

Edgecast

Rules Engine User Guide

edgecast

Disclaimer

Care was taken in the creation of this guide. However, Edgecast cannot accept any responsibility for errors or omissions. There are no warranties, expressed or implied, including the warranty of merchantability or fitness for a particular purpose, accompanying this product.

Trademark Information

EDGECAST is a registered trademark of Edgecast Inc.

About This Guide

Rules Engine (version 4) User Guide

Version 1.91

6/2/2022

© 2022 Edgecast Inc. All rights reserved.

Table of Contents

Rules Engine	1
Overview	1
Getting Started.....	1
Getting to Know Rules Engine.....	2
Key Terms.....	2
Setup Workflow	3
Draft	5
Rule	6
Administering Drafts and Rules	24
Policy	31
Interpreting XML Data	32
Deploy Request	36
Deploy Request Submission.....	36
Deploy Request States	38
Deploy Request History	40
Deploy Request State Notifications	40
Environment.....	41
Production.....	41
Staging.....	41
Match Conditions and Features.....	44
Overview	44
Syntax.....	44
Literal Values.....	44
Wildcard Values (Special Characters)	45
Regular Expressions	46
Match Conditions.....	47

Types of Match Conditions	47
Always	52
Device.....	52
Location.....	79
Origin.....	91
Request	92
URL	106
Features	131
Types of Features.....	131
Access.....	135
Caching.....	139
Comment	160
Headers	161
Logs	167
Optimizer	169
Origin.....	170
Specialty	171
URL	178

Rules Engine

Overview

Rules Engine is designed to be the final authority on how specific types of requests are processed by the CDN.

The MCC contains a variety of settings that may be leveraged to configure how content is delivered through the CDN. However, your unique working environment may require an additional level of customization. For this reason, we provide an interface through which you can create custom policies that will override your MCC configuration, the default behavior of our edge servers, and even the instructions provided by an origin server.

Common uses:

- Override or define a custom cache policy.
- Secure or deny requests for sensitive content.
- Redirect requests.
- Store custom log data.
- **ADN:** Enable a Web Application Firewall profile.

Note: Although Rules Engine is supported on all HTTP-based platforms (i.e., HTTP Large, HTTP Small, and ADN), each configuration is specific to a single platform. For example, a policy created for the HTTP Large platform will not affect how requests to the HTTP Small platform are handled and vice-versa.

Getting Started

Setting up Rules Engine involves the following steps:

1. Create a draft that identifies the type of requests to which a set of actions will be applied.
2. Once the draft has been finalized, convert it into a policy by locking it.
3. Optional. Test the configuration by deploying the policy to the Staging environment.
4. Apply the configuration to live traffic by deploying the policy to the Production environment.

Getting to Know Rules Engine

The configuration of Rules Engine requires an understanding of key terms and the setup workflow.

Key Terms

The following key terms are critical to understanding the workflow through which custom rules that control CDN behavior may be applied to live or test traffic.

Draft

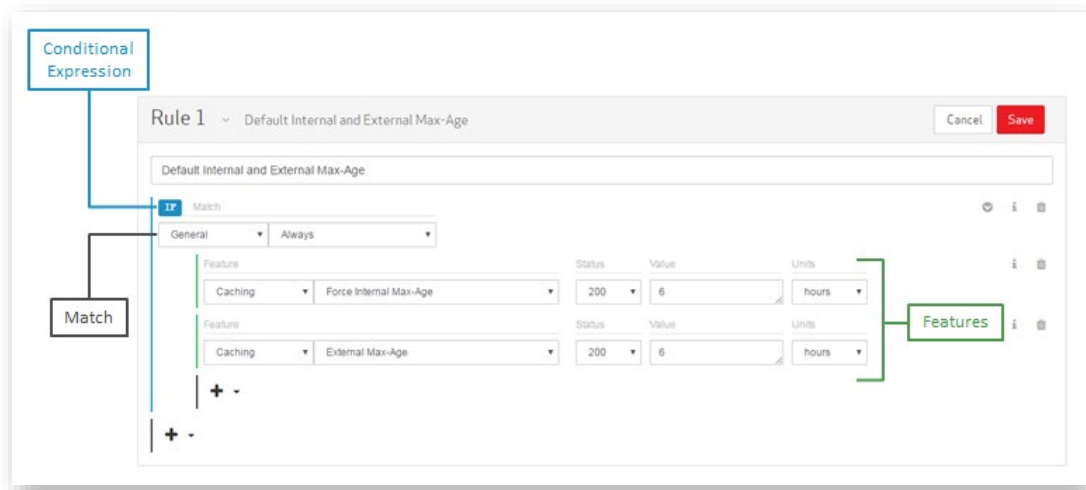
A draft of a policy consists of one or more rules meant to identify requests and the set of actions that will be applied to them. A draft is a work in progress that allows frequent configuration updates without impacting site traffic. Once a draft is ready to be finalized, it should be converted into a read-only policy.

Rule

A rule identifies one or more types of requests and the set of actions that will be applied to them. It consists of:

- A set of conditional expressions that define the logic through which requests are identified.
- A set of match conditions that define the criteria used to identify requests.
- A set of features that define how the CDN will handle the above requests.

These elements are identified in the following illustration.



Sample Rule Illustrating Basic Terminology

Policy

A policy, which consists of a set of read-only rules, provides the means to:

- Create, store, and manage multiple variants of your rules.
- Roll back to a previously deployed version.
- Prepare event-specific rules in advance (e.g., a rule that redirects traffic as a result of a customer origin maintenance.)

Note: Although only a single policy per environment is allowed, policies may be deployed as needed.

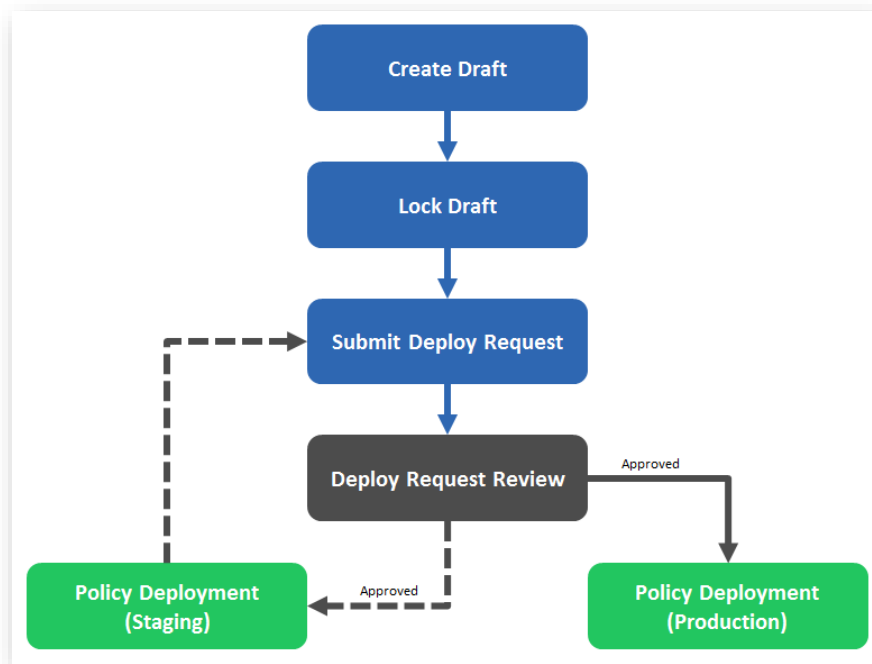
Deploy Request

A deploy request provides a simple and streamlined procedure through which a policy may be quickly applied to the Staging or Production environment. A history of deploy requests is provided to facilitate the tracking of changes applied to those environments.

Note: Only requests that do not pass our automated validation and error detection system will require manual review and approval.

Setup Workflow

Setting up Rules Engine involves deploying a policy to either the Staging or Production environment. The workflow through which this deployment takes place is illustrated below.



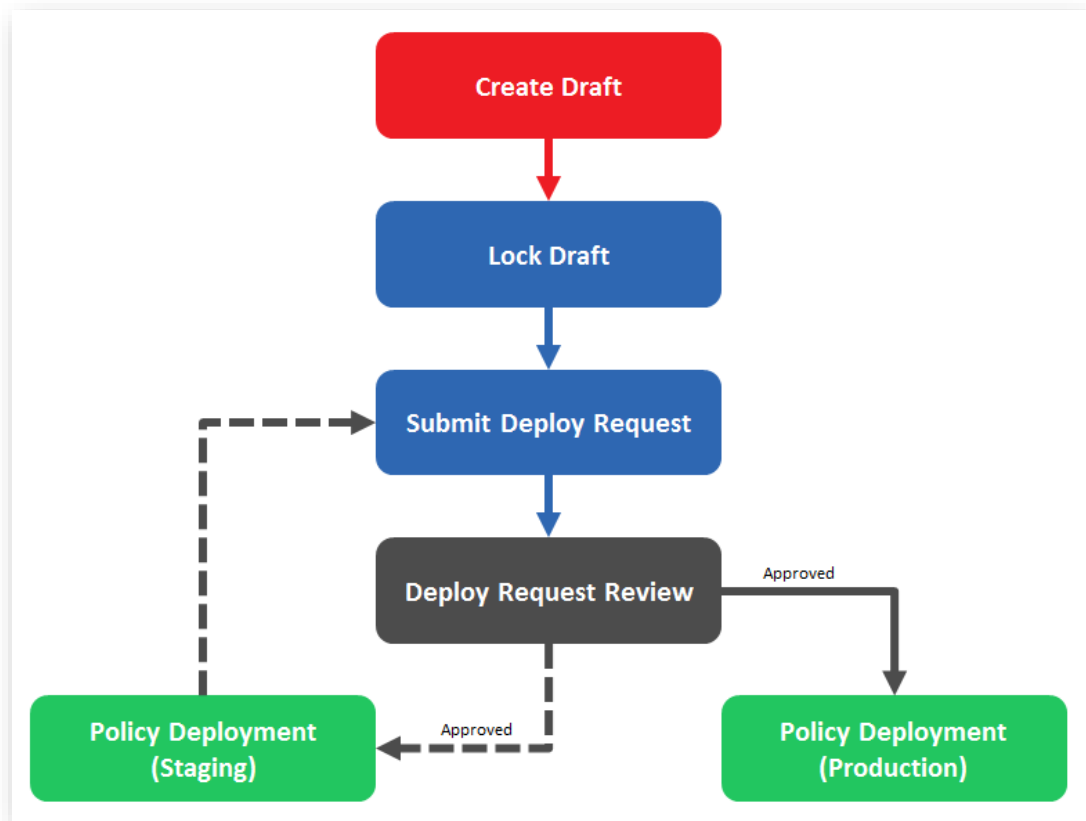
Setup Workflow

The steps in the above illustration are described below.

Name	Description
Create Draft	A draft consists of a set of rules that define how requests for your content should be handled by the CDN.
Lock Draft	Once a draft has been finalized, it should be locked and converted into a read-only policy.
Submit Deploy Request	<p>A deploy request allows a policy to be applied to either test or production traffic.</p> <p>Submit a deploy request to either the Staging or Production environment.</p>
Deploy Request Review	<p>A deploy request undergoes automated validation and error detection.</p> <hr/> <p>Note: Although the majority of deploy requests are automatically approved, manual review is required for more complex policies.</p> <hr/>
Policy Deployment (Staging)	<p>Upon approval of a deploy request to the Staging environment, a policy will be applied to the Staging environment. This environment allows a policy to be tested against mock site traffic.</p> <hr/> <p>Tip: Once the policy is ready to be applied to live site traffic, a new deploy request for the Production environment should be submitted.</p> <hr/>
Policy Deployment (Production)	Upon approval of a deploy request to the Production environment, a policy will be applied to the Production environment. This environment allows a policy to act as the final authority for determining how the CDN should handle live traffic.

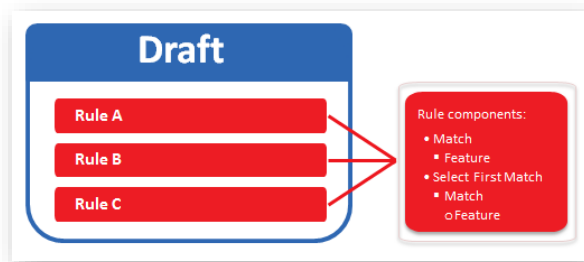
Draft

The first step towards customizing how the CDN handles requests is to create a draft of a policy. A draft must contain one or more rules that will eventually be applied to either test or live site traffic.



Deployment Workflow

A draft consists of rules, match conditions, features, and select first match sections. The relationship between these elements is illustrated below.



Components of a Draft

Key information:

- A draft allows the administration of rules without impacting Production traffic.
- A draft may be viewed and/or modified at any time.
- A draft may be saved in any state. In other words, rule validation will not be enforced when saving a draft.
- Frequent updates may be applied to a draft.
- Once a draft accurately reflects the desired configuration, it should be converted into a read-only policy. This step is a precursor to submitting a deploy request to the Staging or Production environment.
 - This conversion process is irreversible and will prevent additional changes from being applied to it.
 - A new draft may be created from a read-only policy by copying it. In essence, a policy may be used as a template for a new draft. This allows you to easily fine-tune your configuration even after creating a read-only policy.

Rule

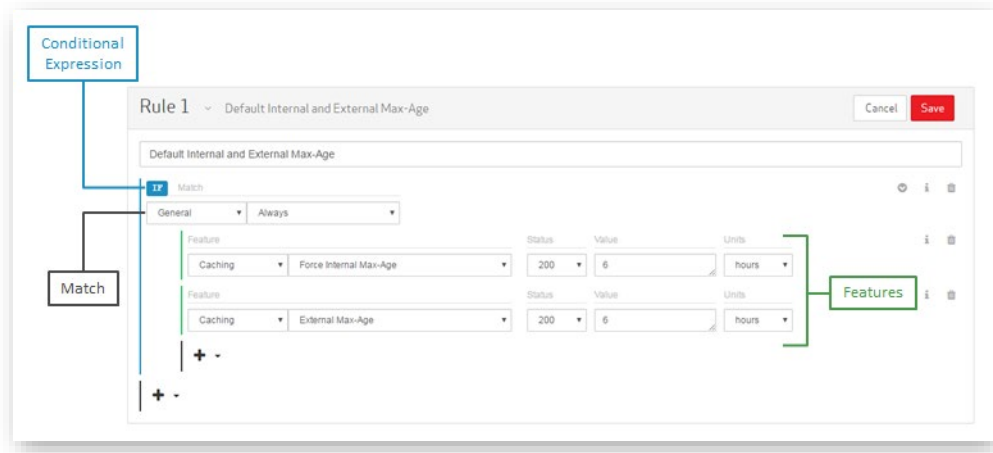
A draft provides the framework through which one or more rules may be created. Each rule provides instructions on how requests will be handled by the CDN. It does this by identifying specific types of requests (i.e., match) and the set of actions (i.e., features) that will be applied to them.

Note: One or more rules may be administered from within a single draft.

A rule consists of:

- A set of conditional expressions that define the logic through which requests are identified.
- A set of matches that define a request type.
- A set of features that define how the CDN will handle the above requests.

These elements are highlighted in the following illustration.



Sample Rule Illustrating Basic Terminology

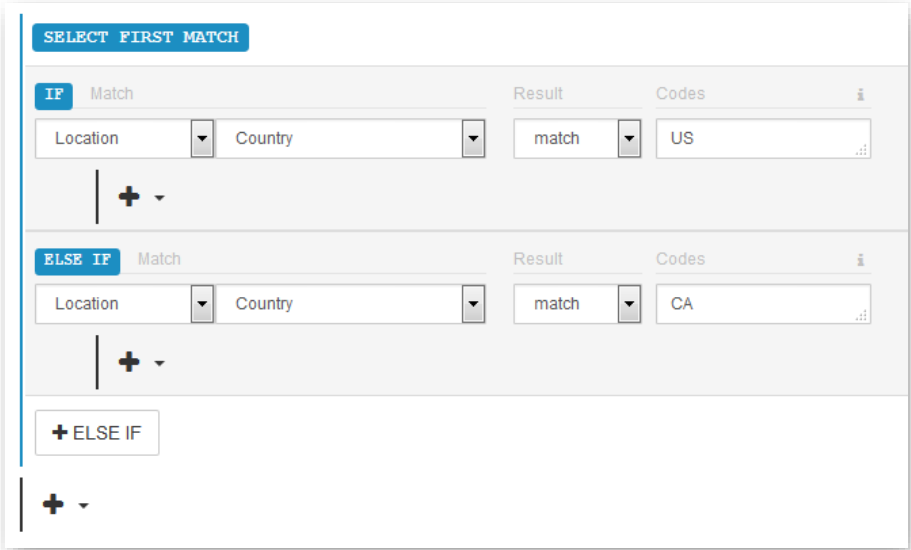
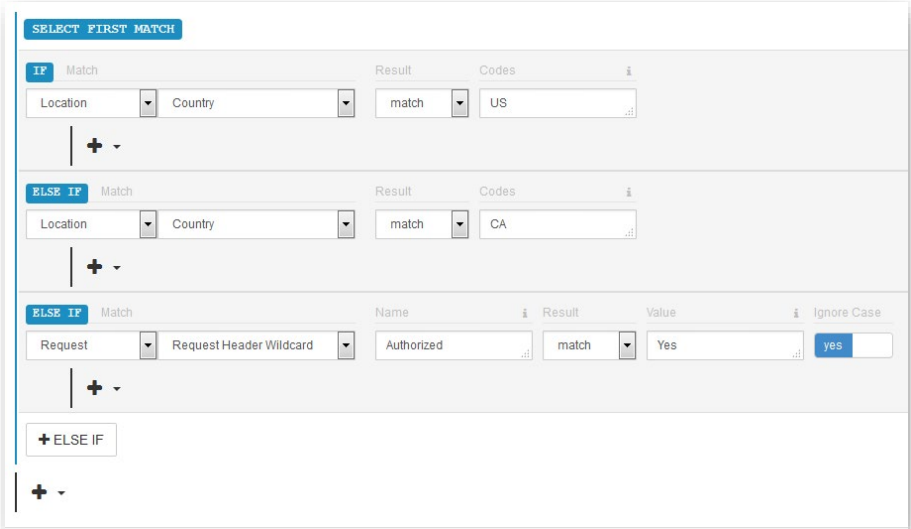
Request Identification

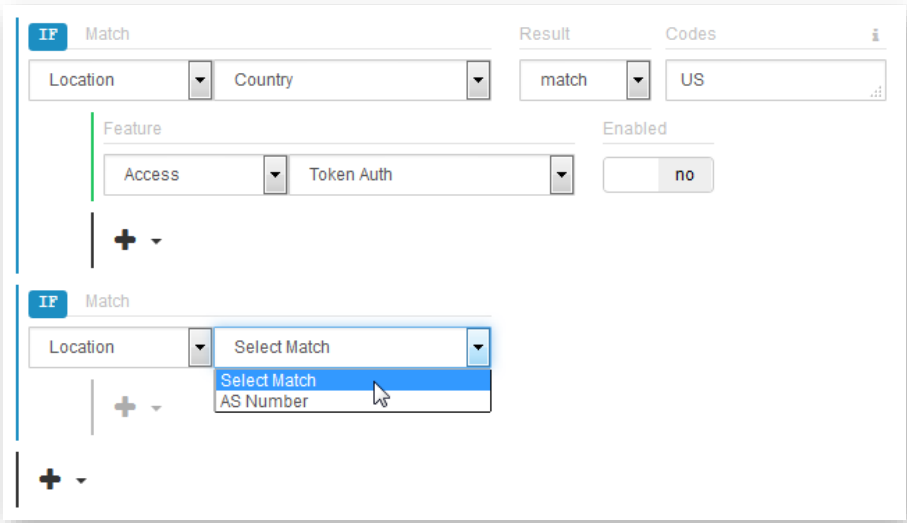
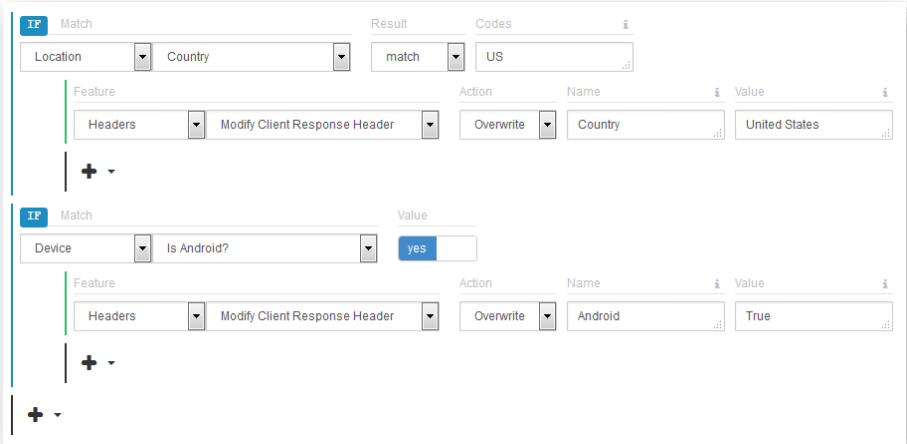
Before one or more custom actions may be applied to CDN traffic, the type of traffic that will be affected must be defined within a rule. Setting this up involves choosing:

1. The logic through which requests will be identified.
2. The criteria for identifying requests (i.e., match conditions).

Request Matching Logic

The initial step for defining how requests are identified is to choose how match conditions will be interpreted.

Type	Description
Mutually Exclusive	<p>Use this type of section (i.e., Select First Match) when a match should only be found when a request satisfies the first match condition out of a set of competing match conditions.</p> <p>A mutually exclusive section that identifies requests by country is illustrated below. This configuration will apply different actions according to the request's origin (i.e., United States or Canada).</p>  <p>As illustrated below, additional competing match conditions may be added by clicking + ELSE IF.</p> 

Type	Description
Independent	<p>This type of match statement is processed independently from all other statements. As a result, it makes it possible for a request to apply multiple features (i.e., custom actions) based off of various match conditions.</p> <p>Unlike a mutually exclusive match section (aka Select First Match), this type of statement does not have an identifying label.</p> <p>Siblings within an independent statement cannot be set to the same match condition. Set up this type of configuration by using a mutually exclusive statement instead. In the following illustration, notice that the Country match condition is unavailable when configuring the match condition at the bottom of the rule.</p>
	
	<p>A rule containing two independent match conditions is illustrated below. In this scenario, a request that is submitted by an Android device from within the United States will satisfy both match conditions. Therefore, the response will contain both the Country and the Android headers.</p>
	

Mutually Exclusive Match Section (Select First Match)

A Select First Match section treats match conditions defined at its root level as mutually exclusive. The following illustration shows two match conditions at the root of a Select First Match section. Only the features associated with the first successful match will be applied to a request. All subsequent match conditions will be ignored.

The screenshot displays a configuration window titled "SELECT FIRST MATCH". It contains two match conditions stacked vertically. Each condition has a tab labeled "IF" and "Match". The first condition has "Location" and "Country" as match criteria, with "match" as the result and "US" as the code. The second condition has "Location" and "Country" as match criteria, with "match" as the result and "CA" as the code. Both conditions have a "+" icon to expand them. Below the second condition is a "+ ELSE IF" button, and at the bottom is another "+" icon.

Mutually Exclusive Match Section (Select First Match) Example

Independent statements may be added within a mutually exclusive section. This type of setup allows additional features (i.e., custom actions) to be applied to qualifying requests. In the following illustration, notice the following:

- If a request is made from the United States, then a Country response header set to "United States" will be applied to the response.
- Additional features may be applied to request if the request was either made from an Android device or if the referrer is "sales.mydomain.com."
- The last match condition (i.e., Request Method = POST) will be ignored for requests made from the United States, since those requests are handled by the first branch in the Select First Match section.

