffic exceeds a certain threshold after TTL expires, the overflow is routed directly to the origin, not necessarily optimally. If traffic remains under the threshold, the route is determined by the winner of the most recent race.
- `sr_force_ssl_fw` (*boolean*): Forces SureRoute to use SSL when requesting the origin's test object, appropriate if your origin does not respond to HTTP requests, or responds with a redirect to HTTPS. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `sr_stat_key_mode` (*enum string*): When set to `default` , caches race results under the race destination's hostname. If set to `custom` , use `custom_stat_key` to specify a custom hostname.
- `custom_stat_key` (*string*): With `sr_stat_key_mode` set to `custom` , this specifies a hostname under which to cache race results. This may be useful when a property corresponds to many origin hostnames. By default, SureRoute would launch races for each origin, but consolidating under a single hostname runs only one race.

Feature previously named: `sureroute`

tcpOptimization

Enables a suite of optimizations targeting buffers, time-outs, and packet loss that improve transmission performance. This behavior is deprecated, but you should not disable or remove it if present.

This behavior does not include any options. Specifying the behavior itself enables it.

Feature previously named: `tcpt optimizations`

Related behaviors: `cacheError` , `cacheRedirect` , `dnsAsyncRefresh` , `cacheKeyQueryParams` , `removeVary` , `cacheKeyIgnoreCase`

tieredDistribution

This behavior allows Akamai edge servers to retrieve cached content from other Akamai servers, rather than directly from the origin. These interim *parent* servers in the *cache hierarchy* (`ch`) are positioned close to the origin, and fall along the path from the origin to the edge server. Tiered Distribution typically reduces the origin server's load, and reduces the time it takes for edge servers to refresh content.

Options

- `status` (*boolean*): When enabled, activates tiered distribution. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `tdmap` (*enum string*): Optionally map the tiered parent server's location close to your origin: `cheu2` for Europe; `chaus` for Australia; `chapac` for China and the Asian Pacific area; `chwus2` , `chcus2` , and `cheus2` for different parts of the United States. Choose `ch` or `ch2` for more global map. A narrower local map minimizes the origin server's load, and increases the likelihood the requested object is cached. A wider global map reduces end-user latency, but decreases the likelihood the requested object is in any given parent server's cache. (This option cannot apply if the property is marked as secure. See [Secure Property Requirements](#) for guidance.)

Feature previously named: `tiereddistribution`

Related behaviors: `removeVary` , `cacheKeyQueryParams` , `cacheError` , `cacheRedirect` , `cacheKeyIgnoreCase` , `dnsAsyncRefresh`

timeout

Sets the HTTP connect timeout.

Options

- `timeout` (*duration string*): Specifies the timeout, for example `10s`.

Feature previously named: `connecttimeout`

Related behaviors: [removeVary](#), [cacheKeyQueryParams](#), [failAction](#), [healthDetection](#), [cacheError](#), [dnsAsyncRefresh](#)

validateEntityTag

Instructs edge servers to compare the request's ETag header with that of the cached object. If they differ, the edge server sends a new copy of the object. This validation occurs in addition to the default validation of `Last-Modified` and `If-Modified-Since` headers.

Options

- `enabled` (*boolean*): Enables the ETag validation behavior. (In the Beta API, please substitute `"true"` and `"false"` string values.)

Feature previously named: `validateetag`

Related behaviors: [cacheKeyIgnoreCase](#), [prefreshCache](#), [removeVary](#), [redirect](#), [cacheError](#), [cacheKeyQueryParams](#)

verifyTokenAuthorization

Verifies Auth 2.0 tokens.

