

## Extended Match and Condition Expressions

<https://campus.barracuda.com/doc/4523/>

You use extended match and condition expressions in content rules, HTTP request rewrite rules, and HTTP response rewrite rules to match the requests and responses that you want to be handled by these rules.

### Quick Reference

<b>Expressions</b>	<ul style="list-style-type: none"> <li>• Element Match</li> <li>• (Expression) [Join (Expression) ...]</li> </ul>
<b>Joins</b>	<ul style="list-style-type: none"> <li>• &amp;&amp;,   </li> </ul>
<b>Elements</b>	<ul style="list-style-type: none"> <li>• Request Elements: Method, HTTP-Version, Client-IP, URI, URI-Path, Header</li> <li>• Request Parameters: Parameter, Pathinfo</li> <li>• Response Elements: Status-code, Response-Header</li> </ul>
<b>Operators</b>	<ul style="list-style-type: none"> <li>• Matching: eq, neq, req, nreq</li> <li>• Containing: co, nco, rco, nrco</li> <li>• Existence: ex, nex</li> </ul>

### Structure

An expression consists of one or more **element matches** that are combined with join operators to indicate AND and OR operations to combine the element matches. Use parentheses to delimit individual element matches when using join operators. You can nest parentheses.

An element match consists of an **element**, an optional **element name**, and an **operator** followed by an optional **value**. Some elements like Header require an element name like User-Agent, whereas some elements like HTTP-Version require no further qualification. Also, some operators like eq (stands for equals) require a value, whereas some operators like ex (stands for exists) require no value.

Tokens are delimited by space and the parenthesis characters. You can use double quotes (") to enclose single tokens which contain parenthesis characters or spaces. Use the backslash (\) to *escape*, or remove the special meaning of the special characters (space and parentheses).

### Operators

You can use the following operators in an element match. The operators are case-insensitive; for

example, eq, Eq, and EQ are all treated the same.

Operator	Description
eq	True if the operand is equal to the given value. A case-insensitive string comparison is performed. Thus, a value of 01 is not the same as a value of 1, whereas one and ONE are treated the same.
neq	True if the operand is not equal to the given value. A case-insensitive string comparison is performed.
co	True if the operand contains the given value.
nco	True if the operand does not contain the given value.
rco	True if the operand contains the given value, which is treated as a regular expression.
nrco	True if the operand does not contain the given value, which is treated as a regular expression.
req	True if the operand matches the given value, which is treated as a regular expression.
nreq	True if the operand does not match the given value, which is treated as a regular expression.
ex	True if the operand exists. A value is not required.
nex	True if the operand does not exist. A value is not required.

## Elements

The elements listed in the following table can only be used in certain expressions. Elements and element names are case-insensitive; for example, Method and METHOD are treated the same.

Element	Description	Restrictions
Method	The HTTP method that was received in the request. Example: (Method eq GET)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>
HTTP-Version	The version of the HTTP protocol of the request. Example: (HTTP-Version eq HTTP/1.1)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>

Header	An HTTP header in the request. An element name is required after the header. Example: (Header Accept co gzip) This example checks if the Accept header contains the string gzip.	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> <li>Response rewrite conditions</li> </ul>
Client-IP	The IP address of the client sending the request. The IP address can be either a host IP address or subnet IP address specified by a mask. Only eq and neq operations are allowed for this element. Examples: (client-ip eq 192.168.1.0/24), (Client-IP eq 192.168.1.10)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>
URI	The uniform resource identifier in the request. This includes any query parameters in the request. Example: (URI rco /abc.*html?userid=b)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>
URI-path	The path portion of the URI, which excludes any query parameters. Example: (URI-path req /\.*copy%20[^/]*)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> </ul>
Pathinfo	The portion of the URL which is interpreted as PATH_INFO on the server. The Barracuda Load Balancer ADC uses a set of known extensions to determine whether a portion of the URL provides information about a file path. For example, if the request URL is /twiki/view.cgi/Engineering, then /Engineering is considered to be the pathinfo rather than part of the URL. Example: (PathInfo rco abc*)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>
Parameter	A parameter in the query string part of the URL and serves as a name-value pair. The special \$NONAME_PARAM parameter is used when the parameter name is absent. Examples: (Parameter sid eq 1234), (Parameter \$NONAME_PARAM co abcd)	Only allowed in: <ul style="list-style-type: none"> <li>Extended match expressions</li> <li>Request rewrite conditions</li> </ul>
Status-code	The status code of the response returned by the servers. Example: (status-code eq 302)	Only allowed in response rewrite conditions.
Response-header	The HTTP response header in the response. The Response-header term must be followed by the name of the header on which the action is to be applied. Example: (Response-Header Set-Cookie co sessionid)	Only allowed in response rewrite conditions.