SELECT FIRST MATCH

IF	Match	Result	Codes
	Location	Country	match US

Feature	Action	Name	Value
Headers	Modify Client Response Header	Overwrite	Country United States

IF	Match	Value
	Device Is Android?	yes

Feature	Action	Name	Value
Headers	Modify Client Response Header	Overwrite	Android True

IF	Match	Result	Value	Ignore Case
	Request Referring Domain Wildcard	match	sales.mydomain.com	yes

Feature	Action	Name	Value
Headers	Modify Client Response Header	Overwrite	Source Sales

ELSE IF	Match	Value
	Request Request Method	POST

Feature	Enabled
Access Deny Access (403)	yes


+ ELSE IF

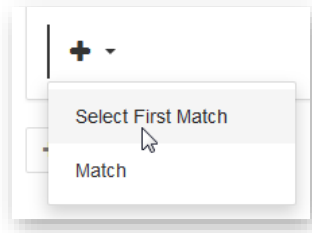
Independent Statements within a Mutually Exclusive Section Example

Key information:

- A "Select First Match" label and a dark blue left border identifies a Select First Match section.
- A rule can only be assigned a single mutually exclusive (aka Select First Match) section.
- Although the set of features that will be applied to a request is limited to the first successful match within a Select First Match section, additional features may be applied to a request as a result of an another rule or an independent match statement located outside of the Select First Match section.

To add a mutually exclusive match statement

1. Create or modify a rule.
2. From the desired position in the rule, click  and then select "Select First Match."



A blank Select First Match section will appear. Make sure to define at least one match condition and feature for both the IF and the ELSE IF statements.

Independent Match Statement

Each independent match statement is processed independently from all other match statements. This allows a request to satisfy multiple match statements.

Note: An independent match statement is any IF statement that is not located at the root of a Select First Match section.


Key information:

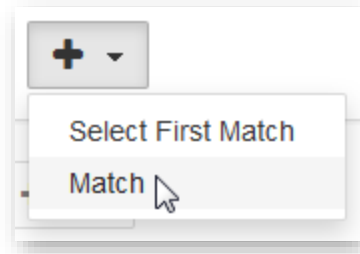
- **Greedy Matching:** A request will try to satisfy as many independent match conditions as it can.

Note: The features associated with all successful match conditions will be applied to the request.

- **Nested Match Conditions:** Greedy matching also applies to nested match conditions. However, the prerequisite for a successful match is that the request must also satisfy the parent match condition.

To add an independent match statement

1. Create or modify a rule.
2. From the desired position in the rule, click  and then select "Match."



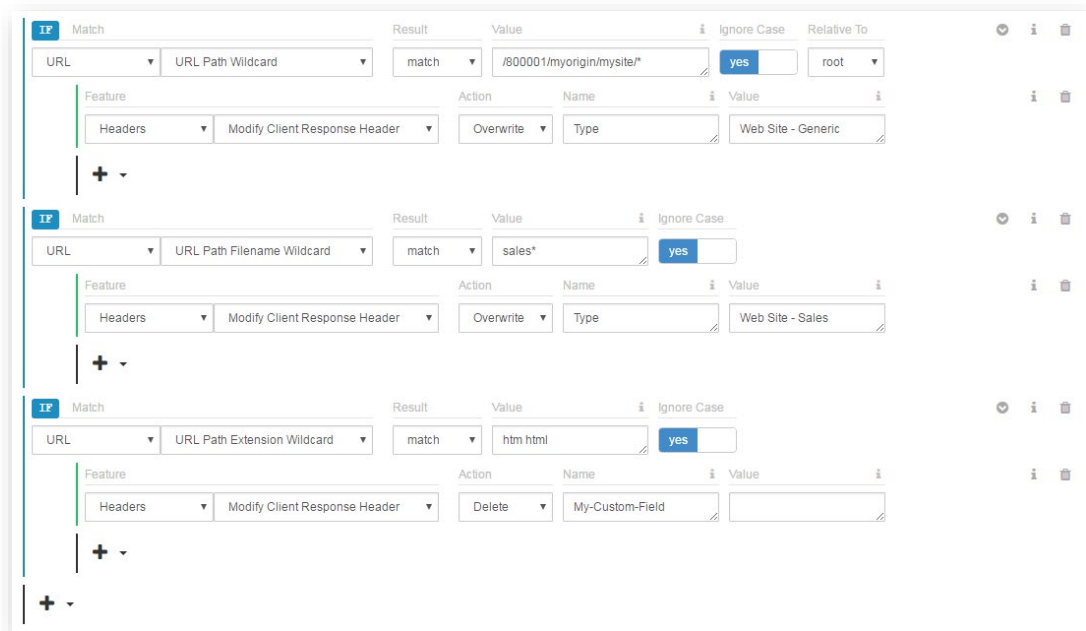
A blank IF statement will appear. Make sure to assign a match condition and at least one feature to it.

Exception

If the same feature will be applied to the request by multiple match conditions, then precedence will be granted to the last match condition.

Example:

The rule illustrated below shows 3 match conditions that define the same feature (i.e., Modify Client Response Header).



If a request satisfies all 3 match conditions, then only the second and third instances will be applied to the request. Specifically, the Type response header will be set to "Web Site - Sales" and the My-Custom-Field response header will be excluded from the response.

Sample Scenarios

Both of the following scenarios illustrate that match conditions are processed separately. The second scenario also illustrates that a nested match condition is processed independently from other nested match conditions.

Scenario #1:

Sibling match conditions allow multiple actions (i.e., features) to be applied to qualified requests. In this scenario, we will examine how actions are applied to requests for the following rule:

IF Match Condition A

Feature x

IF Match Condition B

Feature z

A request will trigger action(s) by satisfying any of the following match conditions:

Match Condition(s)	Resulting Action(s)
A	x
B	z
A + B	x + z

Scenario #2:

Nested match conditions provide more control over when actions (i.e., features) will be applied to requests. In this scenario, we will examine how actions are applied to requests for the following rule:

IF Match Condition A

Feature w

IF Match Condition A1

Feature x

IF Match Condition A2

Feature y

IF Match Condition B

Feature z

A request will trigger action(s) by satisfying any of the following match conditions:

Match Condition(s)	Resulting Action(s)
A	w
A + A1	w + x
A + A2	w + y
B	z

Conditional Expressions

A conditional expression identifies the start of a statement that defines a match condition and one or more features.

Valid conditional expressions are defined below.

Conditional Expression	Description
IF	<p>The meaning of an IF conditional expression varies according to whether it is at the root of a Select First Match statement.</p> <ul style="list-style-type: none">• Select First Match (Root): An IF conditional expression at the root of a Select First Match statement indicates that it is a mutually exclusive statement.• All Other Locations: An IF conditional expression at any other location is treated as an independent match statement. <hr/> <p>Note: An IF statement may be nested under an IF or ELSE IF statement. This type of setup defines an additional match condition that must be met before the set of features associated with that nested match condition may be applied to a request.</p> <hr/>
ELSE IF	<hr/> <p>Note: The ELSE IF conditional expression is only applicable within a Select First Match section.</p> <hr/> <ul style="list-style-type: none">• ELSE IF conditional expressions may only be added at the root level of a Select First Match section.• An ELSE IF conditional expression specifies an alternative match condition that must be met before a set of features may be applied to a request.• The presence of an ELSE IF conditional expression indicates the end of the previous statement.

Match Conditions

The purpose of a match condition is to identify a request. A request may be identified by a variety of criteria (e.g., URL, IP address, request metadata, etc.).

Note: A match condition is defined directly after a conditional expression.

Categories

Setting up a match condition involves selecting a category and then the desired match condition. Each available category is described below.

Category	Description
General	This category contains the "Always" match condition. Use this match condition to define a default set of actions that will be applied to all requests.
Request	This category contains match conditions that identify requests based on its properties (e.g., IP address, cookie, referrer, and header values).
Device	This category contains match conditions that identify requests that originate from a mobile device.
Location	This category contains match conditions that identify requests that originate from a specific network or geographical location as defined by ASN or country.
URL	This category contains match conditions that identify requests based on the request URL.
Origin	This category contains match conditions that identify requests based on the origin server from which content is being requested.

Default Request Handling

One easy way to ensure that all requests to your assets are granted a default behavior is to create a rule that starts with an IF statement set to the Always match and the type of actions that should take place.

Key information:

- Override or supplement the default behavior by:
 - Adding sibling match conditions to the same rule.

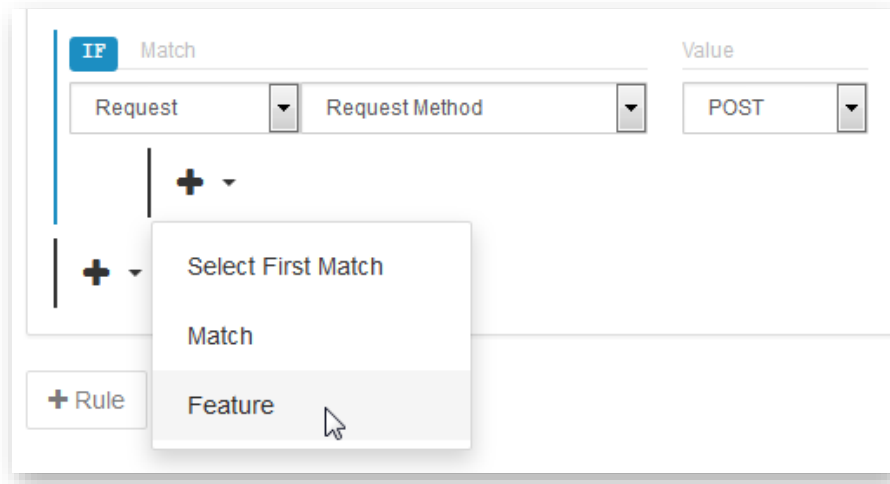
Note: These additional statements cannot be nested under the Always match condition. This type of setup is illogical, since an Always match condition should be applied by default to all requests.

 - Adding another rule.
- **Multiple Rules:** If there are multiple rules, it is recommended that the rule containing the Always match condition be placed at the very top of the list of rules.

Tip: It is not recommended to place a rule containing the Always match below a rule with a conflicting feature. This type of configuration will make the conflicting portion of your other rule irrelevant.

Applying Actions to Requests

The set of actions that will be applied to a particular type of request may be defined by simply adding one or more features to the conditional statement that identifies the desired request type.



Selecting a Feature

Features are organized by category. A brief description for each category is provided below.

Category	Description
General	This category contains the Comment feature which allows notes to be added to a rule.
Caching	<p>This category may be used to define caching, compression, and bandwidth throttling policies.</p> <p>The most common usage for this type of feature is to determine whether content will be cached and for how long.</p>
Logs	<p>This category may be used to define the type of data that is stored in a raw log file.</p> <hr/> <p>Note: This category is intended only for accounts on which raw log file support has been activated and raw log file archival has been enabled. Please contact your CDN account manager to find out more information on raw log file activation.</p> <hr/>
Access	This category may be used to define access control (e.g., deny access to content or secure it via Token-Based Authentication).

Category	Description
Headers	This category may be used to define request and response header behavior (e.g., set or delete header data).
Specialty	This category may be used to define advanced actions through which a custom configuration for a specialized need may be achieved.
Origin	This category may be used to define the instructions provided by an edge server to an origin server.
URL	This category may be used to redirect and rewrite request URLs.
WAF	This category may be used to enable a Web Application Firewall instance.

Rule Precedence

A draft or policy may contain one or more rules. The use of multiple rules facilitates:

- The setup of a default configuration that will be applied to all requests.
- The creation of rules that specialize according to request type or behavior.
- Additional control over how requests for content are handled.

Sample Scenario

In this sample scenario, create the following rules:

Order	Purpose	Description
1	Assign a default cache policy for all requests.	Placing this rule at the top of the list ensures that this cache policy is assigned by default to all requests.
2	Define an alternative cache policy based on origin type.	The rule's position allows it to override the default behavior defined in the first rule for requests to a specific origin.
3	Deny access based on the requester's location.	This rule denies access for requests that originate from a specific location. Although this rule does not contradict the above two rules, segregating these instructions improves readability and facilitates rule management.

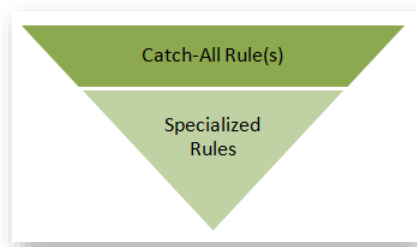
Note: Rule order can drastically affect how requests are handled. In the above example, moving the default cache policy rule below the other rules will nullify the cache policy defined by origin type.

Rule Order

Rules are typically processed in the order that they are listed. If a client's request satisfies the criteria for more than one rule, then the corresponding features will be applied to the request. This could lead to a situation where conflicting actions will take place. In such a case, the last action to take place will take precedence over previous actions. Therefore, it is recommended to place rules that should take precedence as close to the bottom of the list as possible.

Tip: Rules within a draft may be reordered at any time.

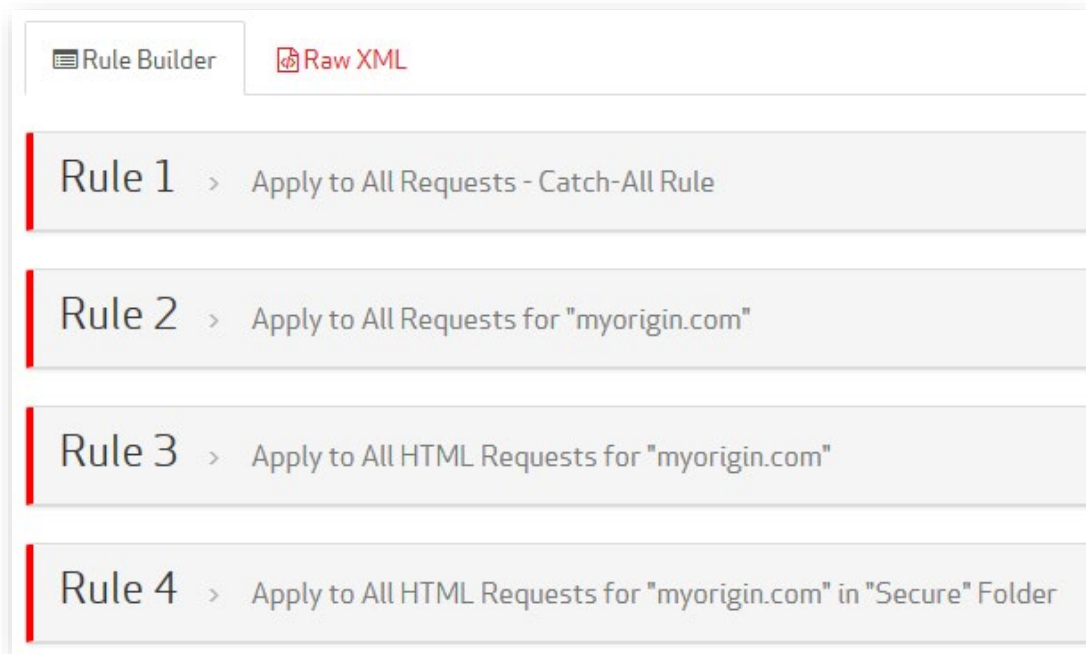
Note: This principle also applies for independent match conditions.



Rule Processing Order

Sample Scenario

The following illustration contains four sample rules that provide different instructions based on the type of request being performed.



Draft containing 4 Sample Rules

We will now examine how a request for an HTML asset from the "Secure" folder on "myorigin.com" will be handled. This request will satisfy the match criteria for all four rules. Therefore, the features associated with each of the rules will be applied to the request. If one or more rules contain conflicting instructions, then the rule closest to the bottom of the list will take precedence. In this sample scenario, the rule called "Apply to All HTML Requests for "myorigin.com" in "Secure" Folder" would take precedence over the other rules.

Tip: A good rule of thumb when determining where a rule should be placed is to order rules according to the level of detail in the criteria. Rules with general criteria should be placed closer to the top of the list, while more detailed criteria should be placed closer to the bottom. This type of configuration allows catch-all rules to assign default handling behavior for your assets without interfering with the manner in which specific types of assets are handled.

Exceptions

The following cases are exceptions to the order-based rule precedence stated above:

1. Identical Top-Level Matching Criteria

If the top-level matching criteria for a rule are the same as that of another rule, then the actions associated with those two rules will take place at the same time. Thus, a rule at the bottom could be combined with a rule at the top of the list. This type of situation would prevent the rule at the bottom from taking precedence over other rules.

2. URL Rewrite Precedence

The URL Rewrite feature takes precedence when multiple features will be applied to a request. This occurs regardless of rule order.

Example

In this sample scenario, a policy contains two rules. The first rule applies the URL Redirect feature, while the second one applies the URL Rewrite feature. If a request satisfies both rules, then the URL Rewrite feature will always be applied to a request before the URL Redirect feature.

3. Token Auth Precedence

The Token Auth feature takes precedence over most features with the exception of the URL Rewrite feature. This occurs regardless of rule order.

Example

In this sample scenario, a policy contains two rules. The first rule applies the URL Redirect feature, while the second one applies the Token Auth feature. If a request satisfies both rules, then the Token Auth feature will always be applied to a request before the URL Redirect feature.

Extending and Overriding CDN Behavior

Rules Engine can be used to override and/or extend the CDN configuration defined in the MCC and the response headers defined by a web server (e.g., Apache or IIS).

- **Web Server (Customer Origin):** A web server can define response headers that will be associated with assets requested through our CDN. Rules Engine can override the values assigned to these response headers.
Additionally, a web server can define a cache policy for the requested content through certain key response headers. Rules Engine can override this cache policy with a custom cache policy for content served through the CDN.
- **Default CDN Response Headers:** If certain key response headers have not been defined by the origin server, then a default response header value may be assigned to the response returned to the user agent and the cached version of the asset. These default response header values can also be overridden by Rules Engine.
- **MCC:** The MCC defines how assets may be accessed through the CDN. Rules Engine may be used to customize the actions that will take place when your content is requested.

For example, a rule may prevent users from a particular country or region from requesting content from a customer origin.

Interaction with Key Features

Rules Engine will not override the configuration defined for the following features:

- Token-Based Authentication
- Country Filtering

The above features secure content by directory. This configuration takes precedence over instructions provided in a rule.

Tip: It is recommended to configure the above features solely within Rules Engine. This type of configuration maximizes flexibility and reduces complexity.

Example:

This example demonstrates how Rules Engine may be used to extend the behavior of Token-Based Authentication.

It assumes the following CDN configuration:

Feature	Configuration
Token-Based Authentication	Secures the following path: /marketing/secure
Rules Engine	<ol style="list-style-type: none">1. Enables Token-Based Authentication for all files.2. Bypasses Token-Based Authentication for HTML files.

Based on this configuration, the CDN will secure content as indicated below.

- All requests for content stored under the /marketing/secure branch will be secured by Token-Based Authentication.
- All requests, with the exception noted below, will be secured by Token-Based Authentication.
- Requests for HTML and HTM files that reside outside of the /marketing/secure branch will not be secured by Token-Based Authentication.

Note: In this scenario, valid tokens are still required for assets that support HTML pages (e.g., JS, CSS, images, etc.) that are not secured by Token-Based Authentication.






Edge Server Compression



Edge server compression should only be configured through either Rules Engine's Compress File Types feature or the **Compression** page.

Important: Attempting to configure edge server compression through both Rules Engine and the **Compression** page will create an invalid configuration that will not be applied to your account. Furthermore, no additional configuration changes will be applied to your account until this conflict is resolved.

Rule Navigation

A navigation bar will appear on the left-hand side of the screen when a draft or policy contains more than 3 rules.

Icon	Name	Description
	Scroll to Top	Scrolls to the first rule in the draft or policy.
	Scroll to Rule Number	Scrolls to a specific rule. To scroll to a specific rule <ol style="list-style-type: none">1. Click on this icon.2. Type the number of the desired rule.3. Hit ENTER.
	Scroll to Bottom	Scrolls to the last rule in the draft or policy.
	Collapse All Rules	Collapse all rules in the draft or policy to only display a header bar. <hr/> Tip: One use for this capability is to make it easier to reorganize rules by dragging and dropping them. <hr/>
	Expand All Rules	Expands all rules in a draft or policy to display the logic on how requests will be identified and the set of actions that will be applied to those requests.

Note: Expand a single rule by clicking on the  that appears directly to the right of the rule number in the header bar (e.g., Rule 1 ).

Administering Drafts and Rules

This section explains how to perform the following administrative tasks:


- Create and modify a draft.
- Convert a draft into a policy.
- Delete a draft.

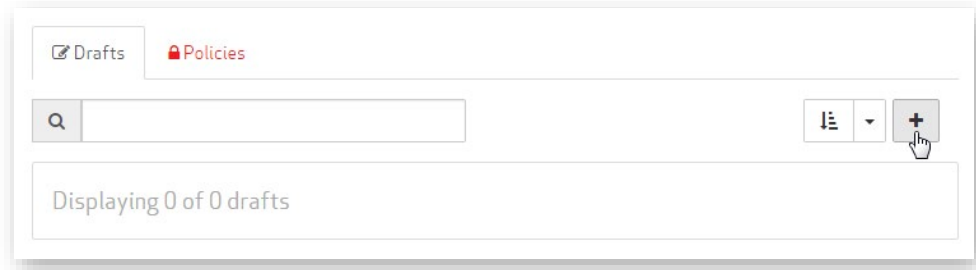
Creating a Draft

Create a draft by either adding a brand new one or by copying an existing draft/policy.

To create a draft

1. Navigate to the **Rules Engine** page corresponding to the desired platform.

2. From the **Drafts** tab, click on the  icon.



3. From the **Create Draft** page, type the name that will be assigned to the new draft.
4. Click **Continue** to save the new draft.
5. Set up a rule and then click **Save** to save it. Repeat this step as needed.

To copy a draft

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. From the **Drafts** tab, click on the name of the desired draft.
3. Click **Duplicate** to create a copy of the draft. The name of the new draft will use this syntax:
Draft - copy

To create a draft from an existing policy

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. Click the **Policies** tab.
3. Click on the name of the desired policy.
4. Click **Duplicate** to create a draft from the policy. The name of the new draft will use this syntax:
Policy - clone from Policy

Modifying a Draft




Modify a draft to fine-tune a Rules Engine configuration.

Tip: Only a policy may be applied to test or live site traffic. This means that once a draft accurately reflects the desired configuration, it should be converted into a read-only policy.

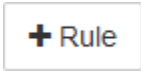
Tip: Use the search bar to filter the drafts listed on the **Drafts** tab by name. Only drafts whose name contain the specified word or phrase will be displayed. Clear the search bar to view all drafts.

Note: All settings within a draft may be modified. However, only a single rule may be modified at any given time. Likewise, a draft may not be renamed until changes to a rule have been saved or canceled.

To modify a draft

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. From the **Drafts** tab, click on the name of the desired draft.
3. Perform one of the following actions:
 - Rename a draft by clicking the  icon next to its name. Type the desired name.

Click the  button to save the new name.
 - Add, copy, move, or delete a rule. Please see the instructions provided below.
 - Modify a rule (e.g., rename it or add/modify/delete an item from it). Please see the instructions provided below.
4. Repeat the previous step as needed.
5. Click **Save** to save your changes.

To add a rule

1. Open the desired draft by clicking on it from the **Drafts** tab of the **Rules Engine** page.
2. Add a rule by clicking .
3. Optional. Type a description for the new rule.
4. Modify the rule as needed. Repeat this step as needed.
5. Click **Save** to save your changes.

Tip: A new rule may be created by copying an existing rule within the current draft.

To copy a rule within a draft

1. Find the desired rule.
2. From the rule's header bar, click the ellipsis (i.e., ...) and then click **Copy Rule**.
3. Determine whether the new rule will be configured exactly like the original rule by selecting one of the following options:
 - **Reset:** All match conditions and features in the copied rule will be reset to their original state.
 - **Preserved:** The configuration associated with matches and features in the copied rule will match the original rule.
4. Click **Copy**. A copy of the rule will appear at the bottom of the list. The term "- copy" will be appended to the name of the copied rule.
5. Modify the copied rule as needed.
6. Click **Save** to add the rule to the draft.

Note: A rule may not be copied while another one is being modified. Save or cancel rule changes before attempting to copy a rule.




To copy a rule to another a draft

1. Find the desired rule.
2. From the rule's header bar, click the ellipsis (i.e., ...) and then click **Transfer Rule**.
3. Navigate to another draft within the same delivery platform.
4. Click **Paste rule from clipboard** which can be found at the bottom of the draft.
The new rule will be appended to the draft and it will be configured exactly like the original rule.
5. Modify the copied rule as needed.
6. Click **Save** to add the rule to the draft.


Note: A rule may not be copied while another one is being modified. Save or cancel rule changes before attempting to copy a rule.

To modify a rule

Reminder: Only a single rule may be modified at any given time.