Options

- `show_advanced` (*boolean*): If enabled, allows you to specify advanced options such as `algorithm` , `escapeHmacInputs` , `ignoreQueryString` , `transitionKey` , and `salt` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `location` (*enum string*): Specifies where to find the token in the incoming request, either `Client_Request_Header` , `Query_String` , or `Cookie` .
- `locationId` (*string*): When `location` is `Client_Request_Header` , specifies the name of the incoming request's header where to find the token.
- `algorithm` (*enum string*): With `show_advanced` enabled, specifies the algorithm that generates the token, either `SHA256` , `SHA1` , or `MD5` , in order of descending security. It must match the method chosen in the token generation code.
- `escapeHmacInputs` (*boolean*): With `show_advanced` enabled, URL-escapes HMAC inputs passed in as query parameters. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `ignoreQueryString` (*boolean*): With `show_advanced` enabled, enabling this removes the query string from the URL used to form an encryption key. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `key` (*string*): The shared secret used to validate tokens, which must match the key used in the token generation code.
- `transitionKey` (*string*): With `show_advanced` enabled, specifies a transition key as a hex value.
- `salt` (*string*): With `show_advanced` enabled, specifies a salt string for input when generating the token, which must match the salt value used in the token generation code.
- `failureResponse` (*boolean*): When enabled, sends an HTTP error when an authentication test fails. (In the Beta API, please substitute `"on"` and `"off"` string values.)

Feature previously named: `token_auth_verify`

Related behaviors: [denyAccess](#) , [modifyOutgoingRequestHeader](#) , [modifyOutgoingResponseHeader](#) , [cacheError](#) , [rewriteUrl](#) , [modifyIncomingRequestHeader](#)

visitor_prioritization

The [Visitor Prioritization Cloudlet](#) is designed to decrease abandonment by providing a user-friendly waiting room experience. Assuming cloudlets are available on your contract, choose **Configure⇒Cloudlets** to control Visitor Prioritization within Control Center. Otherwise use the [Cloudlets API](#) to configure it programmatically.

NOTE: If you want to serve non-HTML API content, such as JSON blocks, see the [asset_prioritization](#) behavior.

Options

- `status` (*boolean*): Enables the Visitor Prioritization behavior. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `nimbus_policy_token` (*string*): Specifies the Cloudlets policy name, available in Control Center's **Cloudlets Policy Manager**.
- `throttled_response_code` (*enum string*): Specifies the response code for requests sent to the waiting room, either `200` or `503` .
- `use_throttled_cpcode` (*boolean*): When enabled, allows you to assign a different CP code that tracks any requests that are sent to the waiting room. (In the Beta API, please substitute `"on"` and `"off"` string values.)
- `throttled_cpcode` (*object*): With `use_throttled_cpcode` enabled, specifies a `cpcode` object for requests sent to the waiting room, including a numeric `id` key and a descriptive `name` :

```
"cpcode": {
  "id" : 12345,
  "name" : "sent to waiting room"
}
```

- `netstorage` (*object*): Specifies the NetStorage domain for the waiting room page. For example:

```
"nshostname": {
  "id" : "id_string",
  "name" : "Example Downloads",
  "downloadDomainName" : "example.download.akamai.com",
  "cpCode" : 12345
}
```

- `wr_directory` (*string*): Specifies the NetStorage directory that contains the static waiting room page, with no trailing slash character.
- `label` (*string*): Specify a suffix to add to the Cloudlet policy's cookies, potentially useful to distinguish one policy from another within the same property.
- `wr_cookie_expiry_seconds` (*number within 0-100 range*): Specifies the duration of the Cloudlet's cookie that holds users in the waiting room. (It sends users back to the waiting room if they refresh the waiting room page.)
- `privileged_cookie_expiry_seconds` (*number within 0-100 range*): Specifies the duration of the privilege cookie, which bypasses Visitor Prioritization for subsequent requests once a user has been allowed through to the site.
- `rolling_privilege_window` (*boolean*): If enabled, users receive a fresh cookie that matches the duration of the Allowed User Cookie. Disable to set a fixed cookie time. (In the Beta API, please substitute "1" and "0" string values.)

Related behaviors: [prefetchable](#) , [prefreshCache](#) , [gzipResponse](#) , [denyAccess](#) , [adaptiveImageCompression](#)

watermarkUrl

Aliases a token to a watermark image URL.