---

## Combining Expressions

---

You can use the following join operators to combine expressions:

Operator	Description
	True if either of the expressions are true.
&&	True only if both the expressions are true.

When you combine element matches, you must enclose each element in parentheses. Combining element matches *without* parentheses *is not allowed*.

Example: `(Header cookie ex) && (URI rco .*\.html) && (Method eq GET)`

## Nesting Expressions

---

You can nest sub-expressions by enclosing parentheses within expressions. This makes the expression more readable and clear.

Example: `(HTTP-Version eq HTTP/1.1) && ((Header Host eq www.example.com) || (Header Host eq website.example.com))`

## Escaping

---

Escape special characters by enclosing entire values in double quotes (") or prefixing characters with a backslash (\). The backslash character escapes all characters, not just the special characters. For example, `\c` stands for the character `c`. In other words, the backslash followed by any character stands for the character, whether or not that character has a special meaning in the syntax.

The space character and the parentheses characters are special characters because they cause the parser to split the string into tokens at these separators. In some cases, it is required to specify these characters as part of the value itself. For example, the User-Agent header typically contains both spaces and parentheses, as in:

User-Agent: Mozilla/5.0 (Linux i686; en-US; rv:1.8.1.3) Firefox/2.0.0.3

When you write expressions or conditions to match the header example above, you can escape the

special characters as follows:

- Header User-Agent eq “Mozilla/5.0 (Linux i686; en-US; rv:1.8.1.3) Firefox/2.0.0.3”
- Header User-Agent eq Mozilla/5.0\ \ (Linux\ i686;\ en-US;\ rv:1.8.1.3)\ \ Firefox/2.0.0.3

To specify the double-quote character itself, it must be escaped with a back-slash. This is true inside a quoted string, or a non-quoted string. Note that the single quote character has no special meaning, and is treated as any other character.

To specify the back-slash character itself, it must be escaped as `\\`. This is true within both quoted strings and non-quoted strings.

## Macro Definitions

The Barracuda Load Balancer ADC supports several macros to assist in configuring policies. The following table describes these macros arranged by the areas where they can be used. The URI in these cases does not include the host.

Macro	Description
<code>\$SRC_ADDR</code>	Inserts the source (client) IP address. You can use it for the new value (Rewrite Value parameter) when inserting or rewriting a header.
<code>\$URI</code>	Specifies the complete request URI including the query string. If you are rewriting or redirecting the URI, use this macro in the new value.
<code>\$AUTH_USER</code>	Adds the username. <sup>(1)</sup> <sup>(2)</sup> <sup>(3)</sup>
<code>\$AUTH_PASSWD</code>	Adds the password. <sup>(1)</sup> <sup>(2)</sup> <sup>(3)</sup>
<code>\$AUTH_GROUPS</code>	Adds the user roles. <sup>(1)</sup> <sup>(2)</sup> <sup>(3)</sup>
URL ACLs	
<code>\$NONAME_PARAM</code>	Inserts a parameter with no name (see <a href="#">No Name Parameters</a> )

### Notes:

<sup>(1)</sup> The URL is not protected. Access control or authentication is disabled. The value substituted for the macro is the special string `NCURLNotProtected`.

<sup>(2)</sup> The client has not logged in. The value substituted for the macros is the special string `NCNoUserSession`.

<sup>(3)</sup> The user does not belong to any groups. The value substituted for `$AUTH_GROUPS` is the special string `NCNoUserRoles`.

## No Name Parameters

There might be times when you want to configure a parameter without a name. For example, consider a site that displays a pop-up window to visitors. A Javascript adds a query string that results in the following GET request:

```
GET /ad?xyz
```

The Barracuda Load Balancer ADC does not learn “no name” parameters such as query strings like "GET /ad?0" added by a Javascript. As a workaround: add a null value [URL ACL](#).

The Barracuda Load Balancer ADC treats xyz as the value of a parameter. In this case, you cannot create an exception rule based on the xyz value because there is no way to associate it with a named parameter.

To address such situations (that is, requests with parameter name-value pairs of the type ?xyz or ?=xyz where xyz is the value), you can use a special token: \$NONAME\_PARAM (case-insensitive). This token lets you create an expression for a parameter without a name, as in the following examples:

```
set = parameter $NONAME_PARAM ex
```

```
set = parameter $NONAME_PARAM eq 0
```

```
set = parameter $noname_param co xyz
```

© Barracuda Networks Inc., 2025 The information contained within this document is confidential and proprietary to Barracuda Networks Inc. No portion of this document may be copied, distributed, publicized or used for other than internal documentary purposes without the written consent of an official representative of Barracuda Networks Inc. All specifications are subject to change without notice. Barracuda Networks Inc. assumes no responsibility for any inaccuracies in this document. Barracuda Networks Inc. reserves the right to change, modify, transfer, or otherwise revise this publication without notice.