1. Open the desired draft by clicking on it from the **Drafts** tab of the **Rules Engine** page.
2. Click the  next to the desired rule.
3. Perform any of the following actions:
 - **Rename a rule:** Rename the rule by simply typing a new description in the edit box that appears directly below the rule header bar.
 - **Add a match condition:**
 - i. Find the location in the rule where a new match condition should be inserted.
 - ii. Click  and then select "Match."
 - **First Match Condition:** A rule's first match condition is always added at the top of the rule.
 - **Subsequent Match Conditions:** The new match condition will be added at the same level as the  button.
 - iii. Select a category and a match condition.
 - iv. Configure the match condition's settings.


Note: A match condition cannot be nested under the Always match condition. However, a match condition may be added as a sibling to the Always match condition. This type of setup allows an additional set of actions to be applied to requests that satisfy that match condition.

- **Add a feature:**
 - i. Find the match condition that must be met before the feature will be applied to a request.
 - ii. Click the  button that appears next to it and then select "Feature." A new feature will be added directly below the match condition.
 - iii. Select a category and a feature.
 - iv. Configure the feature's settings.

- **Add a select first match section:**

- i. Find the location in the rule where a new select first match section should be inserted.



- ii. Click  and then select "Select First Match." A new select first match section will be added directly below it. This section will contain an initial match condition.


- iii. Select a category and a match condition.

- iv. Configure the match condition's settings.

- v. Optional. Add a nested match condition or a feature by clicking on the




button that appears directly below it, selecting "Match" or "Feature," and then configuring it.

- vi. Optional. Add an else if match condition by clicking on the  button that appears directly below it. Select the match condition and configure it. Add a feature to it.




Note: Only a single select first match section may be added per rule.

- **Modify a match condition, select first match, and/or feature:** Simply make the desired changes (e.g., select a different match condition or adjust a setting).
- **Delete a match condition, select first match, or feature:** Click on the  that appears directly to the right of it.

Tip: Hovering over the  icon will highlight the section or feature that will be deleted.

4. Click **Save** to save your changes.

To move a rule

1. Find the desired rule.
2. From the rule's header bar, perform one of the following actions:
 - Click  and drag it to the desired position.
 - Click the ellipsis (i.e., ...) and then select **Move to top** to move this rule to the top of the list.
 - Click the ellipsis (i.e., ...) and then select **Move to bottom** to move this rule to the bottom of the list.
 - Click the ellipsis (i.e., ...) and then type the desired position next to the **Move to** option. Click ENTER to move this rule to the specified position.

Note: A draft will be automatically saved upon moving a rule.

Note: Rules may not be reordered while a rule is being modified. Save or cancel rule changes before attempting to move a rule.

To delete a rule

1. Open the desired draft by clicking on it from the **Drafts** tab of the **Rules Engine** page.
2. From the rule's header bar, click the ellipsis (i.e., ...) and then select **Delete Rule**.
3. Click **Delete** to confirm the rule deletion.

Note: A rule with unsaved changes may not be deleted. Please save or cancel your changes and try again.

Converting a Draft into a Read-Only Policy

Before a draft may be applied to the Production or Staging environment, it must be converted into a read-only policy. Before taking this step, please make sure that the rule(s) defined in the draft accurately reflect(s) how traffic should be handled by the CDN.

Tip: A policy may be applied to the Production or Staging environment via a deploy request.

Tip: Use the search bar to filter the drafts listed on the **Drafts** tab by name. Only drafts whose name contain the specified word or phrase will be displayed. Clear the search bar to view all drafts.

Note: The conversion of a draft to a read-only policy is irreversible. However, a draft may be created by duplicating a policy.

To convert a draft into a read-only policy

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. From the **Drafts** tab, click on the name of the desired draft.
3. Click **Lock Draft as Policy**.

Note: Basic validation will be performed upon converting a draft to a policy. A misconfigured rule (e.g., a blank match condition, feature, or setting) will generate an error. Please review and address any issues with the draft.

Deleting a Draft

A draft may be deleted by viewing it and then clicking **Delete**.

Important: The deletion of a draft is permanent. Once a draft has been deleted, it cannot be recovered.

Tip: Use the search bar to filter the drafts listed on the **Drafts** tab by name. Only drafts whose name contain the specified word or phrase will be displayed. Clear the search bar to view all drafts.

Policy

Policies are designed to:

- Provide an easy way to apply a Rules Engine configuration to the Staging or Production environment.

Note: Only a single policy per environment may be active at any given time.

- Allow the creation, storage, and management of multiple versions of a Rules Engine configuration.

Leverage this capability to:

- Prepare rules in advance of an event (e.g., live event, holiday, or maintenance).
- Roll back to a previously deployed policy.
- Troubleshoot why a new configuration behaves in an unexpected manner.

Tip: Use the search bar to filter the policies listed on the **Policies** tab by name. Only policies whose name contain the specified word or phrase will be displayed. Clear the search bar to view all policies.

Note: A policy may not be modified in any way. However, it may be used as a template for a new draft.

To view a policy

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. Make sure that the **Policies** tab is selected.
3. Click on the name of the desired policy.

To archive a policy

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. Make sure that the **Policies** tab is selected.
3. Click on the name of the desired policy.
4. Click **Archive**.

Note: Active policies and those that are pending deployment to an environment cannot be archived.

Note: Archiving a policy will prevent the policy from being listed on the **Policies** tab.

Interpreting XML Data

A draft may be viewed and constructed through the rule builder. Alternatively, a draft or a policy may be viewed in its raw XML format.

Leverage a code comparison tool (e.g., Code Compare) and the raw XML view to quickly discover the changes between different versions of a policy. This may be accomplished by simply copying both versions of the XML to the code comparison tool and then running a comparison.

Sample XML code is provided below.

```
<policy>
  <rules>
    <rule>
      <description>Default Internal and External Max-Age</description>
      <match.always>
        <feature.caching.force-internal-max-age status="200" value="6" units="hours" />
        <feature.caching.external-max-age status="200" value="6" units="hours" />
      </match.always>
    </rule>
    <rule>
      <description>Default Internal and External Max-Age for HTML and HTML
Assets</description>
      <match.url.url-path-extension.wildcard result="match" value="html htm" ignore-
```



```

case="true">
    <feature.caching.force-internal-max-age status="200" value="5" units="minutes"
/>

    <feature.caching.external-max-age status="200" value="5" units="minutes" />
</match.url.url-path-extension.wildcard>
</rule>
<rule>
    <description>Deny Access to Unauthorized Requests for Private
Information</description>
    <match.url.url-path.wildcard result="match" value="/800001/myorigin/private/*"
ignore-case="true" relative-to="root">
    <match.request.referring-domain.wildcard result="nomatch"
value="secure.example.com" ignore-case="true">
    <feature.access.deny-access enabled="true" />
    </match.request.referring-domain.wildcard>
    </match.url.url-path.wildcard>
</rule>
</rules>
</policy>

```

Each XML tag is described below.

Tip: XML tags will always comply with the hierarchy defined below.

XML Tag	Description
policy	This tag encloses all other tags defined in a policy or a draft.
rules	This tag encloses all rules.
rule	This tag contains all of the tags associated with an individual rule. <hr/> Note: A start and end <rule> tag is generated for each rule in a draft/policy.
description	Enclosed within this tag is an optional description for the current rule.

XML Tag	Description
<code>match.Category.Name</code> <code>result='Comparison' value='Value'</code> <code>Setting(s)</code>	<p>This tag identifies a match condition. The parameters defined for this match condition define its configuration.</p> <hr/> <p>Note: The tag for the Always match condition does not following this naming convention. It is simply <code><match.always></code>.</p> <hr/> <p>This tag may enclose:</p> <ul style="list-style-type: none"> • Match Conditions: A nested match condition identifies an AND IF relationship between two match conditions. Both match conditions must be met before the feature associated with the second match condition may be applied to a request. • Features: Identifies an action (i.e., feature) that will be applied to a request when it satisfies a parent match condition. <hr/> <p>Note: A start and end <code><match></code> tag is generated for each match condition in a rule.</p> <hr/>
<code>feature.Category.Name</code> <code>Setting(s)</code>	<p>This tag identifies a feature that will be applied to a request when it satisfies the parent match condition.</p> <hr/> <p>Note: This tag cannot contain another element. As a result, the closing tag is always specified (e.g., <code><feature.caching.bypass-cache enabled='true'/></code>).</p> <hr/>

XML Tag	Description
select.first-match	<p>This tag identifies the start and end of a select first match section. This type of tag may enclose match conditions.</p> <hr/> <p>Note: Features may be nested under a match condition defined in a select first match section.</p> <hr/> <p>The match conditions defined in this type of section behave differently from those defined outside of it.</p> <p>Differences:</p> <ul style="list-style-type: none"> • Match conditions at the same level are treated as ELSE IF instead of OR conditions. • No additional processing will be performed within a select first match section once a request satisfies a match condition.

Deploy Request

The purpose of a deploy request is to indicate that a policy is ready to be applied to either the Staging or Production environment. Once a deploy request has been submitted, it will undergo an automated validation and error detection system. If no issues are detected, then the policy will be applied to the requested environment.

Note: Although most rules may undergo automated validation, more complex rules require manual review. This manual review process may take up to 4 hours.

Note: It may take up to an hour before a policy is fully deployed to the Production environment, while deployment to the Staging environment should only take approximately 15 minutes.

Deploy Request Submission

The submission of a deploy request may be initiated when viewing a policy. This submission process requires the selection of a target environment (i.e., Staging or Production) and a description of the deploy request.

Note: Please try to assign a relevant description when submitting a deploy request. This information may prove to be helpful when reviewing deploy request history.

Note: Two simultaneous deploy requests to the same environment may not be submitted. Please wait until the previous deploy request has been deployed, rejected, or canceled before submitting another deploy request.

Upon selecting the desired environment, the XML differences between the new policy and the one currently deployed to that environment will be displayed. Review the XML changes to validate that the policy accurately reflects the configuration that should be deployed to that environment.

The following table describes how XML changes are annotated.

Change Type	Line Color	Description
New Line (Addition)	Green	Indicates that the new policy contains a line that is not present in the policy currently deployed to the requested environment.
Deleted Line (Deletion)	Red	Indicates that the new policy does not contain a line that is present in the policy currently deployed to the requested environment.
No Change	White	Indicates that the line is present in both policies and no changes were detected.

A change in a match condition or feature's setting is reported as:

- A deletion of the line that defines the current value for the match condition/feature's setting.
- The addition of the line that proposes a new value for the match condition/feature's setting.

The following illustration demonstrates a change in the Compress File Types feature's media type from "text/html" to "text/htm."

16	16		<match.always>
17	-		<feature.caching.compress-file-types media-types="text/html" />
18	+		<feature.caching.compress-file-types media-types="text/htm" />
19	19		</match.always>

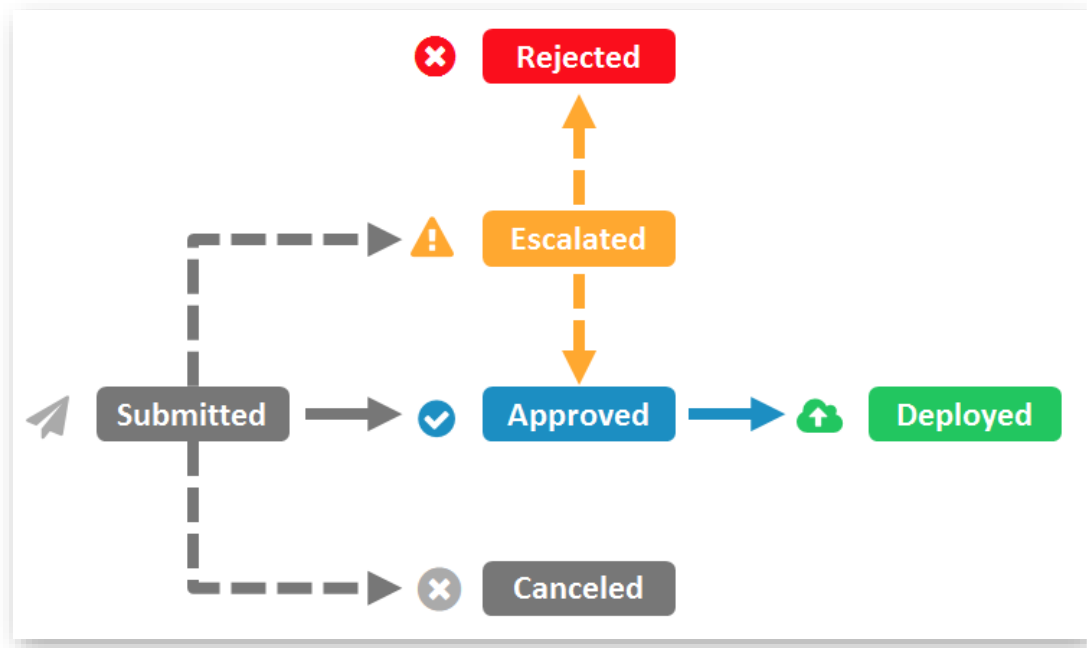
XML Differences

To submit a deploy request

1. Navigate to the **Rules Engine** page corresponding to the desired platform.
2. Click the **Policies** tab.
3. View the desired policy by clicking on its name.
4. Click **Deploy Request**.
5. In the **Environment** option, choose the target environment (i.e., Staging or Production).
The page will be updated to display the differences between the new policy and the policy currently deployed to the selected environment.
6. In the **Message** option, provide a comment that indicates why this deploy request was submitted.
7. Click **Create Deploy Request**.

Deploy Request States

A deploy request must undergo a review and approval process before a policy may be applied to the Staging or Production environment. The workflow for core deploy request states is depicted below.



Deploy Request Status Workflow

A description is provided for each deploy request state below.

State	Description
Submitted	Indicates that a deploy request was submitted by a user.
Canceled	Indicates that the deploy request was canceled by a user. Note: This state will also be applied to a deploy request upon multiple unsuccessful retry attempts.
Pending Review	Indicates that the deploy request is awaiting manual review. Manual review is required when the logic in a deploy request could not be automatically approved by our automated policy review system.
Escalated	Indicates that the deploy request was escalated after manual review. Note: This state only applies to deploy requests for complex policies. As a result, most deploy requests will bypass this state.

State	Description
Rejected	<p>Indicates that the deploy request was rejected and the corresponding policy will not be applied to the target environment.</p> <hr/> <p>Note: A message will indicate the reason why the deploy request was rejected. Duplicate the corresponding policy and then adjust the configuration accordingly.</p> <hr/>
Verification Delayed	<p>Indicates that the deploy request contains logic that is indicative of an invalid configuration.</p> <hr/> <p>Note: Although you can resubmit the deploy request by viewing it and then clicking Retry Deploy Request, the recommended procedure is to cancel the deploy request, duplicate the policy, review/revise all rules, and submit another deploy request.</p> <hr/>
Deployment Delayed	<p>Indicates that the deploy request contains logic that is indicative of an invalid configuration.</p> <hr/> <p>Note: Although you can resubmit the deploy request by viewing it and then clicking Retry Deploy Request, the recommended procedure is to cancel the deploy request, duplicate the policy, review/revise all rules, and submit another deploy request.</p> <hr/>
Approved	Indicates that the deploy request has been approved for deployment.
Deployed	<p>This state is applicable while a policy is being deployed to either the Staging or the Production environment.</p> <hr/> <p>Note: It may take up to 1 hour before a policy is fully deployed to the Production environment, while deployment to the Staging environment should only take approximately 15 minutes.</p> <hr/>

Tip: The last status assigned to each deploy request is indicated on the **Deploy Requests** section of the **Rules Engine** page. Clicking on the desired deploy request will display its historical status activity.

Tip: The policy currently deployed to both Production and Staging environments are indicated in the Production and Staging sections of the **Rules Engine** page. Additionally, historical status activity for the corresponding deploy request may be viewed by following the "View deploy request" link that appears below the policy name.

Deploy Request History

Deploy request history is displayed under the **Deploy Requests** section of the **Rules Engine** page.

Click on a deploy request to view:

- The XML differences between the corresponding policy and the policy that it replaced in the target environment (i.e., Production or Staging).
- A history of the deploy request's status activity. This activity information tracks the deploy request through the status workflow described above.

Deploy Request State Notifications

An email notification may be sent whenever a deploy request enters a new state.

Key information:

- Email notifications are applicable to all deploy requests. However, notifications will only be sent for the states selected on the **Deploy Request Notification** page.
- An email notification may be sent to one or more email addresses. Use a comma to delimit each email address.

Sample configuration:

joe@mycompany.com,mary@mycompany.com,james@widgets.com

- The following syntax will be applied to the subject line for each email notification:
Rules Engine Notification - Platform - Deploy Request ID- State
- The body of the email notification will describe the deploy request.

To set up deploy request state notifications

1. Navigate to the **Deploy Request Notification** page by clicking on the "Notification is *[Enabled/Disabled]*" link from the Rules Engine landing page.
2. Enable notifications by toggling the **Notify me by email when any pending Deploy Request gets updated** option from "off" to "on."
3. Verify that each state for which a notification should be sent is marked.
4. Specify one or more email address(es) in the **Please enter 1 or more email addresses for notifications** option.
5. Click **Save**.

Environment

An environment identifies a CDN network that has a specialized purpose. Rules Engine allows a deploy request to be submitted to either of the following environments:

- **Production:** Identifies the CDN network that serves live traffic.
- **Staging:** Identifies the CDN network that acts as a sandbox for testing new configurations. The policies deployed to this environment will not affect live traffic.

Note: Only a single policy per environment may be active at any given time.

Production

The Production environment identifies the traditional CDN network that is responsible for a live site's content delivery. All CDN settings (e.g., customer origin and edge CNAMEs), with the exception of deploy requests to the Staging environment, are applied to this environment.

Note: Purge requests are applied to both the Staging and Production environments.

Deploy Request Best Practice

The recommended workflow for pushing a new Rules Engine configuration to the Production environment is described below.

1. Submit a deploy request to the Staging environment.
2. Once it is deployed, test and validate the new configuration with mock site traffic.
3. Fine-tune the configuration by duplicating the policy, making the desired changes to the draft, and then repeating steps 1 and 2.
4. Submit a deploy request to the Production environment.

Staging

The Staging environment provides a sandbox through which end-to-end tests of new CDN configurations may be performed without impacting live site traffic. This means that it allows mock site traffic to flow through our Staging network to an origin server. This environment emulates the architecture of the Production environment on a much smaller scale. It also replicates your CDN configuration (e.g., customer origins, edge CNAMEs, Token-based Authentication, etc.).

This environment allows:

- The validation of a new configuration before applying it to live site traffic.
- Experimentation with match conditions and features.

Key information:

- The Staging environment is designed for functional testing and is of a much smaller scale than the Production CDN environment. Therefore, this environment should not be used to perform scale, volume, or throughput testing.
Please keep traffic below the following levels:
 - 50 Mbps
 - 500 requests per second
- Changes to your configuration in this environment do not affect live site traffic.
- Standard usage charges apply to the Staging environment.
- Testing HTTPS traffic via the Staging environment will result in a TLS certificate mismatch.
- With regards to firewall configuration, the Staging environment does not require additional whitelisting. The IP blocks defined on the **Customer Origin** page include our staging POPs.
- With regards to analytics, traffic generated for the Staging environment will be treated as standard traffic. Our reports will not make a distinction between production and staging traffic. However, it may be possible to view usage reports for the Staging environment by generating a report by:
 - **Edge CNAME:** An edge CNAME configuration may be created specifically for the Staging environment. Track staging traffic for this edge CNAME by enabling custom reports and then generating an Edge CNAMEs report.
 - **POP:** The Bandwidth and Data Transferred reports allow report data to be filtered by POP. Use this capability to filter report data for traffic served through our staging POP. A staging POP may be identified via the CSTG label.
- Purge requests are applied to both the Staging and Production environments.
- **Origin Shield:** A staging POP will act as an Origin Shield for customer origins that are protected via Origin Shield.
- **ADN:** A staging POP will act as the primary ADN gateway location for all traffic directed to customer origins.
- **HTTPS Traffic:** HTTPS traffic will generate TLS certificate name mismatch messages on the Staging environment. This message indicates that the hostname defined in the SSL certificate does not match the one defined in the request URL (e.g., staging.wpc.0001.edgecastcdn.net). This message is not indicative of a configuration issue on the Staging environment and it may be safely ignored.

Using the Staging Environment

Requests may be directed to the Staging environment by prepending "staging." to the hostname defined in a CDN URL.

Sample CDN URL (Production):

`http://wpc.0001.edgecastcdn.net/800001/myorigin/marketing/brochure.pdf`

Sample CDN URL (Staging):

`http://staging.wpc.0001.edgecastcdn.net/800001/myorigin/marketing/brochure.pdf`

Edge CNAME URLs

Alternatively, traffic may be directed to the Staging environment via an edge CNAME URL.

This requires performing the following steps:

1. Create an edge CNAME configuration.
2. From your DNS service provider, create a CNAME record that points the above edge CNAME to:

`staging.CDNHostname`

Example: `staging.wpc.0001.edgecastcdn.net`

Match Conditions and Features

Overview

This section provides detailed information on match conditions and features.

Syntax

A value can be defined for certain match conditions and features. This value may consist of:

- Alphanumeric characters
- Special characters (e.g., \, %, *, etc.)

The manner in which special characters will be treated varies according to how a match condition or feature handles text values. A match condition or feature may interpret text in one of the following ways:

- Literal Values
- Wildcard Values
- Regular Expressions

Literal Values

Text that is interpreted as a literal value will treat all special characters, with the exception of the % symbol, as a part of the value that must be matched. In other words, a literal match condition set to "\"'*\\"" will only be satisfied when that exact value (i.e., \\'*\') is found.

Note: A percentage symbol is used to indicate URL encoding (e.g., %20).

Note: Literal match conditions encompass all match conditions that do not support wildcards or regular expressions.

Wildcard Values (Special Characters)

Text that is interpreted as a wildcard value will assign additional meaning to special characters. The following table describes how the following set of characters will be interpreted.

Character	Description
\	<p>A backslash escapes any of the characters defined within this table. A backslash must be specified directly before the special character that should be escaped.</p> <p>For example, the following syntax escapes an asterisk:</p> <ul style="list-style-type: none">• *
%	<p>A percentage symbol is used to indicate URL encoding (e.g., %20).</p>
*	<p>An asterisk is a wildcard that represents zero or more characters.</p>
Space	<p>A space character indicates that a match condition may be satisfied by either of the specified values or patterns.</p>
'value'	<p>A single quote does not have special meaning. However, a set of single quotes is used to indicate that a value should be treated as a literal value. A literal value serves the following purposes:</p> <ul style="list-style-type: none">• It allows a match condition to be satisfied whenever the specified value matches any portion of the comparison value. <p>For example, 'ma' would match any of the following strings:</p> <ul style="list-style-type: none">▪ /business/marathon/asset.htm▪ map.gif▪ /business/template.map • It allows a special character to be specified as a literal character. For example, you may specify a literal space character by enclosing a space character within a set of single quotes (i.e., ' ' or 'sample literal value'). • It allows a blank value to be specified. Specify a blank value by specifying a set of single quotes (i.e., ""). <hr/> <p>Important: If the specified value does not contain a wildcard, then it will automatically be considered a literal value. This means that it is not necessary to specify a set of single quotes.</p> <p>Note: If a backslash does not escape another character in this table, then it will be ignored when specified within a set of single quotes.</p> <p>Note: Another way to specify a special character as a literal character is to escape it using a backslash (i.e., \).</p> <hr/>

Regular Expressions

Regular expressions define a pattern that will be searched for within a text value. Regular expression notation defines specific meanings to a variety of symbols. The following table indicates how special characters are treated by match conditions and features that support regular expressions.

Note: Match conditions and features that support regular expressions are only available with Rules Engine - Advanced Rules. For more information, please contact your CDN account manager.

Note: This section solely highlights how regular expression match conditions and features handle "special characters." The information provided here is not meant to be a comprehensive guide on regular expression usage or syntax.

Special Character	Description
\	<p>A backslash in a regular expression typically:</p> <ul style="list-style-type: none">• Defines a shorthand character class (e.g., \d instead of [0-9]).• Escapes the character that follows it. This causes that character to be treated as a literal value instead of taking on its regular expression meaning. <p>For example, the following syntax escapes an asterisk:</p> <ul style="list-style-type: none">• *
%	<p>The meaning of a percentage symbol depends on its usage.</p> <ul style="list-style-type: none">• %{HTTPVariable}: This syntax identifies an HTTP variable.• %{HTTPVariable}%Pattern: This syntax uses a percentage symbol to identify an HTTP variable and as a delimiter.• \%: Escaping a percentage symbol allows it to be used as a literal value or to indicate URL encoding (e.g., \%20).
*	<p>An asterisk allows the preceding character to be matched zero or more times.</p>
Space	<p>A space character is typically treated as a literal character.</p>
'value'	<p>Single quotes are treated as literal characters. A set of single quotes does not have special meaning.</p>

Match Conditions

A match condition identifies specific types of requests for which a set of features will be performed.