Options

- `token` (*string*): Specifies the string token.
- `image_url` (*string*): Specifies the URL for the watermark image.

Feature previously named: `watermark_tokens`

Related behaviors: [edgeImageConversion](#) , [cacheError](#) , [cacheKeyQueryParams](#) , [modifyOutgoingResponseHeader](#) , [verifyTokenAuthorization](#) , [responseCookie](#)

webApplicationFirewall

This behavior implements a suite of security features that blocks threatening HTTP and HTTPS requests. Use it as your primary firewall, or in addition to existing security measures. Only one referenced configuration is allowed per property, so this behavior typically belongs as part of its default rule.

Options

- `wafconfig` (*object*): An object featuring details about your firewall configuration, for example:

```
"wafconfig": {  
  "configId" : 1,  
  "productionStatus" : "Active",  
  "productionVersion" : 1,  
  "stagingStatus" : "Active",  
  "stagingVersion" : 1  
}
```

Feature previously named: `waf`

Related behaviors: [redirect](#) , [cacheKeyQueryParams](#) , [cacheError](#) , [cacheKeyIgnoreCase](#) , [removeVary](#) , [denyAccess](#)

webdav

Web-based Distributed Authoring and Versioning (WebDAV) is a set of extensions to the HTTP protocol that allows users to collaboratively edit and manage files on remote web

servers. This behavior enables WebDAV, and provides support for the following additional request methods: PROPFIND, PROPPATCH, MKCOL, COPY, MOVE, LOCK, and UNLOCK.

Options

- `status` (*boolean*): Enables the WebDAV behavior. (In the Beta API, please substitute "on" and "off" string values.)

Related behaviors: `allowPut` , `allowDelete` , `allHttpInCacheHierarchy` , `cacheKeyIgnoreCase` , `cacheKeyQueryParams` , `cacheError`

v2015-08-17 criteria

v2015-08-17 criteria

The following represents all rule criteria the Property Manager API supports. The set available to you is determined by the product and modules associated with the property. Use the [List Available Criteria](#) operation to get this information.

This reference specifies match criteria used in the `v2015-08-17 rule format`. See the [most recent set of criteria](#), which corresponds to the `latest` rule format.

bucket

This [read-only criteria](#) matches a specified percentage of requests when used with the required accompanying `spdy` behavior. Contact Akamai Professional Services for help configuring it.

Options

- `bucket` (*number within 0-100 range*): Specifies the percentage of SPDY requests to match.

cacheability

Matches the current cache state.

NOTE: Any `no-store` or `bypass-cache` HTTP headers set on the origin's content overrides properties' `cacheing` instructions.

Options

- `result` (*boolean*): When disabled, reverses the match so that the cache state does *not* match the specified `value` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*enum string*): Content's cache is enabled (`cacheable`) or not (`no-store`), or else is ignored (`bypass-cache`).

clientIp

Matches the IP number of the requesting client.

Options

- `result` (*boolean*): When disabled, reverses the match so that the IP is *not* among the specified set. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*array of string values*): IP or CIDR block, for example: `71.92.0.0/14` .
- `use-headers` (*boolean*): When connecting via a proxy server as determined by the `X-Forwarded-For` header, enabling this option matches the connecting client's IP address rather than the original end client specified in the header.

Feature previously named: `clientip`

clientIpVersion

Matches the version of the IP protocol used by the requesting client.

Options

- `value` (*enum string*): The IP version of the client request, either `4` or `6`.
- `use-headers` (*boolean*): When connecting via a proxy server as determined by the `X-Forwarded-For` header, enabling this option matches the connecting client's IP address rather than the original end client specified in the header.

Feature previously named: `clientipversion`

cloudletsOrigin

Allows Cloudlets Origins, referenced by label, to define their own criteria to assign custom origin definitions. The criteria may match, for example, for a specified percentage of requests defined by the cloudlet to use an alternative version of a website.

This criteria must be paired with a sibling `origin` definition. It should not appear with any other criteria, and an `allowCloudletsOrigins` behavior must appear within a parent rule.