Sample use cases:

- Apply a custom cache policy to requests for content at a particular location.
- Filter requests that originate from a particular:
 - IP address.
 - Country.
- Filter requests that contain a header with the desired value.

Types of Match Conditions

The available types of match conditions are:

- Always
- Device
- Location
- Origin
- Request
- URL

Always

The Always match condition is designed to apply a default set of features to all requests.

Device

These match conditions are designed to identify requests based on the client's user agent.

Name	Purpose
Brand Name	Identifies requests by whether the device's brand name matches a: <ul style="list-style-type: none">• Specific value (Brand Name Literal)• Regular expression (Brand Name Regex)• Specific pattern (Brand Name Wildcard)

Name	Purpose
Device OS	Identifies requests by whether the device's OS matches a: <ul style="list-style-type: none"> • Specific value (Device OS Literal) • Regular expression (Device OS Regex) • Specific pattern (Device OS Wildcard)
Device OS Version	Identifies requests by whether the device's OS version matches a: <ul style="list-style-type: none"> • Specific value (Device OS Version Literal) • Regular expression (Device OS Version Regex) • Specific pattern (Device OS Version Wildcard)
Dual Orientation?	Identifies requests by whether the device supports dual orientation.
HTML Preferred DTD	Identifies requests by whether the device's HTML preferred DTD matches a: <ul style="list-style-type: none"> • Specific value (HTML Preferred DTD Literal) • Regular expression (HTML Preferred DTD Regex) • Specific pattern (HTML Preferred DTD Wildcard)
Image Inlining?	Identifies requests by whether the device supports Base64 encoded images.
Is Android?	Identifies requests by whether the device uses the Android OS.
Is iOS?	Identifies requests by whether the device uses iOS.
Is Smartphone?	Identifies requests by whether the device is a smartphone.
Is Smart TV?	Identifies requests by whether the device is a smart TV.
Is Tablet?	Identifies requests by whether the device is a tablet.
Is Wireless Device?	Identifies requests by whether the device is wireless.
Marketing Name	Identifies requests by whether the device's marketing name matches a: <ul style="list-style-type: none"> • Specific value (Marketing Name Literal) • Regular expression (Marketing Name Regex) • Specific pattern (Marketing Name Wildcard)
Mobile Browser	Identifies requests by whether the device's browser matches a: <ul style="list-style-type: none"> • Specific value (Mobile Browser Literal) • Regular expression (Mobile Browser Regex) • Specific pattern (Mobile Browser Wildcard)

Name	Purpose
Mobile Browser Version	Identifies requests by whether the device's browser version matches a: <ul style="list-style-type: none"> • Specific value (Mobile Browser Version Literal) • Regular expression (Mobile Browser Version Regex) • Specific pattern (Mobile Browser Version Wildcard)
Model Name	Identifies requests by whether the device's model name matches a: <ul style="list-style-type: none"> • Specific value (Model Name Literal) • Regular expression (Model Name Regex) • Specific pattern (Model Name Wildcard)
Progressive Download?	Identifies requests by whether the device supports progressive download.
Release Date	Identifies requests by whether the device's release date matches a: <ul style="list-style-type: none"> • Specific value (Release Date Literal) • Regular expression (Release Date Regex) • Specific pattern (Release Date Wildcard)
Resolution Height	Identifies requests by the device's height.
Resolution Width	Identifies requests by the device's width.

Location

These match conditions are designed to identify requests based on the requester's location.

Name	Purpose
AS Number	Identifies requests that originate from a particular network.
Country	Identifies requests that originate from the specified countries.

Origin

These match conditions are designed to identify requests that point to CDN storage or a customer origin server.

Name	Purpose
CDN Origin	Identifies requests for content stored on CDN storage.
Customer Origin	Identifies requests for content stored on a specific customer origin server.

Request

These match conditions are designed to identify requests based on their properties.

Name	Purpose
Client IP Address	Identifies requests that originate from a particular IP address.
Cookie Parameter	Identifies a request by whether it contains a cookie that matches a: <ul style="list-style-type: none">• Specific value (Cookie Parameter Literal)• Regular expression (Cookie Parameter Regex)• Specific pattern (Cookie Parameter Wildcard)
Edge CNAME	Identifies requests that point to a specific edge CNAME.
Referring Domain	Identifies a request by whether it was referred by a hostname that matches a: <ul style="list-style-type: none">• Specific value (Referring Domain Literal)• Specific pattern (Referring Domain Wildcard)
Request Header	Identifies a request by whether it contains a header that matches a: <ul style="list-style-type: none">• Specific value (Request Header Literal)• Regular expression (Request Header Regex)• Specific pattern (Request Header Wildcard)
Request Method	Identifies requests by their HTTP method.
Request Scheme	Identifies requests by their HTTP protocol.

URL

These match conditions are designed to identify requests based on their URLs.

Name	Purpose
URL Path	Identifies requests by whether their relative path, including filename, matches a: <ul style="list-style-type: none">• Specific value (URL Path Literal)• Regular expression (URL Path Regex)• Specific pattern (URL Path Wildcard)
URL Path Directory	Identifies requests by whether their relative path matches a: <ul style="list-style-type: none">• Specific value (URL Path Directory Literal)• Specific pattern (URL Path Directory Wildcard)
URL Path Extension	Identifies requests by whether their file extension matches a: <ul style="list-style-type: none">• Specific value (URL Path Extension Literal)• Specific pattern (URL Path Extension Wildcard)
URL Path Filename	Identifies requests by whether their filename matches a: <ul style="list-style-type: none">• Specific value (URL Path Filename Literal)• Specific pattern (URL Path Filename Wildcard)
URL Query	Identifies requests by whether their query string matches a: <ul style="list-style-type: none">• Specific value (URL Query Literal)• Regular expression (URL Query Regex)• Specific pattern (URL Query Wildcard)
URL Query Parameter	Identifies requests by whether they contain a query string parameter set to a value that matches a: <ul style="list-style-type: none">• Specific value (URL Query Parameter Literal)• Specific pattern (URL Query Parameter Wildcard)

Always

Category: General

Purpose: Use this type of match condition to define a default set of features that may be applied to all requests.

Note: An Always match condition is designed to define the default CDN behavior for all requests. Therefore, a rule cannot combine this match condition with another one.

Device

Match conditions in the Device category identify requests that originate from a mobile device.

Note: All match conditions in this category require Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Rules Engine can leverage a client's user agent to identify requests that originate from a mobile device. It is able to do so by looking up the client's user agent in the WURFL database. This database contains extensive descriptive information on user agents. This information can be accessed either through match conditions or variables.

Feature Compatibility

Due to the manner in which cache settings are tracked, all Device match conditions are incompatible with the following features:

- Complete Cache Fill (End-of-Life)
- Default Internal Max-Age
- Force Internal Max-Age
- Ignore Origin No-Cache
- Internal Max-Stale

Device Match Conditions

These match conditions are designed to identify requests based on the client's user agent.

Name	Purpose
Brand Name	Identifies requests by whether the device's brand name matches a: <ul style="list-style-type: none">• Specific value (Brand Name Literal)• Regular expression (Brand Name Regex)• Specific pattern (Brand Name Wildcard)
Device OS	Identifies requests by whether the device's OS matches a: <ul style="list-style-type: none">• Specific value (Device OS Literal)• Regular expression (Device OS Regex)• Specific pattern (Device OS Wildcard)
Device OS Version	Identifies requests by whether the device's OS version matches a: <ul style="list-style-type: none">• Specific value (Device OS Version Literal)• Regular expression (Device OS Version Regex)• Specific pattern (Device OS Version Wildcard)
Dual Orientation?	Identifies requests by whether the device supports dual orientation.
HTML Preferred DTD	Identifies requests by whether the device's HTML preferred DTD matches a: <ul style="list-style-type: none">• Specific value (HTML Preferred DTD Literal)• Regular expression (HTML Preferred DTD Regex)• Specific pattern (HTML Preferred DTD Wildcard)
Image Inlining?	Identifies requests by whether the device supports Base64 encoded images.
Is Android?	Identifies requests by whether the device uses the Android OS.
Is iOS?	Identifies requests by whether the device uses iOS.
Is Smartphone?	Identifies requests by whether the device is a smartphone.
Is Smart TV?	Identifies requests by whether the device is a smart TV.
Is Tablet?	Identifies requests by whether the device is a tablet.
Is Wireless Device?	Identifies requests by whether the device is wireless.

Name	Purpose
Marketing Name	Identifies requests by whether the device's marketing name matches a: <ul style="list-style-type: none"> • Specific value (Marketing Name Literal) • Regular expression (Marketing Name Regex) • Specific pattern (Marketing Name Wildcard)
Mobile Browser	Identifies requests by whether the device's browser matches a: <ul style="list-style-type: none"> • Specific value (Mobile Browser Literal) • Regular expression (Mobile Browser Regex) • Specific pattern (Mobile Browser Wildcard)
Mobile Browser Version	Identifies requests by whether the device's browser version matches a: <ul style="list-style-type: none"> • Specific value (Mobile Browser Version Literal) • Regular expression (Mobile Browser Version Regex) • Specific pattern (Mobile Browser Version Wildcard)
Model Name	Identifies requests by whether the device's model name matches a: <ul style="list-style-type: none"> • Specific value (Model Name Literal) • Regular expression (Model Name Regex) • Specific pattern (Model Name Wildcard)
Progressive Download?	Identifies requests by whether the device supports progressive download.
Release Date	Identifies requests by whether the device's release date matches a: <ul style="list-style-type: none"> • Specific value (Release Date Literal) • Regular expression (Release Date Regex) • Specific pattern (Release Date Wildcard)
Resolution Height	Identifies requests by the device's height.
Resolution Width	Identifies requests by the device's width.

Brand Name Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the manufacturer (e.g., Samsung) of the device that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's manufacturer name is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Brand Name Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the manufacturer (e.g., Samsung) of the device that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a ".*" regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's manufacturer name is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Brand Name Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the manufacturer (e.g., Samsung) of the device that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's manufacturer name is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system (e.g., IOS) of the device that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's OS is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system (e.g., IOS) of the device that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's OS is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system (e.g., IOS) of the device that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's OS is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Version Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system version (e.g., 1.0.1) of the device that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's OS version is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Version Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system version (e.g., 1.0.1) of the device that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's OS version is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Device OS Version Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system version (e.g., 1.0.1) of the device that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's OS version is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Dual Orientation?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request supports dual orientation (i.e., portrait and landscape).

HTML Preferred DTD Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request has a preferred document type definition (DTD) for HTML content (e.g., html5) that matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's HTML preferred DTD is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

HTML Preferred DTD Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request has a preferred document type definition (DTD) for HTML content (e.g., html5) that matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a ".*" regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's HTML preferred DTD is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

HTML Preferred DTD Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request has a preferred document type definition (DTD) for HTML content (e.g., html5) that matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's HTML preferred DTD is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Image Inlining?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request supports Base64 encoded images.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is Android?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system of the device that issued the request is Android.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is iOS?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the operating system of the device that issued the request is iOS.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is Smartphone?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request is a smartphone.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is Smart TV?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request is a smart TV.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is Tablet?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request is a tablet. This is an OS-independent description.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Is Wireless Device?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request is a wireless device.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Marketing Name Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the marketing name (e.g., BlackBerry 8100 Pearl) of the device that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's marketing name is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Marketing Name Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the marketing name (e.g., BlackBerry 8100 Pearl) of the device that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's marketing name is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Marketing Name Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the marketing name (e.g., BlackBerry 8100 Pearl) of the device that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's marketing name is a "match" or does not match the specified pattern.

- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the name of the browser (e.g., Chrome) that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's browser name is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the name of the browser (e.g., Chrome) that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a ".*" regular expression will match all requests, since the

specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's browser name is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the name of the browser (e.g., Chrome) that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's browser name is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Version Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the version (e.g., 31) of the browser that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's browser version is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Version Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the version (e.g., 31) of the browser that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's browser version is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Mobile Browser Version Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the version (e.g., 31) of the browser that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's browser version is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Model Name Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the model name (e.g., s3) of the device that issued the request matches a literal value.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's model name is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Model Name Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the model name (e.g., s3) of the device that issued the request matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's model name is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Model Name Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the model name (e.g., s3) of the device that issued the request matches a pattern defined by a wildcard value.

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's model name is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Progressive Download?

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request supports the playback of audio/video while it is still being downloaded.

Note: This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Release Date Literal

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request was added to the WURFL database on a date (e.g., 2013_december) that matches a literal value.

Format: *yyyy_mm*

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's WURFL release date is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Release Date Regex

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request was added to the WURFL database on a date (e.g., 2013_december) that matches the pattern defined by a Perl-compatible regular expression.

Format: *yyyy_mm*

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a `"*"` regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the device's WURFL release date is a "match" or does not match the specified regular expression.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Release Date Wildcard

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by whether the device that issued the request was added to the WURFL database on a date (e.g., 2013_december) that matches a pattern defined by a wildcard value.

Format: *yyyy_mm*

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Specify multiple values by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- The **Result** option determines whether this match condition will be met when the device's WURFL release date is a "match" or does not match the specified pattern.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Resolution Height

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by the height, in pixels, of the device that issued the request.

Key information:

- Height must be specified in pixels
- Height must be specified as an integer value.
An integer value only accepts whole numbers. Additionally, a mathematical operator must be specified to determine when this match condition will be met. These mathematical operators are described below.

Operator	Description
<	This match condition is satisfied when the value generated from the request is less than the specified value.
<=	This match condition is satisfied when the value generated from the request is less than or equal to the specified value.
==	This match condition is satisfied when the value generated from the request is equal to the specified value.
>	This match condition is satisfied when the value generated from the request is greater than the specified value.
>=	This match condition is satisfied when the value generated from the request is greater than or equal to the specified value.

- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Resolution Width

Note: This match condition requires Rules Engine - Mobile Device Detection Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Device

Purpose: Identifies requests by the width, in pixels, of the device that issued the request.

Key information:

- Width must be specified in pixels
- Width must be specified as an integer value.
An integer value only accepts whole numbers. Additionally, a mathematical operator must be specified to determine when this match condition will be met. These mathematical operators are described below.

Operator	Description
<	This match condition is satisfied when the value generated from the request is less than the specified value.
<=	This match condition is satisfied when the value generated from the request is less than or equal to the specified value.
==	This match condition is satisfied when the value generated from the request is equal to the specified value.
>	This match condition is satisfied when the value generated from the request is greater than the specified value.
>=	This match condition is satisfied when the value generated from the request is greater than or equal to the specified value.

- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Location

These match conditions are designed to identify requests based on the requester's location.

AS Number

Category: Location

Purpose: Identifies requests by the network from which the request was issued.

A network is identified by its Autonomous System Number (ASN).

Key information:

- Specify multiple AS numbers by delimiting each one with a single space.
Example: A value of "64514 64515" matches requests arriving from either 64514 or 64515.
- Certain requests may not return a valid AS number. A question mark (i.e., ?) will match requests for which a valid AS number could not be determined.
- The entire AS number for the desired network must be specified. Partial values will not be matched.
- This match condition is satisfied when the client's network is either an exact match (i.e., match) or does not match the specified AS number(s).
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

City Name Literal

Category: Location

Purpose: Identifies requests by the city from which the request was issued.

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified value.
- Certain requests may not return a valid city name. A question mark (i.e., ?) will match requests for which a valid city name could not be determined.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from the specified city.
 - **Does Not Match:** This condition is satisfied when the request does not originate from the specified city.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

City Name Regex

Category: Location

Purpose: Identifies requests by whether they originate from a city whose name matches the pattern defined by a Perl-compatible regular expression.

Key information:

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a ".*" regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the specified regular expression.
- The **Result** option determines whether this match condition will be met when the city from which the request originated from is a "match" or does not match the specified regular expression.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Continent

Category: Location

Purpose: Identifies requests by the continent from which the request was issued.

Key information:

- Specify one or more continents using the following codes:
 - **AF:** Africa
 - **AS:** Asia
 - **EU:** Europe
 - **NA:** North America
 - **OC:** Oceania
 - **SA:** South and Central America
 - **?:** Unknown continent
- Specify multiple continents by delimiting each one with a single space.
- Wildcards are not supported when specifying a continent code.
- Certain requests may not return a valid continent code. A question mark (i.e., ?) will match requests for which a valid continent code could not be determined.
- Continent codes are case-sensitive.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified continents.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified continents.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Country

Category: Location

Purpose: Identifies requests by the country from which the request was issued.

Key information:

- A country can be specified through its country code.
- Specify multiple country codes by delimiting each one with a single space.
- Wildcards are not supported when specifying a country code.
- The "EU" and "AP" country codes do not encompass all IP addresses in those regions.
- Certain requests may not return a valid country code. A question mark (i.e., ?) will match requests for which a valid country code could not be determined.
- Country codes are case-sensitive.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified countries.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified countries.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Replicating Country Filtering Behavior

This match condition allows you to perform a multitude of customizations based on the location from which a request originated. For example, the behavior of the Country Filtering feature can be replicated through the following configuration:

- **URL Path Wildcard Match:** Set the URL Path Wildcard match condition to the directory that will be secured.

Tip: Make sure to append an asterisk to the end of the relative path to ensure that access to all of its children will be restricted by this rule.

- **Country Match:** Set the Country match condition to the desired set of countries.
 - **Allow:** Set the Country match condition to "nomatch" to only allow the specified countries access to content stored in the location defined by the URL Path Wildcard match condition.
 - **Block:** Set the Country match condition to "match" to block the specified countries from accessing content stored in the location defined by the URL Path Wildcard match condition.
- **Deny Access (403) Feature:** Enable the Deny Access (403) feature to replicate the allow or block portion of the Country Filtering feature.

Tip: Instead of duplicating the exact behavior of the Country Filtering feature, add other match conditions to fine-tune access control. For example, video playback may be restricted by a combination of request URL, file type, and country.

DMA Code

Category: Location

Purpose: Identifies requests by the metro code (Designated Market Area - DMA) from which the request was issued.

Note: Should I use Metro Code or DMA Code? Both of these match conditions provide the exact same capability. Although we recommend that you should use the Metro Code match condition, either match condition will work.

Key information:

- Specify a metro code as an integer value.
[Request DMA codes from Nielsen.](#)
- Specify multiple metro codes by delimiting each one with a single space.
- Metro codes are only applicable for traffic from the United States.
- Wildcards are not supported when specifying a metro code.
- Certain requests may not return a valid metro code. A question mark (i.e., ?) will match requests for which a valid metro code could not be determined.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified metro codes.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified metro codes.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Latitude

Category: Location

Purpose: Identifies requests by the latitude from which the request was issued.

Key information:

- Latitude is not precise. It returns the geographic coordinate for the postal code, city, region, or country associated with the IP address that submitted the request.
- Specify latitude as a decimal value from 0 to 90. Prepend - for negative values.
Sample value: 33.9705
- Wildcards are not supported when specifying latitude.
- Certain requests may not return a valid latitude. A question mark (i.e., ?) will match requests for which a valid latitude could not be determined.
- The **Compare** option determines when this condition will be met.
 - **==:** This condition is satisfied when the request originates from the specified latitude.
 - **<:** This condition is satisfied when the request originates from a latitude less than the specified value.
 - **<=:** This condition is satisfied when the request originates from a latitude less than or equal to the specified value.
 - **>:** This condition is satisfied when the request originates from a latitude greater than the specified value.
 - **>=:** This condition is satisfied when the request originates from a latitude greater than or equal to the specified value.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Longitude

Category: Location

Purpose: Identifies requests by the longitude from which the request was issued.

Key information:

- Longitude is not precise. It returns the geographic coordinate for the postal code, city, region, or country associated with the IP address that submitted the request.
- Specify longitude as a decimal value from 0 to 180. Prepend - for negative values.
Sample value: 33.9705
- Wildcards are not supported when specifying longitude.
- Certain requests may not return a valid longitude. A question mark (i.e., ?) will match requests for which a valid longitude could not be determined.
- The **Compare** option determines when this condition will be met.
 - **==:** This condition is satisfied when the request originates from the specified longitude.
 - **<:** This condition is satisfied when the request originates from a longitude less than the specified value.
 - **<=:** This condition is satisfied when the request originates from a longitude less than or equal to the specified value.
 - **>:** This condition is satisfied when the request originates from a longitude greater than the specified value.
 - **>=:** This condition is satisfied when the request originates from a longitude greater than or equal to the specified value.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Metro Code

Category: Location

Purpose: Identifies requests by the metro code (Designated Market Area - DMA) from which the request was issued.

Note: Should I use Metro Code or DMA Code? Both of these match conditions provide the exact same capability. Although we recommend that you should use the Metro Code match condition, either match condition will work.

Key information:

- Specify a metro code as an integer value.
[Request DMA codes from Nielsen.](#)
- Specify multiple metro codes by delimiting each one with a single space.
- Metro codes (DMAs) are only applicable for traffic from the United States.
- Wildcards are not supported when specifying a metro code.
- Certain requests may not return a valid metro code. A question mark (i.e., ?) will match requests for which a valid metro code could not be determined.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified metro codes.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified metro codes.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Postal Code

Category: Location

Purpose: Identifies requests by the postal code from which the request was issued.

Key information:

- Specify a postal code as an integer value.
- Specify multiple postal codes by delimiting each one with a single space.
- A comparison will only be performed against the first 3 characters for Canadian postal codes.
- A comparison will only be performed against the first 2 - 4 characters for United Kingdom postal codes.
- Wildcards are not supported when specifying a postal code.
- Certain requests may not return a valid postal code. A question mark (i.e., ?) will match requests for which a valid postal code could not be determined.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified postal codes.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified postal codes.
- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Region Code

Category: Location

Purpose: Identifies requests by the code for the region (e.g., state or province) from which the request was issued.

Key information:

- Identify the desired region by its region code. A region code, which consists of 1 to 3 alphanumeric characters, identifies a subdivision of a country by the region segment of the corresponding ISO 3166-2 code.

[View ISO 3166-2 codes. \(Wikipedia\)](#)

[View ISO 3166-2 codes. \(UNECE\)](#)

Example:

The ISO 3166-2 code for California is US-CA. Therefore, the region code for California is CA.

- Certain regions have two levels of subdivisions. The specified value will be compared against the most specific region code.

Example:

A request that originates from the Devon (aka Devonshire) county of England, which is part of the United Kingdom (UK), has the following subdivisions: GB and DEV. Requests from this county will be matched against DEV.

- Region codes are only unique within a country. In order to prevent false positives, we strongly recommend that you nest this match condition under the Country match condition.

Example:

Requests from the following regions will report the same region code (i.e., SP).

São Paulo, Brazil (BR-SP)

La Spezia, Italy (IT-SP)

Sandy Point, Bahamas (BS-SP)

- Specify multiple region codes by delimiting each one with a single space.
- Wildcards are not supported when specifying a region code.
- Certain requests may not return a valid region code. A question mark (i.e., ?) will match requests for which a valid region code could not be determined.
- The **Result** option determines when this condition will be met.
 - **Match:** This condition is satisfied when the request originates from one of the specified region codes.
 - **Does Not Match:** This condition is satisfied when the request does not originate from one of the specified region codes.

- Due to the manner in which cache settings are tracked, this match conditions is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Origin

These match conditions are designed to identify requests that point to CDN storage or a customer origin server.

CDN Origin

Category: Origin

Purpose: Identifies requests for content stored on CDN storage.

This match condition is met when the request URI leverages the content access point (e.g., /000001) defined in this match condition.

- **CDN URL:** The request URI must contain the selected content access point.
- **Edge CNAME URL:** The corresponding edge CNAME configuration must point to the selected content access point.

Note: The content access point identifies the service that should serve the requested content.

Note: A rule should not contain multiple origin match conditions (i.e., CDN Origin and Customer Origin). This type of configuration would create a match pattern that could never be satisfied. An exception occurs when using else if statements via the "Select First Match" option.

Example

Note: The following sample configuration assumes that this match condition is satisfied when a request matches the specified value.

Setting this match condition to "/000001" will match the following request:

`http://wpc.0001.edgecastcdn.net/000001/marketing/images/event1.jpg`

However, it will not match against the following request for content on CDN Object Storage:

`http://wpc.0001.edgecastcdn.net/300001/marketing/event.m3u8`

Customer Origin

Category: Origin

Purpose: Identifies requests for content stored on a customer origin server.

Key information:

- This match condition will be satisfied regardless of whether content is requested using a CDN or an edge CNAME URL that points to the selected customer origin.
- A customer origin configuration referenced by a rule may not be deleted from the Customer Origin page. Before attempting to delete a customer origin configuration, make sure that the following configurations do not reference it:
 - Customer Origin match condition
 - An edge CNAME configuration.
- A rule should not contain multiple origin match conditions (i.e., CDN Origin and Customer Origin). This type of configuration would create a match pattern that could never be satisfied.

Note: An exception occurs when using else if statements via the "Select First Match" option.

Request

These match conditions are designed to identify requests based on their properties.

Feature Compatibility

Due to the manner in which cache settings are tracked, all Request match conditions are incompatible with the following features:

- Complete Cache Fill (End-of-Life)
- Default Internal Max-Age
- Force Internal Max-Age
- Ignore Origin No-Cache
- Internal Max-Stale

Client IP Address

Category: Request

Purpose: Identifies requests that originate from a particular IP address.

Key information:

- Make sure to use CIDR notation.
 - Specify multiple IP addresses and/or IP address blocks by delimiting each one with a single space.
 - **IPv4 Example:** 1.2.3.4 10.20.30.40 matches any requests arriving from either 1.2.3.4 or 10.20.30.40.
 - **IPv6 Example:** 1:2:3:4:5:6:7:8 10:20:30:40:50:60:70:80 matches any requests arriving from either 1:2:3:4:5:6:7:8 or 10:20:30:40:50:60:70:80.
 - The syntax for an IP address block is the base IP address followed by a forward slash and the prefix size.
 - **IPv4 Example:** 5.5.5.64/26 matches any requests arriving from 5.5.5.64 through 5.5.5.127.
 - **IPv6 Example:** 1:2:3::0/48 matches any requests arriving from 1:2:3:0:0:0:0:0 through 1:2:3:ffff:ffff:ffff:ffff:ffff.
-
- Note:** IPv6 address blocks should not be fully shortened. As shown in the above example, a trailing 0 is required when shortening fields that consist of 0's.
-
- The **Result** option determines whether this condition will be met when a client's IP address is a "match" or does not match the specified IP address(es).
 - This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Cookie Parameter Literal

Category: Request

Purpose: Identifies requests by whether they contain a cookie set to a literal value.

Key information:

- **Cookie name:**
 - Special characters, including an asterisk, are not supported when specifying a cookie name. This means that only exact cookie name matches are eligible for comparison.
 - Only a single cookie name may be specified per instance of this match condition.
 - Cookie name comparisons are case-insensitive.
- **Cookie value:**
 - The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
 - The Ignore Case option determines whether a case-sensitive comparison will be made against the request's cookie value.
- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires a request to contain the specified cookie with a value that matches the value defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified cookie.
 - It contains the specified cookie, but its value does not match the value defined in this match condition.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Cookie Parameter Regex

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Request

Purpose: Identifies requests by whether they contain a cookie whose value matches the pattern defined by a Perl-compatible regular expression.

Key information:

- **Cookie name:**
 - Special characters, including an asterisk, are not supported when specifying a cookie name. This means that only exact cookie name matches are eligible for comparison.
 - Only a single cookie name may be specified per instance of this match condition.
 - Cookie name comparisons are case-insensitive.
- **Cookie value:**
 - A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "."* regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The **Ignore Case** option determines whether a case-sensitive comparison will be made against the request's cookie value.

- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires a request to contain the specified cookie with a value that matches the regular expression defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified cookie.
 - It contains the specified cookie, but its value does not match the regular expression defined in this match condition.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Cookie Parameter Wildcard

Category: Request

Purpose: Identifies requests by whether they contain a cookie whose value matches a pattern defined by a wildcard value.

Key information:

- **Cookie name:**
 - Special characters, including an asterisk, are not supported when specifying a cookie name. This means that only exact cookie name matches are eligible for comparison.
 - Only a single cookie name may be specified per instance of this match condition.
 - Cookie name comparisons are case-insensitive.
- **Cookie value:**
 - The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

 - Specify multiple cookie values by delimiting each one with a single space.
 - The **Ignore Case** option determines whether a case-sensitive comparison will be made against the request's cookie value.

- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires a request to contain the specified cookie with a value that matches at least one of the patterns defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified cookie.
 - It contains the specified cookie, but its value does not match any of the patterns defined in this match condition.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Edge CNAME

Category: Request

Purpose: Identifies requests that point to a specific edge CNAME.

Key information:

- The list of available edge CNAMEs is limited to those that have been configured on the **Edge CNAMEs** page corresponding to the platform on which Rules Engine is being configured.
- Before attempting to delete an edge CNAME configuration, make sure that an Edge CNAME match condition does not reference it. Edge CNAME configurations that have been defined in a rule cannot be deleted from the **Edge CNAMEs** page.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Referring Domain Literal

Category: Request

Purpose: Identifies requests that were referred by a hostname that matches a literal value.

Key information:

- The hostname associated with the referrer through which content was requested determines whether this condition is met.
- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The **Ignore Case** option determines whether a case-sensitive comparison will be performed.
- The **Result** option determines whether this condition will be met when the referring hostname is a "match" or does not match the specified value.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Sample Scenario

A sample scenario is provided below.

The following sample configuration assumes that this match condition is satisfied when a request matches the specified value.

Value	Description
www.domain.com	This value will be satisfied when the request was referred by a URL whose hostname is "www.domain.com" (e.g., http://www.domain.com/index.html).

Referring Domain Wildcard

Category: Request

Purpose: Identifies requests that were referred by a hostname that matches a pattern defined by a wildcard value.

Key information:

- The hostname associated with the referrer through which content was requested determines whether this condition is met.
- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

Note: If the specified value does not contain an asterisk, then it must be an exact match for the referrer's hostname. For example, specifying "mydomain.com" would not match "www.mydomain.com."

- Specify multiple hostnames by delimiting each one with a single space.
- The **Ignore Case** option determines whether a case-sensitive comparison will be performed.
- The **Result** option determines whether this condition will be met when the referring hostname is a "match" or does not match the specified value(s).
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Sample Scenarios

Sample scenarios are provided below.

The following sample configurations assume that this match condition is satisfied when a request matches the specified value/pattern.

Value	Description
www.domain.com	This pattern will be satisfied when the request was referred by a URL whose hostname is "www.domain.com" (e.g., http://www.domain.com/index.html).
*.domain.com	This pattern will be satisfied when the request was referred by one of the following URLs: <ul style="list-style-type: none">• http://www.domain.com/index.html• http://sub.domain.com/web/main/index.html• https://secure.domain.com/index.html

Request Header Literal

Category: Request

Purpose: Identifies requests by whether they contain a request header set to a literal value.

Key information:

- **Header name:**
 - Header name comparisons are case-insensitive.
 - Spaces in the header name should be replaced with "%20."
 - Only request headers whose name is an exact match to the specified value may satisfy this condition.
- **Header value:**
 - The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
 - The case-sensitivity of header value comparisons is determined by the **Ignore Case** option.
 - Spaces in the header value should be replaced with "%20."
- The **Result** option determines the conditions under which this match condition will be satisfied.

- **Match:** Requires the request to contain the specified header and its value must match the one defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified header.
 - It contains the specified header, but its value does not match the one defined in this match condition.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Sample Scenario

A sample scenario is provided below.

Note: The following sample configuration assumes that this match condition is satisfied when a request matches the specified value.

Header Name	Header Value	Description
Referer	http://www.domain.com/	<p>This configuration will match a request with the following request header:</p> <p>Referer: http://www.domain.com/</p> <p>It will not match a request with the following request header:</p> <p>Referer: http://www.domain.com/page.php</p>

Request Header Regex

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Request

Purpose: Identifies requests by whether they contain a request header whose value matches the pattern defined by a Perl-compatible regular expression.

Key information:

- **Header name:**
 - Header name comparisons are case-insensitive.
 - Spaces in the header name should be replaced with "%20."
 - Only request headers whose name is an exact match to the specified value may satisfy this condition.

- **Header value:**

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a ".*" regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The case-sensitivity of header value comparisons is determined by the **Ignore Case** option.
 - Only request headers whose value is an exact match for the specified regular expression may satisfy this condition.
 - Spaces in the header value should be replaced with "%20."
- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires the request to contain the specified header and its value must match the regular expression defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified header.
 - It contains the specified header, but its value does not match the regular expression defined in this match condition.
 - This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Request Header Wildcard

Category: Request

Purpose: Identifies requests by whether they contain a request header whose value matches a pattern defined by a wildcard value.

Key information:

- **Header name:**
 - Header name comparisons are case-insensitive.
 - Spaces in the header name should be replaced with "%20."
 - Only request headers whose name is an exact match to the specified value may satisfy this condition.
- **Header value:**
 - The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

 - The case-sensitivity of header value comparisons is determined by the **Ignore Case** option.
 - Spaces in the header value should be replaced with "%20."
 - Specify multiple values by delimiting each one with a single space.
- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires the request to contain the specified header and its value must match at least one of the patterns defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified header.
 - It contains the specified header, but its value does not match one of the patterns defined in this match condition.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Sample Scenarios

Sample scenarios are provided below.

The following sample configurations assume that this match condition is satisfied when a request matches one of the specified values.

Header Name	Header Value	Description
Referer	http://www.domain.com/	This configuration will match a request with the following request header: Referer: http://www.domain.com/ It will not match a request with the following request header: Referer: http://www.domain.com/page.php
Referer	http://www.domain.com/* http://blog.domain.com/*	These patterns will match requests from any referring URL from either of the following: <ul style="list-style-type: none">• http://www.domain.com/• http://blog.domain.com/
User-Agent	*Mozilla/5.0 *Android 4.0.4*	This pattern will match requests whose user agent contains the following string: Mozilla/5.0 (Linux; U; Android 4.0.4;

Request Method

Category: Request

Purpose: Identifies requests by their HTTP method.

Only assets that are requested using the selected request method will satisfy this condition.

The available HTTP methods are:

- GET
- HEAD
- POST
- OPTIONS
- PUT
- DELETE
- TRACE
- CONNECT

Key information:

- By default, only the GET request method can generate cached content on our network. All other request methods are simply proxied through our network.
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

Request Scheme

Category: Request

Purpose: Identifies requests by their HTTP protocol.

Only assets that are requested using the selected protocol will satisfy this condition.

Key information:

- The available HTTP protocols are:
 - HTTP
 - HTTPS
- This match condition is incompatible with features defined in the **Feature Compatibility** section above.

URL

These match conditions are designed to identify requests based on their URLs.

URL Path Literal

Category: URL

Purpose: Identifies requests by whether the requested URL points to a relative path that matches a literal value. This relative path includes the filename of the requested asset.

Key information:

- The **Relative to** option determines whether the URL comparison will start before or after the content access point.

Note: A content access point identifies a location by server type (e.g., CDN or customer origin) and your customer account number.

The available values for this option are explained below.

- **Root:** Indicates that the URL comparison will start directly after the CDN hostname.
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- **Origin:** Indicates that the URL comparison will start after the content access point (e.g., /000001 or /800001/myorigin).
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- An edge CNAME URL will be rewritten to a CDN URL prior to URL comparison.
Example:
Both of the following URLs point to the same asset and therefore have the same URL path.
CDN URL:
`http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/path/asset.htm`
Edge CNAME URL:
`http://my.domain.com/path/asset.htm`
URL path (Relative to Root):
`/800001/CustomerOrigin/path/asset.htm`
URL path (Relative to Origin):
`/path/asset.htm`
- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.

- For the purpose of satisfying this condition, query strings in the URL are ignored.
- The case-sensitivity of URL comparisons is determined by the **Ignore Case** option.
- Replace spaces in the URL path with "%20".
- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., %E3%81%93).
 - Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.
Example:
 You should include the following characters instead of こんにちは when defining this match condition's value:
 %E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF
 - The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By default, our CDN service then decodes those characters before the comparison for this match condition is performed.
 - Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- This match condition is satisfied when the relative URL path, including filename, is either an exact match (i.e., match) or does not match the specified value.

URL Path Regex

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: URL

Purpose: Identifies requests by whether the requested URL points to a relative path that matches the pattern defined by a Perl-compatible regular expression. This relative path includes the filename of the requested asset.

Key information:

- An edge CNAME URL will be rewritten to a CDN URL prior to URL comparison.

Example:

Both of the following URLs point to the same asset and therefore have the same URL path.

CDN URL:

`http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/path/asset.htm`

Edge CNAME URL:

`http://my.domain.com/path/asset.htm`

URL path:

`/800001/CustomerOrigin/path/asset.htm`

- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a `"*"` regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- For the purpose of satisfying this condition, query strings in the URL are ignored.
- The case-sensitivity of URL comparisons is determined by the **Ignore Case** option.
- Replace spaces in the URL path with `"%20"`.

- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., %E3%81%93).
 - Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.

Example:

You should include the following characters instead of こんにちは when defining this match condition's value:

%E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF
 - The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By default, our CDN service then decodes those characters before the comparison for this match condition is performed.
 - Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- This match condition is satisfied when the relative URL path is either an exact match (i.e., match) or does not match the regular expression.

URL Path Wildcard

Category: URL

Purpose: Identifies requests by whether the requested URL points to a relative path that matches a pattern defined by a wildcard value. This relative path includes the filename of the requested asset.

Key information:

- The **Relative to** option determines whether the URL comparison will start before or after the content access point.

Note: A content access point identifies a location by server type (e.g., CDN or customer origin) and your customer account number.

The available values for this option are explained below.

- **Root:** Indicates that the URL comparison will start directly after the CDN hostname.
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- **Origin:** Indicates that the URL comparison will start after the content access point (e.g., /000001 or /800001/myorigin).
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- An edge CNAME URL will be rewritten to a CDN URL prior to URL comparison.
Example:
Both of the following URLs point to the same asset and therefore have the same URL path.
CDN URL:
`http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/path/asset.htm`
Edge CNAME URL:
`http://my.domain.com/path/asset.htm`
URL path (Relative to Root):
`/800001/CustomerOrigin/path/asset.htm`
URL path (Relative to Origin):
`/path/asset.htm`
- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- For the purpose of satisfying this condition, query strings in the URL are ignored.
- The case-sensitivity of URL comparisons is determined by the Ignore Case option.
- Replace spaces in the URL path with "%20".
- Specify multiple URL paths by delimiting each one with a single space.
Example:
`/marketing/asset.* /sales/*.htm`
- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., %E3%81%93).
 - Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.
Example:
You should include the following characters instead of こんにちは when defining this match condition's value:
`%E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF`
 - The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By default, our CDN service then decodes those characters before the comparison for this match condition is performed.
 - Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- This match condition is satisfied when the relative URL path, including filename, is either an exact match (i.e., match) or does not match one of the specified patterns.

Sample Scenarios

Sample scenarios are provided below.

The following sample configurations assume that this match condition is satisfied when a request matches one of the specified values.

Value	Relative To	Result
*/test.html */test.php	Root or Origin	This pattern will be satisfied by requests for assets named "test.html" or "test.php" in any folder.
/80ABCD/origin/ text/*	Root	This pattern will be satisfied when the requested asset meets the following criteria: <ul style="list-style-type: none">• It must reside on a customer origin called "origin."• The relative path must start with a folder called "text." In other words, the requested asset may either reside in the "text" folder or one of its recursive subfolders.
/css/ */js/*	Root or Origin	This pattern will be satisfied by all CDN or edge CNAME URLs containing a css or js folder.
*.jpg *.gif *.png	Root or Origin	<p>This pattern will be satisfied by all CDN or edge CNAME URLs ending with .jpg, .gif, or .png.</p> <hr/> <p>Tip: An alternative way to specify this pattern is the URL Path Extension Wildcard match condition.</p> <hr/>
/images/* /media/*	Origin	<p>This pattern will be satisfied by CDN or edge CNAME URLs whose relative path starts with an "images" or "media" folder.</p> <p>CDN URL:</p> <p>http://wpc.0001.edgecastcdn.net/800001/myorigin/images/sales/event1.png</p> <p>Sample Edge CNAME URL:</p> <p>http://cdn.mydomain.com/images/sales/event1.png</p>

URL Path Directory Literal

Category: URL

Purpose: Identifies requests by whether the requested URL points to a relative path that matches a literal value. This relative path excludes the filename of the requested asset.

Key information:

- The **Relative to** option determines whether the URL comparison will start before or after the content access point.

Note: A content access point identifies a location by server type (e.g., CDN or customer origin) and your customer account number.

The available values for this option are explained below.

- **Root:** Indicates that the URL comparison will start directly after the CDN hostname.
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- **Origin:** Indicates that the URL comparison will start after the content access point (e.g., /000001 or /800001/myorigin).
Example:
`http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm`
- An edge CNAME URL will be rewritten to a CDN URL prior to URL comparison.
Example:
Both of the following URLs point to the same asset and therefore have the same URL path.
CDN URL:
`http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/path/asset.htm`
Edge CNAME URL:
`http://my.domain.com/path/asset.htm`
URL path (Relative to Root):
`/800001/CustomerOrigin/path/`
URL path (Relative to Origin):
`/path/`
- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- This match condition is satisfied when the relative URL path, excluding file name, is either an exact match (i.e., match) or does not match the specified value.

- A URL comparison ends right before the filename of the requested asset. A trailing forward slash is the last character in this type of path.
- Replace spaces in the URL path with "%20".
- The case-sensitivity of URL comparisons is determined by the **Ignore Case** option.

URL Path Directory Wildcard

Category: URL

Purpose: Identifies requests by whether the requested URL points to a relative path that matches a pattern defined by a wildcard value. This relative path excludes the filename of the requested asset.

Key information:

- The **Relative to** option determines whether the URL comparison will start before or after the content access point.

Note: A content access point identifies a location by server type (e.g., CDN or customer origin) and your customer account number.

The available values for this option are explained below.

- **Root:** Indicates that the URL comparison will start directly after the CDN hostname.
Example:
http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm
- **Origin:** Indicates that the URL comparison will start after the content access point (e.g., /000001 or /800001/myorigin).
Example:
http://wpc.0001.edgecastcdn.net/800001/myorigin/myfolder/index.htm
- An edge CNAME URL will be rewritten to a CDN URL prior to URL comparison.
Example:
Both of the following URLs point to the same asset and therefore have the same URL path.
CDN URL:
http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/path/asset.htm
Edge CNAME URL:
http://my.domain.com/path/asset.htm
URL path (Relative to Root):
/800001/CustomerOrigin/path/
URL path (Relative to Origin):
/path/

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- A URL comparison ends right before the filename of the requested asset. A trailing forward slash is the last character in this type of path.
- Replace spaces in the URL path with "%20".
- Specify multiple URL paths by delimiting each one with a single space.
Example:
/sales/ /marketing/
- The case-sensitivity of URL comparisons is determined by the **Ignore Case** option.
- This match condition is satisfied when the relative URL path, excluding file name, is either an exact match (i.e., match) or does not match one of the specified URL patterns.

Example:

None of the following values are an exact match for the relative path of the above URLs. Therefore, this match condition will not be satisfied when it is configured to match one of the following values:

- /pa
- ath/
- /path

Sample Scenarios

Sample scenarios are provided below.

Note: The following sample configurations assume that this match condition is satisfied when a request matches one of the specified values.

Value	Relative To	Result
*/Location1/ */Location2/	Root or Origin	This pattern will be satisfied when the parent folder for the requested asset is called either "Location1" or "Location2."
/800001/origin/text/*	Root	This pattern will be satisfied when the requested asset meets the following criteria: <ul style="list-style-type: none">• It must reside on a customer origin called "origin."• The relative path must start with a folder called "text." In other words, the requested asset may either reside in the "text" folder or one of its recursive subfolders.
/css/ */js/*	Root or Origin	This pattern will be satisfied by all CDN or edge CNAME URLs containing a css or js folder.
/images/* /media/*	Origin	This pattern will be satisfied by CDN or edge CNAME URLs whose relative path starts with an "images" or "media" folder. CDN URL: http://wpc.0001.edgecastcdn.net/800001/myorigin/images/sales/event1.png Sample Edge CNAME URL: http://cdn.mydomain.com/images/sales/event1.png
css *js* *images* *media*	Root or Origin	This pattern will be satisfied by CDN or edge CNAME URLs that contain one or more of the following values in the relative path: css, js, images, or media. The following sample URL path would be a match for this criteria due to the presence of "js": /jsmith/folder1

URL Path Extension Literal

Category: URL

Purpose: Identifies requests by whether the requested URL points to an asset with a file extension that matches a literal value.

Key information:

- This match condition looks for a URL that ends with a period (.) and the specified file extension. Therefore, make sure that any file extensions specified in the Value option do not contain a leading period.

Correct:

htm

Incorrect:

.htm

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The case-sensitivity of file extension comparisons is determined by the **Ignore Case** option.
- This match condition is satisfied when the requested asset's file extension is either an exact match (i.e., match) or does not match the specified file extension.

Example:

Specifying "htm" will match for "htm" assets, but not "html" assets.

Sample Scenario

A sample scenario is provided below.

Note: The following sample configuration assume that this match condition is satisfied when a request matches the specified value.

Value	Description
asp	This match condition will be satisfied when it finds a URLs that ends with the following value: .asp

URL Path Extension Wildcard

Category: URL

Purpose: Identifies requests by whether the requested URL points to an asset with a file extension that matches a pattern defined by a wildcard value.

Key information:

- This match condition looks for a URL that ends with a period (.) and the specified file extension. Therefore, make sure that any file extensions specified in the **Value** option do not contain a leading period.
Correct: htm
Incorrect: .htm
- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- The case-sensitivity of file extension comparisons is determined by the **Ignore Case** option.
- Specify multiple file extensions by delimiting each one with a single space.
Sample syntax:
htm html
- This match condition is satisfied when the requested asset's file extension is either an exact match (i.e., match) or does not match one of the specified file extensions.
Example:
Specifying "htm" will match for "htm" assets, but not "html" assets.

Sample Scenario

A sample scenario is provided below.

Note: The following sample configuration assume that this match condition is satisfied when a request matches one of the specified values.

Value	Description
asp aspx php html	This pattern will be satisfied when it finds any URLs that end with the following values: <ul style="list-style-type: none">• .asp• .aspx• .php• .html

URL Path Filename Literal

Category: URL

Purpose: Identifies requests by whether the requested URL points to an asset with a filename that matches a literal value.

Note: For the purposes of this match condition, a filename consists of the name of the requested asset, a period, and the file extension (e.g., index.html).

Key information:

- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The case-sensitivity of filename comparisons is determined by the **Ignore Case** option.
- Replace spaces in the filename with "%20".
- This match condition is satisfied when the requested asset's filename is either an exact match (i.e., match) or does not match the specified value.

URL Path Filename Wildcard

Category: URL

Purpose: Identifies requests by whether the requested URL points to an asset with a filename that matches a pattern defined by a wildcard value.

Note: For the purposes of this match condition, a filename consists of the name of the requested asset, a period, and the file extension (e.g., index.html).

Key information:

- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- The case-sensitivity of filename comparisons is determined by the **Ignore Case** option.
- Specify multiple file extensions by delimiting each one with a single space.

Sample syntax:

index.htm index.html

- Replace spaces in the filename with "%20".
- If a wildcard character (i.e., *) has not been specified, then only an exact match will satisfy this match condition.

Example:

Specifying "Presentation.ppt" will match an asset called "Presentation.ppt," but not one called "Presentation.pptx."

- This match condition is satisfied when the requested asset's filename is either an exact match (i.e., match) or does not match one of the specified patterns.

URL Query Literal

Category: URL

Purpose: Identifies requests by whether the query string of the requested URL matches a literal value.

Key information:

- The value associated with this match condition will be compared against the entire request's query string.
- For the purposes of this option, a query string starts with the first character after the question mark (?) delimiter for the query string. Therefore, the text specified in the **Value** option should not include a leading question mark (?).
- The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.
- The case-sensitivity of query string comparisons is determined by the **Ignore Case** option.
- Certain characters require URL encoding. Use the percentage symbol to URL encode the following characters:

Character	URL Encoding
Space	%20
&	%25
%	%25

- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., %E3%81%93).
 - Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.
Example:
You should include the following characters instead of こんにちは when defining this match condition's value:
%E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF
 - The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By default, our CDN service then decodes those characters before the comparison for this match condition is performed.

- Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- Due to the manner in which cache settings are tracked, this match condition is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

URL Query Regex

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: URL

Purpose: Identifies requests by whether the query string of the requested URL matches the pattern defined by a Perl-compatible regular expression.

Key information:

- The value associated with this match condition will be compared against the entire request's query string.
- For the purposes of this option, a query string starts with the first character after the question mark (?) delimiter for the query string. Therefore, the text specified in the **Value** option should not include a leading question mark (?).
- A value may be defined using any combination of numbers, letters, and/or symbols. This value will be interpreted as a regular expression. Regular expressions are useful for defining a pattern of characters.

Note: Although regular expressions are quite powerful and can generate flexible solutions, it is strongly recommended that a regular expression statement be constructed, or at the very least proofed, by a developer on your team. This will ensure that the desired behavior will take place.

For example, specifying a "." regular expression will match all requests, since the specified pattern matches 0 or more characters. A clearer way to match all requests is through the use of the Always match condition.

- The case-sensitivity of query string comparisons is determined by the **Ignore Case** option.
- Certain characters require URL encoding. Use the percentage symbol to URL encode the following characters:

Character	URL Encoding
Space	%20
&	%25
%	%25

- Double-escape special regular expression characters (e.g., `\^$.+)` to include a backslash in the regular expression.

Example:

Value	Interpreted As
<code>\+</code>	<code>+</code>
<code>\\+</code>	<code>\+</code>

- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., `%E3%81%93`).

- Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.

Example:

You should include the following characters instead of こんにちは when defining this match condition's value:

`%E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF`

- The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By default, our CDN service then decodes those characters before the comparison for this match condition is performed.
- Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- Due to the manner in which cache settings are tracked, this match condition is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

URL Query Wildcard

Category: URL

Purpose: Identifies requests by whether the query string of the requested URL matches a pattern defined by a wildcard value.

Key information:

- The value associated with this match condition will be compared against the entire request's query string.
- For the purposes of this option, a query string starts with the first character after the question mark (?) delimiter for the query string. Therefore, the text specified in the **Value** option should not include a leading question mark (?).
- The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- The case-sensitivity of query string comparisons is determined by the **Ignore Case** option.
- Certain characters require URL encoding. Use the percentage symbol to URL encode the following characters:

Character	URL Encoding
Space	%20
&	%25
%	%25

- Specify multiple values by delimiting each one with a single space.
- Matching against URLs that contain non-US-ASCII characters requires that you specify encoded Unicode characters (e.g., %E3%81%93).

- Encode all Unicode characters before setting the **Value** option. This match condition only accepts encoded Unicode characters.

Example:

You should include the following characters instead of こんにちは when defining this match condition's value:

%E3%81%93%E3%82%93%E3%81%AB%E3%81%A1%E3%81%AF

- The majority of user agents (e.g., web browsers) encode non-US-ASCII characters in the request URL before submitting the request to our CDN. By

default, our CDN service then decodes those characters before the comparison for this match condition is performed.

- Use the **Encoded** option to determine whether your match value, as defined in the Value option, will be decoded prior to comparison with the request's URL path.
 - **URL Normalization:** You must set this option to **Yes** when a URL normalization customization has been applied to your traffic. This configuration ensures that your match value remains encoded.
 - **Standard:** Use the default configuration (i.e., **No**) which allows our service to decode your match value.
- Due to the manner in which cache settings are tracked, this match condition is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

URL Query Parameter Literal

Category: URL

Purpose: Identifies requests by whether a query string parameter in the requested URL matches a literal value.

Tip: This match condition provides an easy way to specify a parameter name/value combination. Consider using the URL Query Parameter Wildcard match condition for more flexibility when matching a query string parameter.

Key information:

- **Parameter name:**
 - Query parameter name comparisons are case-insensitive.
 - Spaces in the parameter name should be replaced with "%20."
 - Only query parameters whose name is an exact match to the specified value may satisfy this condition.
- **Parameter value:**
 - The value associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. All characters, with the

exception of the % symbol, will be treated as a fixed value and cannot take on a special meaning.

- Certain characters require URL encoding. Use the percentage symbol to URL encode the following characters:

Character	URL Encoding
Space	%20
&	%25
%	%25

- The case-sensitivity of parameter value comparisons is determined by the **Ignore Case** option.
- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires the request to contain the specified parameter and its value must match the one defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified parameter.
 - It contains the specified parameter, but its value does not match the one defined in this match condition.
- Due to the manner in which cache settings are tracked, this match condition is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Sample Scenario

A sample scenario is provided below.

Note: The following sample configuration assume that this match condition is satisfied when a request matches the specified value.

Name	Value	Result
User	Joe	This pattern will be satisfied when the query string for a requested URL is "?user=joe."

URL Query Parameter Wildcard

Category: URL

Purpose: Identifies requests by whether a query string parameter in the requested URL matches a pattern defined by a wildcard value.

Key information:

- **Parameter name:**
 - Query parameter name comparisons are case-insensitive.
 - Replace spaces in the parameter name with "%20".
 - Only query parameters whose name is an exact match to the specified value may satisfy this condition.
- **Parameter value:**
 - The value(s) associated with this match condition may be defined using any combination of numbers, letters, and/or symbols. This type of value also supports the use of special characters.

Note: Each specified value may contain one or more asterisks. Each asterisk will match a sequence of one or more characters.

- Certain characters require URL encoding. Use the percentage symbol to URL encode the following characters:

Character	URL Encoding
Space	%20
&	%25
%	%25

- Specify multiple query string parameter values by delimiting each one with a single space. This match condition is satisfied when a request contains one of the specified name/value combinations.

Example 1:

ValueA ValueB

This configuration will match the following query string parameters:

Parameter1=ValueA

Parameter1=ValueB

Example 2:

Value%20A Value%20B

This configuration will match the following query string parameters:

Parameter1=Value%20A

Parameter1=Value%20B

- Only exact matches to at least one of the specified query string name/value combinations will satisfy this condition.
Using the configuration in the above example, the parameter name/value combination "Parameter1=ValueAdd" would not be considered a match. However, setting the **Value** option to either of the following values would match that name/value combination:
 - ValueA ValueB ValueAdd
 - ValueA* ValueB
- The case-sensitivity of header value comparisons is determined by the **Ignore Case** option.
- The **Result** option determines the conditions under which this match condition will be satisfied.
 - **Match:** Requires the request to contain the specified parameter and its value must match one of the patterns defined in this match condition.
 - **Does Not Match:** Requires that the request satisfy either of the following criteria:
 - It does not contain the specified parameter.
 - It contains the specified parameter, but its value does not match one of the patterns defined in this match condition.
- Due to the manner in which cache settings are tracked, this match condition is incompatible with the following features:
 - Complete Cache Fill (End-of-Life)
 - Default Internal Max-Age
 - Force Internal Max-Age
 - Ignore Origin No-Cache
 - Internal Max-Stale

Sample Scenarios

Sample scenarios are provided below.

Note: The following sample configurations assume that this match condition is satisfied when a request matches the specified value.

Name	Value	Result
User	Joe	This pattern will be satisfied when the query string for a requested URL is "?user=joe."
User	*	This pattern will be satisfied when the query string for a requested URL contains a "user" parameter.
Email	Joe*	This pattern will be satisfied when the query string for a requested URL contains an Email parameter that starts with "Joe."

Features

A feature defines the type of action that will be applied to the type of request identified by a set of match conditions.

Types of Features

The available types of features are:

- Access
- Caching
- Comment
- Headers
- Logs
- Origin
- Specialty
- URL
- Web Application Firewall

Access

These features are designed to control access to content.

Name	Purpose
Deny Access (403)	Determines whether all requests are rejected with a 403 Forbidden response.
Token Auth	Determines whether Token-Based Authentication will be applied to a request.
Token Auth Denial Code	Determines the type of response that will be returned to a user when a request is denied due to Token-Based Authentication.
Token Auth Ignore URL Case	Determines whether URL comparisons made by Token-Based Authentication will be case-sensitive.
Token Auth Parameter	Determines whether the Token-Based Authentication query string parameter should be renamed.

Caching

These features are designed to customize when and how content is cached.

Name	Purpose
Bandwidth Parameters	Determines whether bandwidth throttling parameters (i.e., <code>ec_rate</code> and <code>ec_prebuf</code>) will be active.
Bandwidth Throttling	Throttles the bandwidth for the response provided by our edge servers.
Bypass Cache	Determines whether the request can leverage our caching technology.
Cache-Control Header Treatment	Controls the generation of Cache-Control headers by the edge server when External Max-Age feature is active.
Cache-Key Query String	Determines whether the cache-key will include or exclude query string parameters associated with a request.
Cache-Key Rewrite	Rewrites the cache-key associated with a request.
Complete Cache Fill (End-of-Life)	Determines what happens when a request results in a partial cache miss on an edge server.
Compress File Types	Defines the file formats that will be compressed on the server.
Default Internal Max-Age	Determines the default max-age interval for edge server to origin server cache revalidation.
Expires Header Treatment	Controls the generation of Expires headers by an edge server when the External Max-Age feature is active.
External Max-Age	Determines the max-age interval for browser to edge server cache revalidation.
Force Internal Max-Age	Determines the max-age interval for edge server to origin server cache revalidation.
H.264 Support (HTTP Progressive Download)	Determines the types of H.264 file formats that may be used to stream content.
H.264 Support Video Seek Params	Overrides the names assigned to parameters that control seeking through H.264 media when using HTTP Progressive Download.
Honor No-Cache Request	Determines whether an HTTP client's no-cache requests will be forwarded to the origin server.
Ignore Origin No-Cache	Determines whether our CDN will ignore certain directives served from an origin server.
Ignore Unsatisfiable Ranges	Determines the response that will be returned to clients when a request generates a 416 Requested Range Not Satisfiable status code.
Internal Max-Stale	Controls how long past the normal expiration time a cached asset may be served from an edge server when the edge server is unable to revalidate the cached asset with the origin server.

Name	Purpose
Partial Cache Sharing	Determines whether a request can generate partially cached content.
Prevalidate Cached Content	Determines whether cached content will be eligible for early revalidation before its TTL expires.
Refresh Zero-Byte Cache Files	Determines how an HTTP client's request for a 0-byte cache asset is handled by our edge servers.
Set Cacheable Status Codes	Defines the set of status codes that can result in cached content.
Stale Content Delivery on Error	Determines whether expired cached content will be delivered when an error occurs during cache revalidation or when retrieving the requested content from the customer origin server.
Stale While Revalidate	Improves performance by allowing our edge servers to serve stale client to the requester while revalidation takes place.

Comment

The Comment feature allows a note to be added within a rule.

Headers

These features are designed to add, modify, or delete headers from the request or response.

Name	Purpose
Age Response Header	Determines whether an Age response header will be included in the response sent to the requester.
Debug Cache Response Headers	Determines whether a response may include response headers which provide information on the cache policy for the requested asset.
Modify Client Request Header	Overwrites, appends, or deletes a header from a request.
Modify Client Response Header	Overwrites, appends, or deletes a header from a response.
Set Client IP Custom Header	Allows the IP address of the requesting client to be added to the request as a custom request header.

Logs

These features are designed to customize the data stored in raw log files.

Name	Purpose
Custom Log Field 1	Determines the format and the content that will be assigned to the custom log field in a raw log file.
Log Query String	Determines whether a query string will be stored along with the URL in access logs.
Mask Client Subnet	Determines whether a client's IP address will be masked for logging and reporting purposes.

Origin

These features are designed to control how the CDN communicates with an origin server.

Name	Purpose
Maximum Keep-Alive Requests	Defines the maximum number of requests for a Keep-Alive connection before it is closed.
Proxy Special Headers	Defines the set of CDN-specific request headers that will be forwarded from an edge server to an origin server.

Specialty

These features provide advanced functionality that should only be used by advanced users.

Name	Purpose
Cacheable HTTP Methods	Determines the set of additional HTTP methods that can be cached on our network.
Cacheable Request Body Size	Defines the threshold for determining whether a POST response can be cached.
QUIC	Determines whether the client will be informed that our CDN service supports QUIC.
Streaming Optimization	Tunes your caching configuration to optimize performance for live streams and to reduce the load on the origin server.
User Variable	Assigns a value to a user-defined variable that is passed to your bespoke traffic processing solution.

URL

These features allow a request to be redirected or rewritten to a different URL.

Name	Purpose
Follow Redirects	Determines whether requests can be redirected to the hostname defined in the Location header returned by a customer origin server.
URL Redirect	Redirects requests via the Location header.
URL Rewrite	Rewrites the request URL.

Web Application Firewall

The Web Application Firewall feature determines whether a request will be screened by Web Application Firewall.

Access

These features are designed to control access to content.

Deny Access (403)

Category: Access

Purpose: Determines whether requests will be rejected with a 403 Forbidden response.

Valid values are:

Enabled	Result
Yes	Causes all requests that satisfy the matching criteria to be rejected with a 403 Forbidden response.
No	Restores the default behavior. The default behavior is to allow the origin server to determine the type of response that will be returned.

Default Behavior: Disabled

Tip: One possible use for this feature is to associate it with a Request Header Wildcard match condition to block access to HTTP referrers that are using inline links to your content.

Token Auth

Important: This feature requires the Token-Based Authentication feature which must be purchased separately. Contact your CDN account manager to activate it.

Category: Access

Purpose: Determines whether Token-Based Authentication will be applied to a request.

Key information:

- If Token-Based Authentication is enabled, then only requests that provide an encrypted token and comply to the requirements specified by that token will be honored.
- Token values will be encrypted and decrypted using the encryption key(s) defined by the **Primary Key** and the **Backup Key** options on the **Token Auth** page. These encryption keys are platform-specific.
- This feature takes precedence over most features with the exception of the URL Rewrite feature.
- Configure this feature by setting the **Enabled** option to one of the following values:

Enabled	Result
Yes	Protects the requested content with Token-Based Authentication. Only requests from clients that provide a valid token and meet its requirements will be honored. SFTP transactions are excluded from Token-Based Authentication.
No	Restores the default behavior. The default behavior is to allow your Token-Based Authentication configuration to determine whether a request will be secured.

Default Behavior: Disabled.

Token Auth Denial Code

Important: This feature requires the Token-Based Authentication feature which must be purchased separately. Contact your CDN account manager to activate it.

Category: Access

Purpose: Determines the type of response that will be returned to a user when a request is denied access due to Token-Based Authentication.

The available response codes are listed below.

Response Code	Response Name	Description
301	Moved Permanently	This status code redirects unauthorized users to the URL specified in the Location header.
302	Found	This status code redirects unauthorized users to the URL specified in the Location header. This status code is the industry standard method of performing a redirect.
307	Temporary Redirect	This status code redirects unauthorized users to the URL specified in the Location header.
401	Unauthorized	Combining this status code with the WWW-Authenticate response header allows you to prompt a user for authentication.
403	Forbidden	This is the standard 403 Forbidden status message that an unauthorized user will see when trying to access protected content.
404	File Not Found	This status code indicates that the HTTP client was able to communicate with the server, but the requested content was not found.
410	Gone	This status code indicates that the HTTP client was able to communicate with the server, but the requested content was not found. The use of this status code indicates that this condition is permanent.

URL Redirection

This feature supports URL redirection to a user-defined URL when it is configured to return a 3xx status code. This user-defined URL can be specified by performing the following steps:

1. Select a 3xx response code for the Token Auth Denial Code feature.
2. Select "Location" from the **Header Name** option.
3. Set the **Header Value** option to the desired URL.

If a URL is not defined for a 3xx status code, then the standard response page for a 3xx status code will be returned to the user.

Note: URL redirection is only applicable for 3xx response codes.

Note: The **Header Value** option supports alphanumeric characters, quotation marks, and spaces.

Authentication

This feature supports the capability to include the WWW-Authenticate header when responding to an unauthorized request for content protected by Token-Based Authentication. If the WWW-Authenticate header has been set to "basic" in your configuration, then the unauthorized user will be prompted for account credentials.

The above configuration can be achieved by performing the following steps:

1. Select "401" as the response code for the Token Auth Denial Code feature.
2. Select "WWW-Authenticate" from the **Header Name** option.
3. Set the **Header Value** option to "basic."

Note: The WWW-Authenticate header is only applicable for 401 response codes.

Token Auth Ignore URL Case

This feature requires the Token-Based Authentication feature which must be purchased separately. Contact your CDN account manager to activate it.

Category: Access

Purpose: Determines whether URL comparisons made by Token-Based Authentication will be case-sensitive.

The parameters affected by this feature are:

- ec_url_allow
- ec_ref_allow
- ec_ref_deny

Valid values are:

Enabled	Result
Yes	Causes our edge server to ignore case when comparing URLs for Token-Based Authentication parameters.
No	Restores the default behavior. The default behavior is for URL comparisons for Token Authentication to be case-sensitive.

Default Behavior: Disabled.

Token Auth Parameter

Category: Access

Purpose: Determines whether Token-Based Authentication will expect a token value to be specified as an undefined query string parameter or a custom query string parameter.

Key information:

- The **Name** option defines the query string parameter name through which a token may be specified.
- The **Name** option cannot be set to "ec_token."
- Make sure that the name defined in the **Name** option only contains valid URL characters.
- The available states for the **Enabled** option are described below.

Enabled	Result
Yes	The Name option defines the name of the query string parameter where Token-Based Authentication will expect a token value. Example: <code>http://cdn.mydomain.com/secure/asset.html?mycustomname=Token</code>
No	Configures Token-Based Authentication to expect a token value to be included as an undefined query string parameter in the request URL. Example: <code>http://cdn.mydomain.com/secure/asset.html?Token</code>

Default Behavior: Disabled.

Caching

These features are designed to customize when and how content is cached.

Bandwidth Parameters

Important: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Determines whether bandwidth throttling parameters (i.e., `ec_rate` and `ec_prebuf`) will be active.

Bandwidth throttling parameters determine whether the data transfer rate for a client's request will be limited to a custom rate.

Enabled	Result
Yes	Allows our edge servers to honor bandwidth throttling requests.
No	Causes our edge servers to ignore bandwidth throttling parameters. The requested content will be served normally (i.e., without bandwidth throttling).

Default Behavior: Enabled.

Bandwidth Throttling

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Determines whether the response provided by our edge servers will be throttled after a predefined time interval.

Note: This feature should not be leveraged on requests that throttle bandwidth through query string parameters (i.e., `ec_rate` and `ec_prebuf`).

Both of the following options must be defined to properly set up bandwidth throttling.

Option	Description
Prebuf seconds	Set this option to the number of seconds that our edge servers will wait until throttling bandwidth. The purpose of this time period of unrestricted bandwidth is to prevent a media player from experiencing stuttering or buffering issues due to bandwidth throttling.
Kbytes per second	Set this option to the maximum bandwidth (Kb per second) that may be used to deliver the response.

Default Behavior: Off.

Bypass Cache

Note: This feature requires Rules Engine - Advanced Rules when used with the HTTP Large and HTTP Small platforms.

Category: Caching

Purpose: Determines whether the request may leverage our caching technology.

Enabled	Result
Yes	Forces all requests to fall through to the origin server even if the content was previously cached on edge servers.
No	Allows edge servers to cache assets according to the cache policy defined in its response headers.

Default Behavior:

- **HTTP Large/HTTP Small:** Disabled
- **ADN:** Enabled

Note: This feature can be useful for setting up staging directories where content is always fetched from the origin server.

Cache-Control Header Treatment

Category: Caching

Purpose: Controls the generation of Cache-Control headers by the edge server when the External Max-Age feature is active.

Tip: The easiest way to achieve this type of configuration is to place the External Max-Age and the Cache-Control Header Treatment features in the same statement.

Value	Result
Pass Through	<p>Ensures that the following actions will take place:</p> <ul style="list-style-type: none">• Ensures that the Cache-Control header produced by the External Max-Age feature is never added to the response.• If the origin server produces a Cache-Control header, it will pass through to the end-user. <hr/> <p>Note: If the origin server does not set a Cache-Control header, then this mode may cause the response to not include a Cache-Control header.</p> <hr/>
Overwrite	<p>Ensures that the following actions will take place:</p> <ul style="list-style-type: none">• Overwrites the Cache-Control header generated by the origin server.• Adds the Cache-Control header produced by the External Max-Age feature to the response.
Add if Missing	<p>If a Cache-Control header was not received from the origin server, then this option adds the Cache-Control header produced by the External Max-Age feature. This option is useful for ensuring that all assets will be assigned a Cache-Control header.</p>
Remove	<p>This option ensures that a Cache-Control header is not included with the header response. If a Cache-Control header has already been assigned, then it will be stripped from the header response.</p>

Default Behavior: Overwrite

Cache-Key Query String

Category: Caching

Purpose: Determines whether the cache-key will include or exclude query string parameters associated with a request.

Key information:

- Specify one or more query string parameter name(s) that will either be included or excluded from the cache-key.
 - Specify multiple parameters by delimiting each name with a single space.
- This feature determines whether query string parameters will be included or excluded from the cache-key. Additional information is provided for each option below.

Type	Description
Include	Indicates that each specified parameter should be included in the cache-key. A unique cache-key will be generated for each request that contains a unique value for a query string parameter defined in this feature.
Include All	Indicates that a unique cache-key will be created for each request to an asset that includes a unique query string. This type of configuration is not typically recommended since it may lead to a small percentage of cache hits. This will increase the load on the origin server, since it will have to serve more requests. This configuration duplicates the caching behavior known as "unique-cache" on the Query-String Caching page.
Exclude	Indicates that only the specified parameter(s) will be excluded from the cache-key. All other query string parameters will be included in the cache-key.
Exclude All	Indicates that all query string parameters will be excluded from the cache-key. This configuration duplicates the default caching behavior, which is known as "standard-cache" on the Query-String Caching page.

Tip: The power of Rules Engine allows the customization for the implementation of query string caching. For example, query string caching may be configured to only apply to certain locations or file types.

Note: Duplicate the query string caching behavior known as "no-cache" on the **Query-String Caching** page by creating a rule that contains a URL Query Wildcard match condition and a Bypass Cache feature. The URL Query Wildcard match condition should be set to an asterisk (*).

Sample Scenarios

Sample usage for this feature is provided below. A sample request and the default cache-key are provided below.

- **Sample request:**
`http://wpc.0001.edgecastcdn.net/800001/Origin/folder/asset.htm?sessionid=1234&language=EN&userid=01`
- **Default cache-key:**
`/800001/Origin/folder/asset.htm`

Include

Sample configuration:

- **Type:** Include
- **Parameter(s):** language

This type of configuration would generate the following query string parameter cache-key:

`/800001/Origin/folder/asset.htm?language=EN`

Include All

Sample configuration:

- **Type:** Include All

This type of configuration would generate the following query string parameter cache-key:

`/800001/Origin/folder/asset.htm?sessionid=1234&language=EN&userid=01`

Exclude

Sample configuration:

- **Type:** Exclude
- **Parameter(s):** sessionid userid

This type of configuration would generate the following query string parameter cache-key:

`/800001/Origin/folder/asset.htm?language=EN`

Exclude All

Sample configuration:

- **Type:** Exclude All

This type of configuration would generate the following query string parameter cache-key:

`/800001/Origin/folder/asset.htm`

Cache-Key Rewrite

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Rewrites the cache-key associated with a request.

A cache-key is the relative path that identifies an asset for the purposes of caching. In other words, our servers will check for a cached version of an asset according to the path defined in its cache-key.

Configure this feature by defining both of the following options:

Option	Description
Source	<p>Identify requests whose cache-key will be rewritten via a relative path. This relative path, which starts directly after the hostname in the CDN URL, is shown in blue font below.</p> <p>http://wpc.0001.edgecastcdn.net/800001/CustomOrigin/Path/file.ext</p> <hr/> <p>Important: Verify that the specified pattern does not conflict with the parent match conditions defined for this feature.</p> <hr/> <p>Valid syntax varies according to the number of customer origins that you would like to match.</p> <ul style="list-style-type: none">• Specific Origin: Match against a specific origin by specifying it via a content access point followed by a regular expression for the desired path. Syntax: <i>{Content Access Point}/{Regular Expression}</i> Example: Match all requests to a customer origin called myorigin via the following syntax: <i>/800001/myorigin/(.*)</i>• Multiple Customer Origins: Match against multiple origins by delimiting each desired customer origin using a " ". Syntax: <i>/80{Account Number}/{Origin 1} {Origin 2} {Origin N}/{Regular Expression}</i> Example: Use this pattern to define a capture group that matches a customer origin group called marketing or sales: <i>/800001/(marketing sales){Regular Expression}</i>• All Customer Origins: Match against all of your customer origins by specifying ".*" or "(.*)".