Options

- `originId` (*string*): The Cloudlets Origins identifier, limited to alphanumeric and underscore characters.

Feature previously named: `conditionalOriginId`

contentDeliveryNetwork

A [read-only criteria](#) that specifies the type of Akamai network handling the request. Contact Akamai Professional Services for help configuring it.

Options

- `result` (*boolean*): When disabled, reverses the match so that the request is *not* served from the specified `network` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `network` (*enum string*): Match requests served from either the `production` network, or when `use-staging` is in effect.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field.

Feature previously named: `network_match`

contentType

Matches the HTTP response header's `Content-Type` .

Options

- `result` (*boolean*): When disabled, reverses the match so that the `Content-Type` is *not* among the specified set. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*array of string values*): `Content-Type` response header value, for example `text/html` .
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field, so that specifying `text/*` matches both `text/html` and `text/css` .
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

Feature previously named: `contenttype`

deviceCharacteristic

Match various aspects of the device or browser making the request.

Based on the value of the `characteristic` option, the expected value is either a boolean, a number, or a string, possibly representing a version number. Each type of value requires a different `value-` field:

- `value-boolean` specifies a boolean value. - `value-number` specifies a numeric value. - `value-version` specifies a version number string value. - `value-string` specifies any other string value, to which the `value-case` and `value-wildcard` options apply.

Options

- `characteristic` (*enum string*): Aspect of the device or browser to match. The following values are boolean:
 - `accept_third_party_cookie`
 - `ajax_support_javascript`
 - `cookie_support`
 - `dual_orientation` (whether the display adapts to portrait/landscape orientation)
 - `full_flash_support`
 - `gif_animated`
 - `is_mobile`
 - `is_tablet` (subset of `is_mobile`)
 - `is_wireless_device`

The following are numeric values:

- `physical_screen_height` (millimeters)
- `physical_screen_width` (millimeters)
- `resolution_height` (pixels)
- `resolution_width` (pixels)
- `xhtml_support_level`

The following are version string values:

- `device_os_version`
- `mobile_browser_version`

The following are string values:

- `brand_name` (such as `Samsung` or `Apple`)
- `device_os`
- `marketing_name` (such as `Samsung Illusion`)
- `mobile_browser`
- `model_name` (such as `SCH-I110`)
- `op-string` (*boolean*): Specifies string-matching operators: `beginswith` , `endswith` , and `contains` for substring matches, or `true` and `false` for complete matches. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `op-number` (*enum string*): When the `characteristic` expects a numeric value, compares the specified `value-number` against the matched client, using the following operators: `eq` , `ge` , `gt` , `le` , `lt` , `ne` .
- `op-version` (*enum string*): When the `characteristic` expects a version string value, compares the specified `value-version` against the matched client, using the following operators: `version-eq` , `version-ge` , `version-gt` , `version-le` , `version-lt` , `version-ne` .
- `value-boolean` (*boolean*): When the `characteristic` expects a boolean value, this sets the value to `true` or `false` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value-string` (*array of string values*): When the `characteristic` expects a string, this specifies the set of values.
- `value-number` (*number*): When the `characteristic` expects a numeric value, this specifies the number.
- `value-version` (*string*): When the `characteristic` expects a version number, this specifies it as a string.
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value-string` field.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value-string` field.

Feature previously named: `devicecharacteristics`

fileExtension

Matches the requested filename's extension, if present.

Options

- `result` (*boolean*): When disabled, reverses the set of matching results so that the requested file does *not* match any of the specified extensions. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*array of string values*): An array of file extension strings, with no leading dot characters, for example `png` , `jpg` , `jpeg` , and `gif` .
- `case` (*boolean*): When enabled, the match is case-sensitive.

Feature previously named: `ext`

filename

Matches the requested filename, or test whether it is present.

Options

- `result` (*enum string*): If `true` or `false` , matches whether the `value` matches. If `empty` or `not_empty` , matches whether the specified filename is part of the path.
- `value` (*array of string values*): Matches the filename component of the request URL. Wildcards are allowed, where `?` matches a single character and `*` matches more than one. For example, specify `filename.*` to accept any extension.
- `case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

hostname

Matches the requested hostname.

Options

- `result` (*boolean*): When disabled, reverses the match so that the hostname is *not* among the specified set. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `host` (*array of string values*): A list of hostnames. Wildcards match, so `*.example.com` matches both `m.example.com` and `www.example.com`.

Feature previously named: `hoit`

matchAdvanced

A [read-only criteria](#) that specifies Akamai XML metadata. It can only be configured on your behalf by Akamai Professional Services.

Options

- `description` (*string*): A human-readable description of what the XML block does.
- `open_xml` (*string*): An XML string that opens the relevant block.
- `close_xml` (*string*): An XML string that closes the relevant block.

Feature previously named: `advancedmatch`

matchCpCode

Match the assigned content provider code.

Options

- `cpcode` (*object*): Specifies an object that encodes information about the `cpcode` to match, including an `id` key and a descriptive `name` :

```
"cpcode": {  
  "id" : 12345,  
  "name" : "my cpcode"  
}
```

Feature previously named: `matchcpcode`

matchResponseCode

Match a set or range of HTTP response codes.

Options

- `result` (*boolean*): When disabled, reverses the match so that the response code does *not* match the `value` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value-type` (*enum string*): The type of match. If set to `checklist` , matches discrete codes specified in the `value` field. If set to `range` , considers `range-start-value` and `range-end-value` instead.
- `value` (*array of string values*): A set of response codes to match, for example `["404","500"]` .
- `range-start-value` (*string*): Specifies the start of a range of responses when `value-type` is set to `range` . For example: `400` to match anything from `400` to `500` .
- `range-end-value` (*string*): Specifies the end of a range of responses when `value-type` is set to `range` . For example: `500` to match anything from `400` to `500` .

Feature previously named: `responsecode`

originTimeout

Matches when the origin responds with a timeout error.

Options

- `result` (*enum string*): Toggle the match. When this option is disabled, the match occurs when there is *no* timeout.

Feature previously named: `origintimeout`

path

Matches the URL's non-hostname path component.

Options

- `result` (*boolean*): When disabled, reverses the match so that the requested pathname does *not* match any of those specified as a `value`. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*array of string values*): Matches the URL path, excluding leading hostname and trailing query parameters. The path is relative to the server root, for example `/blog`. The `value` accepts `*` or `?` wildcard characters, for example `/blog/*/2014`.
- `case` (*boolean*): When enabled, the match is case-sensitive.

Feature previously named: `wildcard`

queryStringParameter

Matches query string field names or values.

Options

- `name` (*string*): The name of the query field, for example, `q` in `?q=string`.
- `result` (*enum string*): Narrows the match according to the following criteria:
 - `exists` or `doesntexist` tests whether the query field's `name` is present in the requesting URL.
 - `true` or `false` tests whether the field's `value` string matches.
 - `islessthan` or `ismorethan` matches ranges when the `value` is numeric.
 - `isbetween` also matches numeric ranges, but considers `lowerbound` and `upperbound` fields rather than `value`.
- `value` (*array of string values*): The value of the query field, for example, `string` in `?q=string`.
- `lowerbound` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's minimum value.
- `upperbound` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's maximum value.
- `name-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `name` field.
- `name-case` (*boolean*): When enabled, the match is case-sensitive for the `name` field.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field.
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.
- `value-escape` (*boolean*): When enabled, matches when the `value` is URL-escaped.

Feature previously named: `querystring`

random

Matches a specified percentage of requests. Use this match to apply behaviors to a percentage of your incoming requests that differ from the remainder, useful for A/B testing, or to offload traffic onto different servers.

Options

- `bucket` (*number within 0-100 range*): Specify a percentage of random requests to which to apply a behavior. Any remainders do not match.

requestCookie

Match the cookie name or value passed with the request.