Option	Description
	<p>Syntax: <code>/80{Account Number}/{.*}/{Regular Expression}</code></p> <p>Example: Use this pattern to define a capture group that matches all of your customer origins: <code>/800001/{.*}/{Regular Expression}</code></p> <hr/> <p>Note: You may only define a customer origin using a literal value (e.g., marketing) or one of the regular expressions patterns defined above. All other regular expression syntax is disallowed when defining a customer origin. This limitation does not apply to the relative path defined after the content access point.</p> <hr/>
Destination	<p>Identify the new cache-key via a relative path. This relative path, which starts directly after the hostname in the CDN URL, is shown in blue font in the following sample CDN URL.</p> <p>http://wpc.0001.edgecastcdn.net/800001/CustomOrigin/Path/file.ext</p> <hr/> <p>Tip: Use HTTP variables to dynamically construct this relative path. However, you may not use response metadata (e.g., <code>%{resp_ResponseHeader}</code>) when defining a cache-key.</p> <hr/> <p>Valid syntax varies according to number of origins being matched in the Source option.</p> <ul style="list-style-type: none"> <p>Specific Origin: Specify the content access point that identifies the desired origin followed by a regular expression for the desired path.</p> <p>Syntax: <code>{Content Access Point}/{Regular Expression}</code></p> <p>Example: Use a backreference (i.e., <code>\$1</code>) to include text captured from the source. <code>/800001/mycustomerorigin/new-path/\$1</code></p> <p>All or Multiple Customer Origins: If the Source option matches all or multiple customer origins using a capture group, then you may use a backreference (i.e., <code>\$1</code>) to reinsert the name of the customer origin into the cache-key.</p> <p>Syntax: <code>/80{Account Number}/\$1/{Regular Expression}</code></p> <p>Example: Use this pattern to ensure that requests point to the same customer origin: <code>/800001/\$1/{Regular Expression}</code></p> <hr/> <p>Note: You may define a customer origin using either a literal value (e.g., marketing) or a backreference. All other regular expression syntax is disallowed when defining a customer origin. This limitation does not apply to the relative path defined after the content access point.</p> <hr/>

Default Behavior: A request's cache-key is determined by the request URI.

Complete Cache Fill (End-of-Life)

Important: This feature, which is undergoing end-of-life, has been superseded by the Partial Cache Sharing feature. Please update your rules to use the Partial Cache Sharing feature instead of this one.

Category: Caching

Purpose: Determines how a request that results in a partial cache miss will be handled.

A partial cache miss describes the cache status for an asset that was not completely downloaded to an edge server. If an asset is only partially cached on an edge server, then the next request for that asset will be forwarded again to the origin server.

Important: This feature is not available for the HTTP Small or the ADN platform. The typical traffic on both of these platforms consists of relatively small assets. The size of the assets served through these platforms helps mitigate the effects of partial cache misses, since the next request will typically result in the asset being cached on that POP.

A partial cache miss typically occurs after a user aborts a download or for assets that are solely requested using HTTP range requests. This feature is most useful for large assets where users will not typically download them from start to finish (e.g., videos). As a result, this feature is only available on the HTTP Large platform.

It is recommended to enable this feature for larger files, since it will reduce the load on your customer origin server and increase the speed at which your customers download your content.

Enabled	Result
Yes	Forces the edge server to initiate a background fetch of the asset from the origin server. After which, the asset will be in the edge server's local cache.
No	Prevents an edge server from performing a background fetch for the asset. This means that the next request for that asset from that region will cause an edge server to request it from the customer origin server.

Default Behavior: Disabled.

Important: Due to the manner in which cache settings are tracked, this feature cannot be associated with the following match conditions: AS Number, Client IP Address, Cookie Parameter Literal, Cookie Parameter Regex, Cookie Parameter Wildcard, Country, Device, Edge CNAME, Referring Domain Literal, Referring Domain Wildcard, Request Header Literal, Request Header Regex, Request Header Wildcard, Request Method, Request Scheme, URL Query Literal, URL Query Regex, URL Query Wildcard, URL Query Parameter Literal, and URL Query Parameter Wildcard.

Compress File Types

Category: Caching

Purpose: Defines the file formats that are eligible for edge server compression.

Important: Edge server compression should only be defined by either Rules Engine or on the Compression page. Attempting to configure edge server compression through both Rules Engine and the Compression page will create an invalid configuration that will not be applied to your account. Furthermore, no additional configuration changes will be applied to your account until this conflict is resolved.

A file format can be specified using its media type (aka content type). Media type is platform-independent metadata that allows our servers to identify the file format of a particular asset. A list of common media types is provided below.

Media Type	Description
text/plain	Plain text files
text/html	HTML files
Cascading Style Sheets (CSS)	text/css
application/x-javascript	JavaScript
application/javascript	JavaScript

Key information:

- Specify multiple media types by delimiting each one with a single space.
- This feature will only compress assets whose size is less than 1 MB. Larger assets will not be compressed by our servers.
- Certain types of content, such as images, video, and audio media assets (e.g., JPG, MP3, MP4, etc.), are already compressed. Additional compression on these types of assets will not significantly diminish file size. Therefore, the compression of these types of assets is not recommended.
- Wildcard characters, such as asterisks, are not supported.
- Edge server compression should only be defined by either Rules Engine or on the **Compression** page. Before adding this feature to a rule, make sure to set the **Compression Disabled** option on the **Compression** page for the platform to which this rule will be applied.

Default Internal Max-Age

Category: Caching

Purpose: Determines the default max-age interval for edge server to origin server cache revalidation. In other words, the amount of time that will pass before an edge server will check whether a cached asset matches the asset stored on the origin server.

Key information:

- This action will only take place for responses from an origin server that did not assign a max-age indication in the Cache-Control or Expires header.
- This action will not take place for assets that are not deemed cacheable.
- This action does not affect browser to edge server cache revalidations.

Note: Browser to edge server revalidation is determined by the Cache-Control or Expires headers sent to the browser, which can be customized with the External Max-Age feature.

- The results of this action do not have an observable effect on the response headers and the content returned from edge servers for your content, but it may have an effect on the amount of revalidation traffic sent from edge servers to your origin server.
- Configure this feature by performing the following steps:
 1. Select the status code for which this default internal max-age policy will be applied.

Note: This feature is only applicable when the response to the request results in the selected status code.

2. Specify an integer value and then selecting the desired time unit (i.e., seconds, minutes, hours, etc.). This value defines the default internal max-age interval.
- Due to the manner in which cache settings are tracked, this feature cannot be associated with the following match conditions: AS Number, Client IP Address, Cookie Parameter Literal, Cookie Parameter Regex, Cookie Parameter Wildcard, Country, Device, Edge CNAME, Referring Domain Literal, Referring Domain Wildcard, Request Header Literal, Request Header Regex, Request Header Wildcard, Request Method, Request Scheme, URL Query Literal, URL Query Regex, URL Query Wildcard, URL Query Parameter Literal, and URL Query Parameter Wildcard.

Default Value: Requests that do not provide a max-age indication via either their Cache-Control or Expires header will be assigned a default internal max-age interval of 7 days.

Expires Header Treatment

Category: Caching

Purpose: Determines how an edge server will handle the Expires header in the response sent to the client. This feature overrides the Expires header generated by the External Max-Age feature.

Tip: The easiest way to achieve this type of configuration is to place the External Max-Age and the Expires Header Treatment features in the same statement.

Value	Result
Pass Through	<p>Ensures that the following actions will take place:</p> <ul style="list-style-type: none">Ensures that the Expires header produced by the External Max-Age feature is never added to the response.If the origin server produces an Expires header, it will pass through to the end-user. <hr/> <p>Note: If the origin server does not set an Expires header, then this mode may cause the response to not include an Expires header.</p>
Overwrite	<p>Ensures that the following actions will take place:</p> <ul style="list-style-type: none">Overwrites the Expires header generated by the origin server.Adds the Expires header produced by the External Max-Age feature to the response.
Add if Missing	<p>If an Expires header was not received from the origin server, then this option adds the Expires header produced by the External Max-Age feature. This option is useful for ensuring that all assets will be assigned an Expires header.</p>
Remove	<p>Ensures that an Expires header is not included with the header response. If an Expires header has already been assigned, then it will be stripped from the header response.</p>

Default Behavior: Overwrite

External Max-Age

Category: Caching

Purpose: Determines the max-age interval for browser to edge server cache revalidation. In other words, the amount of time that will pass before a browser can check for a new version of an asset from an edge server.

Enabling this feature will generate Cache-Control:max-age and Expires headers from our edge servers and send them to the HTTP client. By default, these headers will overwrite those created by the origin server. However, the Cache-Control Header Treatment and the Expires Header Treatment features may be used to alter this behavior.

Key information:

- This action does not affect edge server to origin server cache revalidations. These types of revalidations are determined by the Cache-Control/Expires headers received from the origin server, and can be customized with the Default Internal Max-Age and the Force Internal Max-Age features.
- Configure this feature by specifying an integer value and then selecting the desired time unit (i.e., seconds, minutes, hours, etc.). This value defines the external max-age interval that will be applied.
- Setting this feature to a negative value causes our edge servers to send a Cache-Control:no-cache and an Expires time that is set in the past with each response to the browser. Although an HTTP client will not cache the response, this setting will not affect our edge servers' ability to cache the response from the origin server.

Default Behavior: The Cache-Control/Expires headers cached with the response of the origin server will pass through to the browser.

Force Internal Max-Age

Category: Caching

Purpose: Determines the max-age interval for edge server to origin server cache revalidation. In other words, the amount of time that will pass before an edge server can check whether a cached asset matches the asset stored on the origin server.

Key information:

- This feature will override the max-age interval defined in Cache-Control or Expires headers generated from an origin server.
- This feature does not affect browser to edge server cache revalidations. These types of revalidations are determined by the Cache-Control or Expires headers sent to the browser.
- This feature does not have an observable effect on the response delivered by an edge server to the requester. However, it may have an effect on the amount of revalidation traffic sent from our edge servers to the origin server.

- Configure this feature by:

1. Select the status code for which this internal max-age policy will be applied.

Note: This feature is only applicable when the response to the request results in the selected status code.

2. Specify an integer value and then selecting the desired time unit (i.e., seconds, minutes, hours, etc.). This value defines the internal max-age interval that will be applied.

- Due to the manner in which cache settings are tracked, this feature cannot be associated with the following match conditions: AS Number, Client IP Address, Cookie Parameter Literal, Cookie Parameter Regex, Cookie Parameter Wildcard, Country, Device, Edge CNAME, Referring Domain Literal, Referring Domain Wildcard, Request Header Literal, Request Header Regex, Request Header Wildcard, Request Method, Request Scheme, URL Query Literal, URL Query Regex, URL Query Wildcard, URL Query Parameter Literal, and URL Query Parameter Wildcard.

Default Behavior: Disabled. An internal max-age interval will not be assigned to requested assets. If the original header does not contain caching instructions, then the asset will be cached according to the active setting in the Default Internal Max-Age feature.

H.264 Support (HTTP Progressive Download)

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Determines the types of H.264 file formats that may be leveraged when streaming content via HTTP Progressive Download.

Key information:

- Define a space-delimited set of allowed H.264 file extensions in the **File Extensions** option.

Tip: By default, HTTP Progressive Download supports MP4 and F4V file extensions. The **File Extensions** option overrides this behavior. Maintain MP4 and F4V support by including those file extensions when setting this option.

- Each file extension must start with a period.

Sample configuration:

.mp4 .f4v

Default Behavior: HTTP Progressive Download supports MP4 and F4V media by default.

H.264 Support Video Seek Params

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Overrides the names assigned to parameters that control seeking through H.264 media when using HTTP Progressive Download.

Key information:

- This feature overrides the default name assigned to the following seek parameters:
 - ec_seek
 - ec_end

Tip: Leverage this capability to match seek parameter names to the names assigned to your player's native parameters.

- A valid name may only consist of alphanumeric characters, dashes, underscores, and periods.

Default Behavior: By default, HTTP Progressive Download looks for ec_seek and ec_end parameters in the query string.

Honor No-Cache Request

Category: Caching

Purpose: Determines whether a client's no-cache request will be forwarded to the origin server.

A no-cache request occurs when the HTTP client sends a Cache-Control:no-cache and/or Pragma:no-cache header in the HTTP request.

Enabled	Result
Yes	Allows an HTTP client's no-cache requests to be forwarded to the origin server, and the origin server will return the response headers and the body through the edge server back to the HTTP client.
No	Restores the default behavior. The default behavior is to prevent no-cache requests from being forwarded to the origin server.

Tip: For all production traffic, it is highly recommended to leave this feature in its default disabled state. Otherwise, origin servers will not be shielded from end-users who may inadvertently trigger many no-cache requests when refreshing web pages, or from the many popular media players that are coded to send a no-cache header with every video request. Nevertheless, this feature can be useful to apply to certain non-production staging or testing directories, in order to allow fresh content to be pulled on-demand from the origin server.

Note: The cache status that will be reported for a request that is allowed to be forwarded to an origin server due to this feature is TCP_Client_Refresh_Miss. The Cache Statuses report, which is available in the Core Reports module, provides statistical information by cache status. This allows you to track the number and percentage of requests that are being forwarded to an origin server due to this feature.

Default Behavior: Disabled.

Ignore Origin No-Cache

Category: Caching

Purpose: Determines whether our CDN will ignore the following directives served from an origin server:

- Cache-Control: private
- Cache-Control: no-store
- Cache-Control: no-cache
- Pragma: no-cache

Key information:

- Configure this feature by defining a space-delimited list of status codes for which the above directives will be ignored.
- The set of valid status codes for this feature are: 200, 203, 300, 301, 302, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, and 505.
- Due to the manner in which cache settings are tracked, this feature cannot be associated with the following match conditions: AS Number, Client IP Address, Cookie Parameter Literal, Cookie Parameter Regex, Cookie Parameter Wildcard, Country, Device, Edge CNAME, Referring Domain Literal, Referring Domain Wildcard, Request Header Literal, Request Header Regex, Request Header Wildcard, Request Method, Request Scheme, URL Query Literal, URL Query Regex, URL Query Wildcard, URL Query Parameter Literal, and URL Query Parameter Wildcard.

Default Behavior: The default behavior is to honor the above directives.

Ignore Unsatisfiable Ranges

Category: Caching

Purpose: Defines the response for a request that generates a 416 Requested Range Not Satisfiable status code.

By default, this status code is returned when the specified byte-range request cannot be satisfied by an edge server and an If-Range request header field was not specified.

Enabled	Result
Yes	Prevents our edge servers from responding to an invalid byte-range request with a 416 Requested Range Not Satisfiable status code. Instead our servers will deliver the requested asset and return a 200 OK to the client.
No	Restores the default behavior. The default behavior is to honor the 416 Requested Range Not Satisfiable status code.

Default Behavior: Disabled.

Internal Max-Stale

Category: Caching

Purpose: Controls how long past the normal expiration time a cached asset may be served from an edge server when the edge server is unable to revalidate the cached asset with the origin server.

Normally, when an asset's max-age time expires, the edge server will send a revalidation request to the origin server. The origin server will then respond with either a 304 Not Modified to give the edge server a fresh lease on the cached asset, or else with 200 OK to provide the edge server with an updated version of the cached asset.

If the edge server is unable to establish a connection with the origin server while attempting such a revalidation, then this Internal Max-Stale feature controls whether, and for how long, the edge server may continue to serve the now-stale asset.

Note that this time interval starts when the asset's max-age expires, not when the failed revalidation occurs. Therefore, the maximum period during which an asset can be served without successful revalidation is the amount of time specified by the combination of max-age plus max-stale. For example, if an asset was cached at 9:00 with a max-age of 30 minutes and a max-stale of 15 minutes, then a failed revalidation attempt at 9:44 would result in an end-user receiving the stale cached asset, while a failed revalidation attempt at 9:46 would result in the end user receiving a 504 Gateway Timeout.

Any value configured for this feature is superseded by Cache-Control:must-revalidate or Cache-Control:proxy-revalidate headers received from the origin server. If either of those headers is received from the origin server when an asset is initially cached, then the edge server will not serve a stale cached asset. In such a case, if the edge server is unable to revalidate with the

origin when the asset's max-age interval has expired, then the edge server will return a 504 Gateway Timeout.

Key information:

- Configure this feature by performing the following steps:
 1. Select the status code for which this max-stale policy will be applied.

Note: This feature is only applicable when the response to the request results in the selected status code.
 2. Specify an integer value and then selecting the desired time unit (i.e., seconds, minutes, hours, etc.). This value defines the internal max-stale that will be applied.
- Due to the manner in which cache settings are tracked, this feature cannot be associated with the following match conditions: AS Number, Client IP Address, Cookie Parameter Literal, Cookie Parameter Regex, Cookie Parameter Wildcard, Country, Device, Edge CNAME, Referring Domain Literal, Referring Domain Wildcard, Request Header Literal, Request Header Regex, Request Header Wildcard, Request Method, Request Scheme, URL Query Literal, URL Query Regex, URL Query Wildcard, URL Query Parameter Literal, and URL Query Parameter Wildcard.

Default Behavior: 2 minutes

Partial Cache Sharing

Category: Caching

Purpose: Determines whether a request can generate partially cached content.

This partial cache may then be used to fulfill new requests for that content until the requested content is fully cached.

Enabled	Result
Yes	Requests can generate partially cached content.
No	Requests can only generate a fully cached version of the requested content.

Default Behavior: Disabled.

Prevalidate Cached Content

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Determines whether cached content will be eligible for early revalidation before its TTL expires.

Define the amount of time prior to the expiration of the requested content's TTL during which it will be eligible for early revalidation.

Default Behavior: Revalidation may only take place after the cached content's TTL has expired.

Refresh Zero-Byte Cache Files

Category: Caching

Purpose: Determines how our edge servers handle a client's request for content that was cached as a 0-byte file.

Valid values are:

Enabled	Result
Yes	Causes our edge server to re-fetch the asset from the origin server.
No	Restores the default behavior. The default behavior is to serve up valid cache assets upon request.

Tip: This feature is not required for correct caching and content delivery, but may be useful as a workaround. For example, dynamic content generators on origin servers can inadvertently result in 0-byte responses being sent to the edge servers. These types of responses are typically cached by our edge servers. If you know that a 0-byte response is never a valid response for such content, then this feature can prevent these types of assets from being served to your clients.

Default Behavior: Disabled.

Set Cacheable Status Codes

Category: Caching

Purpose: Defines the set of status codes that can result in cached content.

Note: By default, caching is only enabled for 200 OK responses.

Define a space-delimited set of the desired status codes.

Key information:

- Please also enable the Ignore Origin No-Cache feature. If that feature is not enabled, then non-200 OK responses may not be cached.
- The set of valid status codes for this feature are: 203, 300, 301, 302, 305, 307, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 500, 501, 502, 503, 504, and 505.
- This feature cannot be used to disable caching for responses that generate a 200 OK status code.

Default Behavior: Caching is only enabled for responses that generate a 200 OK status code.

Stale Content Delivery on Error

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Determines whether expired cached content will be delivered when an error occurs during cache revalidation or when retrieving the requested content from a customer origin server.

Enabled	Result
Yes	Stale content will be served to the requester when an error occurs during a connection to an origin server. <hr/> Tip: Use the Internal Max-Stale feature to configure the length of time after TTL expiration during which stale content may be delivered. <hr/>
No	The origin server's error will be forwarded to the requester.

Default Behavior: Disabled.

Stale While Revalidate

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Caching

Purpose: Improves performance by allowing our edge servers to serve stale content to the requester while revalidation takes place.

Configure this feature by specifying an integer value and then selecting the desired time unit (i.e., seconds, minutes, hours, etc.). This value defines the length of time past TTL expiration during which stale content may be delivered.

The following formula indicates the length of time during which stale content may be delivered:
TTL + Stale While Revalidate Time

Default Behavior: Revalidation must take place before the requested content can be served.

Comment

Category: General

Purpose: Adds a note.

One use for this feature is to provide additional information on the general purpose of a rule or why a particular match condition or feature was added to the rule.

Key information:

- A maximum of 150 characters may be specified.
- Make sure to only use alphanumeric characters.
- This feature does not affect the behavior of the rule. It is merely meant to provide an area where you can provide information for future reference or that may help when troubleshooting the rule.

Headers

These features are designed to add, modify, or delete headers from the request or response.

Age Response Header

Category: Headers

Purpose: Determines whether an Age response header will be included in the response sent to the requester.

Enabled	Result
Yes	The Age response header will be included in the response sent to the requester.
No	The Age response header will be excluded from the response sent to the requester.

Default Behavior: Disabled.

Debug Cache Response Headers

Category: Headers

Purpose: Determines whether a response may include debug cache response headers which provide information on the cache policy for the requested asset.

Our CDN returns debug cache response headers when both of the following are true:

- The Debug Cache Response Headers feature has been enabled on the desired request.
- The request defines the set of debug cache response headers that will be included in the response.

Request debug cache response headers by setting the X-EC-Debug request header to the desired debug cache response headers:

X-EC-Debug: {*Debug Cache Header 1*},{*Debug Cache Header 2*},{*Debug Cache Header N*}

Example:

X-EC-Debug: x-ec-cache,x-ec-check-cacheable,x-ec-cache-key,x-ec-cache-state

Enabled	Result
Yes	Requests that include the X-EC-Debug header return a response that includes the specified debug cache response headers.
No	The response always excludes debug cache response headers.

Default Behavior: Disabled.

Modify Client Request Header

Category: Headers

Purpose: Each request contains a set of request headers that describe it. This feature can either:

- Append or overwrite the value assigned to a request header. If the specified request header does not exist, then this feature will add it to the request.
- Delete a request header from the request.

Note: Requests that are forwarded to an origin server will reflect the changes made by this feature.

One of the following actions can be performed on a request header:

Option	Description	Example
Overwrite	The request header value will be set to the specified value.	Request header value (Client): Value1 Request header value (Rules Engine): Value2 New request header value: Value2
Append	The specified value will be added to end of the existing request header value.	Request header value (Client): Value1 Request header value (Rules Engine): Value2 New request header value: Value1Value2
Delete	Deletes the specified request header.	Request header value (Client): Value1 Modify Client Request Header configuration: Delete the request header in question. Result: The specified request header will not be forwarded to the origin server.

Key information:

- Make sure that the value specified in the **Name** option is an exact match for the desired request header.
- Case is not taken into account for the purpose of identifying a header.
For example, any of the following variations of the Cache-Control header name can be used to identify it:
 - cache-control
 - CACHE-CONTROL
 - cachE-Control
- Make sure to only use alphanumeric characters, dashes, or underscores when specifying a header name.
- Deleting a header will prevent it from being forwarded to an origin server by our edge servers.
- The **Value** option supports the use of HTTP variables to dynamically set the request header.
- The following headers are reserved and cannot be modified by this feature:
 - forwarded-for
 - host
 - via
 - warning
 - x-forwarded-for
 - All header names that start with "x-ec" are reserved.

Modify Client Response Header

Category: Headers

Purpose: Each response contains a set of response headers that describe it. This feature can either:

- Append or overwrite the value assigned to a response header. If the specified request header does not exist, then this feature will add it to the response.
- Delete a response header from the response.

Note: By default, response header values are defined by an origin server and by our edge servers.

One of the following actions can be performed on a response header:

Option	Description	Example
Overwrite	The response header value will be set to the specified value.	Response header value (Origin/Edge Server): Value1 Response header value (Rules Engine): Value2 New response header value: Value2
Append	The specified value will be added to the end of the existing response header value.	Response header value (Origin/Edge Server): Value1 Response header value (Rules Engine): Value2 New response header value: Value1Value2
Delete	Excludes a header from the response.	Response header value (Origin/Edge Server): Value1 Response header value (Rules Engine): <i>Blank</i> Result: The specified response header will be excluded from the response provided by our edge servers to the requester.