Options

- `name` (*string*): The name of the cookie, for example, `visitor` in `visitor:anon`.
- `result` (*enum string*): Narrows the match according to the following criteria:
 - `exists` or `doesntexist` tests whether the cookie `name` exists.
 - `true` or `false` tests whether the field's `value` string matches.
 - `isbetween` tests whether a numeric cookie `value` falls between `lower` and `upper`.
- `value` (*string*): The cookie's value, for example, `anon` in `visitor:anon`.
- `lower` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's minimum value.
- `upper` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's maximum value.
- `name-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `name` field.
- `name-case` (*boolean*): When enabled, the match is case-sensitive for the `name` field.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field.

- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

Feature previously named: `requestcookie`

requestHeader

Match HTTP header names or values.

Options

- `name` (*string*): The name of the request header, for example `Accept-Language`.
- `result` (*enum string*): Narrows the match according to the following criteria:
 - `exists` or `doesntexist` tests whether the field `name` exists.
 - `true` or `false` tests whether the field's `value` string matches.
- `value` (*array of string values*): The request header's value, for example `en-US` when the header `name` is `Accept-Language`.
- `name-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `name` field.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field.
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

Feature previously named: `requestheader`

requestMethod

Specify the request's HTTP verb.

Options

- `result` (*boolean*): When disabled, reverses the match so that the request method does *not* match the `value` . (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*enum string*): Any of the following HTTP methods: `CONNECT` , `COPY` , `GET` , `HEAD` , `HTTP_DELETE` , `OPTIONS` , `POST` , `PUT` , and `TRACE` . Also the following WebDAV headers: `DAV_COPY` , `DAV_LOCK` , `DAV_MKCOL` , `DAV_MOVE` , `DAV_PROPFIND` , `DAV_PROPPATCH` , and `DAV_UNLOCK` .

Feature previously named: `requestmethod`

requestProtocol

Matches whether the request uses the HTTP or HTTPS protocol.

Options

- `value` (*enum string*): Either `HTTP` or `HTTPS` .

Feature previously named: `requestprotocol`

responseHeader

Match HTTP header names or values.

Options

- `name` (*string*): The name of the response header, for example `Content-Type` .
- `result` (*enum string*): Narrows the match according to the following criteria:

- `exists` or `doesntexist` tests whether the HTTP field `name` is present.
- `true` or `false` tests whether the field's `value` string matches.
- `islessthan` or `ismorethan` matches ranges when the `value` is numeric.
- `isbetween` also matches numeric ranges, but considers `lowerbound` and `upperbound` fields rather than `value`.
- `value` (*array of string values*): The response header's value, for example `application/x-www-form-urlencoded` when the header `name` is `Content-Type`.
- `lowerbound` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's minimum value.
- `upperbound` (*string*): When the `value` is numeric and the `result` is set to `isbetween`, this field specifies the match's maximum value.
- `name-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `name` field.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field.
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

Feature previously named: `responseheader`

time

Specifies ranges of times during which the request occurred.

Options

- `behavior` (*enum string*): Specifies how to define the range of time:
 - `beginning` if the duration is indefinite, using the value of `begindate`.
 - `between` sets a single range between two dates, using the values of `begindate` and `enddate`.

- `lasting` also sets a single range, but based on duration relative to the starting time. It relies on the values of `lastingdate` and `lastingduration` .
- `repeating` allows a `lasting` -style range to repeat at regular intervals. It relies on the values of `repeatbegindate` , `repeatduration` , and `repeatinterval` .
- `repeatinterval` (*duration string*): Sets the time between each repeating time period's starting points when `behavior` set to `repeating` .
- `repeatduration` (*duration string*): Sets the duration of each repeating time period with `behavior` set to `repeating` .
- `lastingduration` (*duration string*): With `behavior` set to `lasting` , specifies the end of a time period as a duration relative to the `lastingdate` .
- `lastingdate` (*ISO 8601 format date/time string*): Sets the start of a fixed time period with `behavior` set to `lasting` .
- `repeatbegindate` (*ISO 8601 format date/time string*): Sets the start of the initial time period with `behavior` set to `repeating` .
- `applydst` (*boolean*): Adjusts the start time plus repeat interval to account for daylight saving time. Applies when the current time and the start time use different systems, daylight and standard, and the two values are in conflict. (In the Beta API, please substitute "on" and "off" string values.)
- `begindate` (*ISO 8601 format date/time string*): Sets the start of a time period with `behavior` set to `beginning` or `between` .
- `enddate` (*ISO 8601 format date/time string*): Sets the end of a fixed time period with `behavior` set to `between` .

tokenAuthorization

Match on Auth Token 2.0 verification results.

Options

- `result` (*enum string*): Either `pass` if there are no errors, `fail` for any errors specified by `statusList`, or `failAll` if there are any errors.
- `statusList` (*array of string values*): Match specific failure cases:

<code>failure:expired_token</code>	<code>failure:missing_expiration_time</code>
<code>failure:invalid_acl_delimiter</code>	<code>failure:missing_token</code>
<code>failure:invalid_delimiter</code>	<code>failure:missing_url</code>
<code>failure:invalid_hmac</code>	<code>failure:need_url_xor_acl</code>
<code>failure:invalid_hmac_key</code>	<code>failure:token_not_valid_yet</code>
<code>failure:invalid_ip</code>	<code>failure:unauthorized_ip</code>
<code>failure:invalid_token</code>	<code>failure:unauthorized_url</code>
<code>failure:invalid_url</code>	<code>failure:unsupported_version</code>

Feature previously named: `token_auth_match`

userAgent

Matches the user agent string that helps identify the client browser and device.

Options

- `result` (*boolean*): When disabled, reverses the match so that the client's user agent does *not* match the `value`. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `value` (*array of string values*): The `User-Agent` header's value. For example, `Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)`.
- `value-wildcard` (*boolean*): When enabled, allows `*` and `?` wildcard matches in the `value` field. For example, `*Android*`, `*iPhone5*`, `*Firefox*`, or `*Chrome*`.
- `value-case` (*boolean*): When enabled, the match is case-sensitive for the `value` field.

Feature previously named: `useragent`

userLocation

The client browser's approximate geographic location, determined by looking up the IP address in a database.

Options

- `field` (*enum string*): Indicates the geographic scope: `CONTINENT` , `COUNTRY_CODE` , or `REGION_CODE` for states or provinces within a country.
- `result` (*boolean*): When disabled, reverses the match so that the client's location does *not* match the set of values. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `country-value` (*array of string values*): ISO 3166-1 country code, such as `US` or `CN` .
- `continent-value` (*array of string values*): Continent code, one of `AF` (Africa), `AS` (Asia), `EU` (Europe), `NA` (North America), `SA` (South America), `OC` (Oceania), or `OT` (Antarctica).
- `region-value` (*array of string values*): ISO 3166 country and region codes, for example `US:MA` for Massachusetts or `JP:13` for Tokyo.
- `check-ips` (*enum string*): Specifies which IP addresses determine the user's location:
 - `connecting` considers the connecting client's IP address.
 - `headers` considers IP addresses specified in the `X-Forwarded-For` header, succeeding if any of them match.
 - `both` behaves like `headers` , but also considers the connecting client's IP address.
- `use-headers` (*enum string*): When connecting via a proxy server as determined by the `X-Forwarded-For` header, enabling this option matches the connecting client's IP address rather than the original end client specified in the header.

Feature previously named: `userlocation`

userNetwork

Matches details of the network over which the request was made, determined by looking up the IP address in a database.