Key information:

- Make sure that the value specified in the **Name** option is an exact match for the desired response header.
- Case is not taken into account for the purpose of identifying a header.
For example, any of the following variations of the Cache-Control header name can be used to identify it:
 - cache-control
 - CACHE-CONTROL
 - cachE-Control
- Deleting a header will prevent it from being forwarded to the requester.
- The **Value** option supports the use of HTTP variables to dynamically set the response header.
- The following headers are reserved and cannot be modified by this feature:
 - accept-ranges
 - age
 - connection
 - content-encoding
 - content-length
 - content-range
 - date
 - server
 - trailer
 - transfer-encoding
 - upgrade
 - vary
 - via
 - warning
 - All header names that start with "x-ec" are reserved.

Set Client IP Custom Header

Category: Headers

Purpose: Adds a custom request header that identifies the requesting client by IP address.

Configure this feature by performing the following steps:

1. Determine whether a client's IP address will be logged via a custom header.

Valid values are:

Enabled	Result
Yes	The requesting client's IP address will be logged in a custom request header.
No	Restores the default behavior. By default, the requesting client's IP address is not logged in a request header.

Default Behavior: Disabled.

2. If this feature is enabled, define the name of the custom request header to which the requesting client's IP address will be logged.

The **Name** option defines the name of the custom request header where the client's IP address will be stored.

Note: This feature allows a customer origin server to find out client IP addresses through a custom request header. If the request is served from cache, then the origin server will not be informed of the client's IP address. Therefore, it is recommended that this feature be used with ADN or assets that will not be cached.

Please make sure that the specified header name does not match any of the following:

- Standard request header names. A list of standard header names can be found in RFC 2616.
- Reserved header names:
 - forwarded-for
 - host
 - via
 - warning
 - x-forwarded-for
 - All header names that start with "x-ec" are reserved.

Logs

These features are designed to customize the data stored in raw log files.

Custom Log Field 1

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Logs

Purpose: Determines the format and the content that will be assigned to the custom log field in a raw log file.

The main purpose behind this custom field is to allow you to determine which request and response header values will be stored in your log files.

Note: By default, the custom log field is called "x-ec_custom-1." However, the name of this field can be customized from the **Raw Log Settings** page.

The formatting that you should use to specify request and response headers is defined below.

Header Type	Format	Examples
Request Header	<code>%{RequestHeader}i</code>	<code>%{Accept-Encoding}i</code> <code>%{Referer}i</code> <code>%{Authorization}i</code>
Response Header	<code>%{ResponseHeader}o</code>	<code>%{Age}o</code> <code>%{Content-Type}o</code> <code>%{Cookie}o</code>

Key information:

- A custom log field can contain any combination of header fields and plain text.
- Valid characters for this field include the following: alphanumeric (i.e., 0-9, a-z, and A-Z), dashes, colons, semi-colons, apostrophes, commas, periods, underscores, equal signs, parentheses, brackets, and spaces. The percentage symbol and curly braces are only allowed when used to specify a header field.
- The spelling for each specified header field must match the desired request/response header name.
- When specifying multiple headers, it is recommended to use a separator to identify each header.

For example, an abbreviation may be used to identify each header.

Sample Syntax:

AE: %{Accept-Encoding}i A: %{Authorization}i CT: %{Content-Type}o

Default Value: -

Log Query String

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Logs

Purpose: Determines whether a query string will be stored along with the URL in access logs.

Enabled	Result
Yes	Allows the storage of query strings when recording URLs in an access log. If a URL does not contain a query string, then this option will not have an effect.
No	Restores the default behavior. The default behavior is to ignore query strings when recording URLs in an access log.

Default Behavior: Disabled.

Mask Client Subnet

Category: Logs

Purpose: Masks the client's subnet for logging and reporting purposes.

This feature masks a client's subnet by:

- **IPv4:** Setting the last octet to 0.

Example:

Applying this feature to a client whose IP address is 100.100.200.50 would result in the following masked IP address: 100.100.200.0

- **IPv6:** Setting the last 32 bits to 0.

Example:

Applying this feature to a client whose IP address is 2002:db5:2:gg22:42:1234 would result in the following masked IP address: 2002:db5:2:gg22:0:0

Tip: Use this feature as part of your General Data Protection Regulation (GDPR) compliance strategy.

Default Behavior: Disabled. By default, the system logs a client's IP address without masking.

Optimizer

These features determine whether a request will undergo the optimizations provided by Edge Optimizer.

Edge Optimizer

Note: This feature requires the ADN platform and the Edge Optimizer feature.

Category: Optimizer

Purpose: Determines whether Edge Optimizer may be applied to a request.

If this feature has been enabled, then the following criteria must also be met before the request will be processed by Edge Optimizer:

- The requested content must use an edge CNAME URL.
- The edge CNAME referenced in the URL must correspond to a site whose configuration has been activated in a rule.

Enabled	Result
Yes	Indicates that the request is eligible for Edge Optimizer processing.
No	Restores the default behavior. The default behavior is to deliver content over the ADN platform without any additional processing.

Default Behavior: Disabled.

Edge Optimizer - Instantiate Configuration

Note: This feature requires the ADN platform and the Edge Optimizer feature.

Category: Optimizer

Purpose: Instantiates or activates the Edge Optimizer configuration associated with a site.

Key information:

- This feature should only instantiate an active site configuration. Please make sure that the **Active** option has been marked on the site's configuration.
- Instantiation of a site configuration is required before requests to the corresponding edge CNAME can be processed by Edge Optimizer.
- This instantiation only needs to be performed a single time per site configuration. A site configuration that has been instantiated will remain in that state until the Edge Optimizer – Instantiate Configuration feature that references it is removed from the rule.
- The instantiation of a site configuration does not mean that all requests to the corresponding edge CNAME will automatically be processed by Edge Optimizer. The Edge Optimizer feature determines whether an individual request will be processed.

Default Behavior: Site configurations are inactive by default.

Origin

These features are designed to control how the CDN communicates with an origin server.

Maximum Keep-Alive Requests

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Origin

Purpose: Defines the maximum number of requests for a Keep-Alive connection before it is closed.

Setting the maximum number of requests to a low value is strongly discouraged and may result in performance degradation.

Key information:

- Specify this value as a whole integer.
- Do not include commas or periods in the specified value.

Default Value: 10,000 requests

Proxy Special Headers

Category: Origin

Purpose: Defines the set of CDN-specific request headers that will be forwarded from an edge server to an origin server.

Key information:

- Each CDN-specific request header defined in this feature will be forwarded to an origin server.
- Prevent a CDN-specific request header from being forwarded to an origin server by removing it from this list.

Default Behavior: All CDN-specific request headers will be forwarded to the origin server.

Specialty

These features provide advanced functionality that should only be used by advanced users.

Cacheable HTTP Methods

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Specialty

Purpose: Determines the set of HTTP methods that are eligible for caching on our network.

Key information:

- This feature supports the following HTTP methods: POST and PUT.
- Use a space character to delimit multiple HTTP methods.
- By default, only requests whose body is smaller than 14 Kb are eligible for caching.

Tip: Use the Cacheable Request Body Size feature to set the maximum request body size for cache-eligible requests.

- GET requests are unaffected by this feature. Including or excluding the GET method when defining this feature will not impact whether GET requests are eligible for caching.

Default Behavior: Only GET requests are eligible for caching.

Cacheable Request Body Size

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Specialty

Purpose: Defines the threshold for determining whether a response may be cached.

This threshold is determined by specifying a maximum request body size. Requests that contain a larger request body will not be cached.

Key information:

- This feature is only applicable when POST or PUT responses are eligible for caching. Use the Cacheable HTTP Methods feature to enable POST/PUT request caching.
- The request body is taken into consideration for:
 - x-www-form-urlencoded values
 - Ensuring a unique cache-key
- Defining a large maximum request body size may impact data delivery performance.
 - Recommended Value: 14 Kb
 - Minimum Value: 1 Kb

Default Behavior: 14 Kb

QUIC

Category: Specialty

Purpose: Determines whether the client will be informed that our CDN service supports QUIC.

Important: This feature may only be used with the Always or Edge CNAME match conditions.

Key information:

- This feature may only be enabled for either of the following traffic profiles:
 - All traffic
 - One or more edge CNAMEs.
- Configure this feature by setting the **Enabled** option to **Yes**.

Enabled	Result
Yes	Informs the client that our CDN service supports QUIC by including the alt-svc header in the response sent to the client. This response header also informs the client the set of QUIC versions that our CDN service supports.
No	<hr/> Important: This state is disallowed. <hr/> Policies that contain a rule with this feature set to "No" cannot be deployed.

Default Behavior: Disabled. The default behavior is to be agnostic with regards to the alt-svc response header.

Revalidate While Stale

Important: This feature only affects how often our system communicates with your origin server. Do not confuse this feature with the Stale While Revalidate feature that allows the delivery of stale content.

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Specialty

Purpose: Determines whether our service will attempt revalidation with your origin server when communication has been temporarily disabled due to repeated origin server availability issues.

Key information:

- **Background Information:** The system temporarily disables an origin server configuration after repeated TCP connection failures. This is known as stale mode. Once stale mode expires, the system will allow a single revalidation request to the origin server. The origin server's response determines how the system will handle future requests.
 - **Success:** If the system is able to establish a TCP connection with the origin server, then it will enable the origin server configuration for all future requests.
 - **Failure:** The system will extend stale mode for a longer time period. This process is repeated until a TCP connection can be established with the origin server.
- This feature determines whether the system will attempt to connect to your origin server while it is in stale mode.

Enabled	Result
Yes	The system will attempt to connect to your origin server while it is in stale mode according to the time interval defined within the Revalidate While Stale Timer feature.
No	The system will not attempt to connect to your origin server while it is in stale mode.

- This capability requires both the Revalidate While Stale and Revalidate While Stale Timer features.
- This feature does not affect requests that have been assigned a Cache-Control: must-revalidate directive. This directive requires a successful validation on the origin server in order to serve stale content.

Default Behavior: Disabled.

Revalidate While Stale Timer

Important: This feature only affects how often our system communicates with your origin server. Do not confuse this feature with the Stale While Revalidate feature that allows the delivery of stale content.

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Specialty

Purpose: Determines how often, in seconds, the system will attempt to connect to an unavailable origin server.

Key information:

- This capability requires both the Revalidate While Stale and Revalidate While Stale Timer features.
- This feature determines how often the system will attempt revalidation with an origin server whose configuration is in stale mode due to repeated TCP connection failures. However, it does not apply to requests that have been assigned a Cache-Control: must-revalidate directive.

Default Behavior: By default, the system will not attempt to connect to your origin server while it is in stale mode.

Streaming Optimization Feature

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: Specialty

Purpose: Tunes your caching configuration to optimize performance for live streams and to reduce the load on the origin server.

Key information:

- The scope of this feature is automatically restricted to HTTP streaming requests.
- Only use this feature to automatically tune your caching configuration for live streams. Do not enable it for video on-demand (VOD) streaming.

- Configure this feature by setting the **Enabled** option to **Yes**.

Enabled	Result
Yes	Applies an optimized caching configuration to your HTTP streaming requests.
No	Important: This state is disallowed. Policies that contain a rule with this feature set to "No" cannot be deployed.

- This feature cannot be disabled. Since this feature should not be enabled for VOD streaming, it is important to configure your match conditions so that they are not satisfied by VOD playback requests.

Sample Scenario:

One way to only apply this feature to live streams is to place it below a URL Path Regex match condition that is configured as follows:

Setting	Value
Result	Match
Value	/2[14].*
Ignore Case	yes

The above configuration will only apply this feature for Dynamic Cloud Packaging live streaming playback requests.

- This feature may only be used with the following match conditions: Always, Customer Origin, CDN Origin, URL Path Literal, URL Path Regex, URL Path Wildcard, URL Path Directory Literal, URL Path Directory Wildcard, URL Path Filename Literal, and URL Path Filename Wildcard.

Default Behavior: Disabled. Your cache configuration will determine the caching behavior for HTTP streaming requests.

User Variable

Category: Specialty

Purpose: Assigns a value to a user-defined variable that is passed to your bespoke traffic processing solution.

Key information:

- This feature is only applicable when:
 - Custom logic that is specific to your traffic controls how requests will be processed.

Note: Our CDN service supports the capability to define customized traffic processing logic. This solution addresses specialized customer needs that cannot be implemented via Rules Engine. If your CDN traffic requires a bespoke solution, then please contact our Solutions Engineer team.

- This bespoke solution expects a variable.

Note: Upon implementing a bespoke solution, a member of our Solutions Engineer team will provide information about a variable's purpose and the information that should be passed to it.

Note: Variables defined by this feature will be ignored when either a bespoke solution has not been defined for your CDN account or the specified variable has not been defined within your solution.

- HTTP variables may not be used when defining a variable.
- Valid characters for the variable name are: alphanumeric, dashes, underscores, and periods.

URL

These features allow a request to be redirected or rewritten to a different URL.

Follow Redirects

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: URL

Purpose: Determines whether requests may be redirected to the hostname defined in the Location header returned by a customer origin server.

Important: All requests, regardless of HTTP method (e.g., POST and PUT), are redirected as GET requests.

Note: Requests can only be redirected to edge CNAMEs that correspond to the same platform.

Valid values are:

Enabled	Result
Yes	Requests are allowed to follow redirects.
No	Requests will not be redirected.

Default Behavior: Disabled.

URL Redirect

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: URL

Purpose: Redirects requests via the Location header.

Configuration

The configuration of this feature requires setting the following options:

Option	Description
Source	<p>Identify requests that will be redirected via a relative path. This relative path, which starts directly after the hostname in the CDN URL, is shown in blue font below.</p> <p>http://wpc.0001.edgecastcdn.net/800001/CustomOrigin/Path/file.ext</p> <hr/> <p>Important: Verify that the specified pattern does not conflict with the parent match conditions defined for this feature.</p> <hr/> <p>Valid syntax varies according to the number of customer origins that you would like to match.</p> <ul style="list-style-type: none">• Specific Origin: Match against a specific origin by specifying it via a content access point followed by a regular expression for the desired path. Syntax: <i>{Content Access Point}/{Regular Expression}</i> Example: Match all requests to a customer origin called myorigin via the following syntax: <i>/800001/myorigin/(.*)</i>• Multiple Customer Origins: Match against multiple origins by delimiting each desired customer origin using a " ". Syntax: <i>/80{Account Number}/{(Origin 1} {Origin 2} {Origin N} {Regular Expression}</i> Example: Use this pattern to define a capture group that matches a customer origin group called marketing or sales: <i>/800001/(marketing sales)/{Regular Expression}</i>• All Customer Origins: Match against all of your customer origins by specifying ".*" or "(.*)". Syntax: <i>/80{Account Number}/{(.*)/{Regular Expression}</i>

Option	Description
	<p>Example: Use this pattern to define a capture group that matches all of your customer origins: <code>/800001/(.*)/{Regular Expression}</code></p> <hr/> <p>Note: You may only define a customer origin using a literal value (e.g., marketing) or one of the regular expressions patterns defined above. All other regular expression syntax is disallowed when defining a customer origin. This limitation does not apply to the relative path defined after the content access point.</p> <hr/>
Destination	<p>Define the URL path to which the above requests will be redirected.</p> <p>Dynamically construct this URL path using:</p> <ul style="list-style-type: none"> • A regular expression pattern • HTTP variables <hr/> <p>Note: You may not use response metadata (e.g., <code>%{resp_ResponseHeader}</code>) when constructing this path.</p> <hr/> <p>Tip: It is highly recommended to use an absolute URL. The use of a relative URL may redirect CDN URLs to an invalid path.</p> <hr/> <p>Note: URL encoding should not be applied to the specified URL.</p> <hr/>
Code	Select the response code that will be returned to the requester.

Sample Scenario

In this example, we will demonstrate how to redirect an edge CNAME URL that resolves to this base CDN URL:

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures`

Qualifying requests will be redirected to this base edge CNAME URL:

`http://cdn.mydomain.com/resources`

This URL redirection may be achieved through the following configuration:

The screenshot shows the 'if' configuration interface. At the top, there's a 'Match' section with 'General' and 'Always' dropdowns. Below that is a '+ Feature' button. The main configuration area has four fields: 'Feature' (set to 'URL'), 'URL Redirect' (set to 'URL Redirect'), 'Source' (set to '/800001/marketing/brochures(.*)'), and 'Destination' (set to '%{scheme}://cdn.mydomain.com/resources\$1'). The 'Code' field is set to 'Select'.

URL Redirect Example

Key points:

- The URL Redirect feature defines the request URLs that will be redirected. As a result, additional match conditions are not required. Although the match condition was defined as "Always," only requests that point to the "brochures" folder on the "marketing" customer origin will be redirected.

Sample requests:

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/widgets.pdf`

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/campaignA/final/productC.ppt`

`http://marketing.mydomain.com/brochures/widgets.pdf`

`http://brochures.mydomain.com/campaignA/final/productC.ppt`

- All matching requests will be redirected to the edge CNAME URL defined in the **Destination** option.

Sample scenario #1:**Sample request (CDN URL):**

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/widgets.pdf`

Request URL (after redirect):

`http://cdn.mydomain.com/resources/widgets.pdf`

Sample scenario #2:**Sample request (Edge CNAME URL):**

`http://marketing.mydomain.com/brochures/widgets.pdf`

Request URL (after redirect):

`http://cdn.mydomain.com/resources/widgets.pdf`

Sample scenario #3:**Sample request (Edge CNAME URL):**

`http://brochures.mydomain.com/campaignA/final/productC.ppt`

Request URL (after redirect):

`http://cdn.mydomain.com/resources/campaignA/final/productC.ppt`

- The Request Scheme (`%{scheme}`) variable was leveraged in the **Destination** option. This ensures that the request's scheme remains unchanged after redirection.
- The URL segments that were captured from the request are appended to the new URL via `"$1."`

URL Rewrite

Note: This capability requires Rules Engine - Advanced Rules which must be purchased separately. Contact your CDN account manager to activate it.

Category: URL

Purpose: Rewrites the request URL.

Key information:

- The configuration of this feature requires setting the following options:

Option	Description
Source	<p>Identify requests that will be rewritten via a relative path. This relative path, which starts directly after the hostname in the CDN URL, is shown in blue font below.</p> <p>http://wpc.0001.edgecastcdn.net/800001/CustomerOrigin/Path/file.ext</p> <hr/> <p>Important: Verify that the specified pattern does not conflict with the parent match conditions defined for this feature.</p> <hr/> <p>Valid syntax varies according to the number of customer origins that you would like to match.</p> <ul style="list-style-type: none">Specific Origin: Match against a specific origin by specifying it via a content access point followed by a regular expression for the desired path. Syntax: <i>{Content Access Point}/{Regular Expression}</i> Example: Match all requests to a customer origin called myorigin via the following syntax: <i>/800001/myorigin/(.*)</i>Multiple Customer Origins: Match against multiple origins by delimiting each desired customer origin using a " ". Syntax: <i>/80{Account Number}/{Origin 1} {Origin 2} {Origin N}/{Regular Expression}</i> Example: Use this pattern to define a capture group that matches a customer origin group called marketing or sales: <i>/800001/(marketing sales){Regular Expression}</i>All Customer Origins: Match against all of your customer origins by specifying ".*" or "(.*)". Syntax: <i>/80{Account Number}/(.*)/{Regular Expression}</i>

Option	Description
	<p>Example: Use this pattern to define a capture group that matches all of your customer origins: <i>/800001/(.*)/{Regular Expression}</i></p> <hr/> <p>Note: You may only define a customer origin using a literal value (e.g., marketing) or one of the regular expressions patterns defined above. All other regular expression syntax is disallowed when defining a customer origin. This limitation does not apply to the relative path defined after the content access point.</p> <hr/>
Destination	<p>Define the relative path to which the above requests will be rewritten. This relative path, which starts directly after the hostname in the CDN URL, is shown in blue font in the following sample CDN URL.</p> <p><i>http://wpc.0001.edgecastcdn.net/800001/</i><i>CustomerOrigin/Path/file.ext</i></p> <hr/> <p>Tip: Use HTTP variables to dynamically construct this relative path. However, you may not use response metadata (e.g., <i>%{resp_ResponseHeader}</i>) when constructing this relative path.</p> <hr/> <p>Valid syntax varies according to number of origins being matched in the Source option.</p> <ul style="list-style-type: none"> <p>Specific Origin: Specify the content access point that identifies the desired origin followed by a regular expression for the desired path.</p> <p>Syntax: <i>{Content Access Point}/{Regular Expression}</i></p> <p>Example: Use a backreference (i.e., \$1) to include text captured from the source. <i>/800001/mycustomerorigin/new-path/\$1</i></p> <p>All or Multiple Customer Origins: If the Source option matches all or multiple customer origins using a capture group, then you may use a backreference (i.e., \$1) to reinsert the name of the customer origin into the URL.</p> <p>Syntax: <i>/80{Account Number}/\$1/{Regular Expression}</i></p> <p>Example: Use this pattern to ensure that requests point to the same customer origin: <i>/800001/\$1/{Regular Expression}</i></p> <hr/> <p>Note: You may define a customer origin using either a literal value (e.g., marketing) or a backreference. All other regular expression syntax is disallowed when defining a customer origin. This limitation does not apply to the relative path defined after the content access point.</p> <hr/>

- This feature allows our edge servers to rewrite the URL without performing a traditional redirect. This means that the requester will receive the same response code as if the rewritten URL had been requested.
- This feature takes precedence when multiple features will be applied to a request.

Sample Scenario

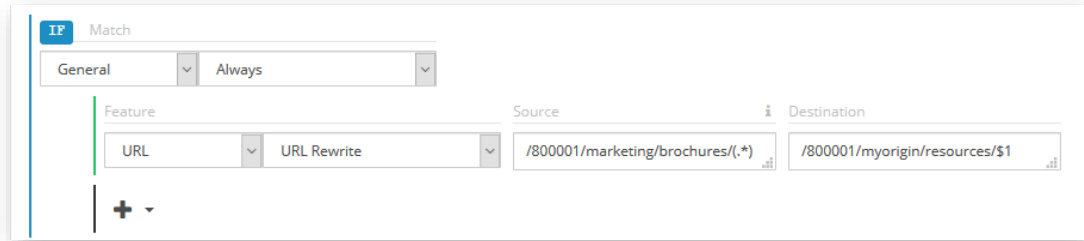
In this example, we will demonstrate how to rewrite an edge CNAME URL that resolves to this base CDN URL:

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/`

Qualifying requests will be rewritten to this base CDN URL:

`http://wpc.0001.edgecastcdn.net/800001/myorigin/resources/`

This URL rewrite may be achieved through the following configuration:



URL Rewrite Example

Key points:

- The URL Rewrite feature defines the request URLs that will be rewritten. As a result, additional match conditions are not required. Although the match condition was defined as "Always," only requests that point to the "brochures" folder on the "marketing" customer origin will be rewritten.

Sample requests:

`http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/widgets.pdf`

`http://marketing.mydomain.com/brochures/widgets.pdf`

`http://brochures.mydomain.com/campaignA/final/productC.ppt`

- All matching requests will be rewritten using the relative path defined in the **Destination** option.
Sample scenario #1:
Sample request (CDN URL):
http://wpc.0001.edgecastcdn.net/800001/marketing/brochures/widgets.pdf
Request URL (after rewrite):
http://wpc.0001.edgecastcdn.net/800001/myorigin/resources/widgets.pdf
Sample scenario #2:
Sample request (Edge CNAME URL):
http://marketing.mydomain.com/brochures/widgets.pdf
Request URL (after rewrite):
http://wpc.0001.edgecastcdn.net/800001/myorigin/resources/widgets.pdf
Sample scenario #3:
Sample request (Edge CNAME URL):
http://brochures.mydomain.com/campaignA/final/productC.ppt
Request URL (after rewrite):
http://wpc.0001.edgecastcdn.net/800001/myorigin/resources/campaignA/final/productC.ppt
- The URL segments that were captured from the request are appended to the new URL via "\$1."

Compatibility

This feature includes matching criteria that must be met before it can be applied to a request. In order to prevent setting up conflicting match criteria, this feature is incompatible with the following match conditions:

- AS Number
- CDN Origin
- Client IP Address
- Customer Origin
- Request Scheme
- URL Path Literal
- URL Path Regex
- URL Path Wildcard
- URL Path Directory Literal
- URL Path Directory Wildcard
- URL Path Extension Literal
- URL Path Extension Wildcard
- URL Path Filename Literal
- URL Path Filename Wildcard
- URL Query Literal
- URL Query Regex
- URL Query Wildcard
- URL Query Parameter Literal
- URL Query Parameter Wildcard