Options

- `field` (*enum string*): The type of information to match, either `BW` for *bandwidth*, `NETWORK` for specific networks, or more general `NETWORK_TYPE`.
- `result` (*boolean*): When disabled, reverses the match so that the client's network does *not* match the `value`. (In the Beta API, please substitute `"true"` and `"false"` string values.)
- `network-type-value` (*array of string values*): Specifies the basic type of network, either `cable`, `dialup`, `dsl`, `fios`, `isdn`, `mobile`, or `uverse`.
- `network-value` (*array of string values*): Any set of specific networks:

airtel	fibertel
alpha_internet	francetelecom
altitudetelecom	free
aol	freecom
arnet	h3g
asahi	hinet
att	ibm
bellcanada	idecnet
biglobe	ij4u
bitmailer	infosphere
bouygues	janet
brighthouse	jazztel
bskyb	justnet
bt	livedoor
cablevision	mci
cernet	mediacom
chinamobile	mediaone
chinanet	microsoft
chinaunicom	mil
clearwire	nerim
colt	newnet
comcast	@nifty
completel	numericable
compuserve	ocn
covad	odn
dion	ono
dreamnet	panasonic-hi-ho
dtag	plala
dti	plusnet
earthlink	prodigy
easynet	qwest
eurociber	rediris
fastweb	renater

reserved	tikitiki
retevision	timewarner
roadrunner	tiscali
rogers	tmobile
seednet	turktelekom
seikyo_internet	uni2
sfr	uninet
shaw	upc
so-net	uunet
sprint	verizon
suddenlink	virginmedia
talktalk	vodafone
telefonica	wakwak
telstra	wind
terramexico	windstream
ti	zero

- `bandwidth-value` (*array of string values*): Bandwidth range in bits per second, either `1` , `57` , `257` , `1000` , `2000` , or `5000` .
- `check-ips` (*enum string*): Specifies which IP addresses determine the user's network:
 - `connecting` considers the connecting client's IP address.
 - `headers` considers IP addresses specified in the `X-Forwarded-For` header, succeeding if any of them match.
 - `both` behaves like `headers` , but also considers the connecting client's IP address.
- `use-headers` (*enum string*): When connecting via a proxy server as determined by the `X-Forwarded-For` header, enabling this option matches the connecting client's IP address rather than the original end client specified in the header.

Feature previously named: `usernetwork`

Notice

Akamai secures and delivers digital experiences for the world's largest companies. Akamai's Intelligent Edge Platform surrounds everything, from the enterprise to the cloud, so customers and their businesses can be fast, smart, and secure. Top brands globally rely on Akamai to help them realize competitive advantage through agile solutions that extend the power of their multi-cloud architectures. Akamai keeps decisions, apps, and experiences closer to users than anyone — and attacks and threats far away. Akamai's portfolio of edge security, web and mobile performance, enterprise access, and video delivery solutions is supported by unmatched customer service, analytics, and 24/7/365 monitoring. To learn why the world's top brands trust Akamai, visit www.akamai.com, blogs.akamai.com, or [@Akamai](https://twitter.com/Akamai) on Twitter. You can find our global contact information at www.akamai.com/locations.

Akamai is headquartered in Cambridge, Massachusetts in the United States with operations in more than 57 offices around the world. Our services and renowned customer care are designed to enable businesses to provide an unparalleled Internet experience for their customers worldwide. Addresses, phone numbers, and contact information for all locations are listed on www.akamai.com/locations.

© 2022 Akamai Technologies, Inc. All Rights Reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system or translated into any language in any form by any means without the written permission of Akamai Technologies, Inc. While precaution has been taken in the preparation of this document, Akamai Technologies, Inc. assumes no responsibility for errors, omissions, or for damages resulting from the use of the information herein. The information in this document is subject to change without notice. Without limitation of the foregoing, if this document discusses a product or feature in beta or limited availability, such information is provided with no representation or guarantee as to the matters discussed, as such products/features may have bugs or other issues.

Akamai and the Akamai wave logo are registered trademarks or service marks in the United States (Reg. U.S. Pat. & Tm. Off). Akamai Intelligent Edge Platform is a trademark in the United States. Products or corporate names may be trademarks or registered trademarks of other companies and are used only for explanation and to the owner's benefit, without intent to infringe.

Published December 21, 2022