Dimensions, metrics, and attributes in HTTP monitoring

You can define custom metrics to monitor certain types of measurable data specific to your network environment or software. You can analyze and categorize operation attributes (text entities retrieved from requests and responses of a web application operation) . You can extract miscellaneous parameters from the request or response body. You can also define grouping attributes by specifying the rules to be applied to the request URL or response body.

A custom metric is a non-standard metric you can use to count values specific to your web application. Custom metrics can be, for example, the number of items sold or the total value of a transaction. The values are reported as user-defined metrics on the report server.

Extracting an operation attribute, which is a text entity retrieved from the requests and responses of a web application operation, can help to diagnose and report specific events or errors caused by end-user actions.

Miscellaneous parameters are text strings available in the URL request or response body. You need to define recognizable text patterns conveying the miscellaneous parameters that then can be used in DC RUM reports as dimensions and enable additional ways of grouping data under specific categories of your interest. Miscellaneous parameters, unlike parameters extracted for URLs with parameters, are not defined together with an accompanying URL. When extracting Miscellaneous parameters, only the initial hit triggering the web page load is taken into account. The extracted Miscellaneous parameters must not be longer than 1030 bytes.

Grouping attributes are text strings available in the URL request or response body that uniquely identify clients. You need to define recognizable text patterns conveying the grouping attributes that then can be used in DC RUM reports as dimensions and enable additional ways of grouping data under specific categories of your interest.

To define dimensions, metrics, or attributes that you wish to extract and monitor:

1. Open the **Dimensions, Metrics, Attributes** tab.
    a. In the **RUM Console**, open **Devices and Connections Manage Devices**.
    b. For the AMD you want to manage, select **Open configuration**.
    c. On the **AMD Configuration** screen, select **Software Services User-Defined Software Services**.
    d. On the **User-Defined Software Services** screen, select the software service for which you want to edit dimensions, metrics, or attributes.
    e. Right-click in the **Rules** table to add or edit a rule.
    f. On the **Edit Rule** screen, open the **Dimensions, Metrics, Attributes** tab.
2. Add or open a definition to be monitored and reported in a specific way.
   In the **Dimensions, Metrics, Attributes** table, right-click and choose **Add** to create a new definition, or **Open** to open an existing definition. The **Dimensions, Metrics, Attributes** window will open.
3. **Choose how the value should be reported**

   Custom metric
   You can define up to five custom metrics to monitor certain types of measurable data that is specific to your network environment or software. Use this mechanism if you want to obtain non-standard measurements extracted from the HTTP, XML, or SOAP traffic.

   A custom metric is a non-standard metric you can use to count values specific to your web application. Custom metrics can be, for example, the number of items sold or the total value of a transaction. The values are reported as user-defined metrics on the report server.

   Custom metrics can be defined on the level of:

   - software service
   - URL
   - URL parameters

   The number of metrics for each level is globally limited to five.

   Choose the custom metric level that matches the characteristics of the traffic to monitor. For example, to monitor an HTTP software services in which URL monitoring is not deployed, define the custom metric for the software service custom metric level. However, to define URLs and URL parameters, use the appropriate custom metric levels for each. The level you choose should directly relate to where the information to monitor can be found

   Also consider performance issues when choosing a custom metric level. For example, if you define custom metrics for a software service globally, the rule will be applied to all URLs that the analyzer processes, and could possibly negatively impact the performance of the AMD. This may be unnecessary if you only want to extract this type of data only from two types of URLs, in which case you can define the rule at the URL or even URL parameter level.

The custom metric values are collected during traffic monitoring and can be configured in the RUM Console for the HTTP and transactional software services (SOAP and XML). By default, the new metric names are derived from the field name in an HTTP or XML request. These names can be changed later on the report server for easier identification. The custom metric values are presented by the report server in dedicated columns that show the number of occurrences and the sum and average values.

Grouping attribute
Grouping attributes are text strings available in the URL request or response body that uniquely identify clients. You need to define recognizable text patterns conveying the grouping attributes that then can be used in DC RUM reports as dimensions and enable additional ways of grouping data under specific categories of your interest.

Note that the rules defined for URL or URL with parameters have a higher priority than those defined at the software service level.

Miscellaneous parameter
Miscellaneous parameters are text strings available in the URL request or response body. You need to define recognizable text patterns conveying the miscellaneous parameters that then can be used in DC RUM reports as dimensions and enable additional ways of grouping data under specific categories of your interest. Miscellaneous parameters, unlike parameters extracted for URLs with parameters, are not defined together with an accompanying URL. When extracting Miscellaneous parameters, only the initial hit triggering the web page load is taken into account. The extracted Miscellaneous parameters must not be longer than 1030 bytes.

When defining the rule, you can use one of six methods to search for the parameters in the request URL or you can also use one additional method to search for the dimension in the response body.

Note that the rules defined for URL or URL with parameters have a higher priority than those defined at the software service level.

Operation attribute
Configuring extraction of request operation attributes is almost identical to the process for custom metrics, the main difference being that the custom metrics functionality is used to extract and report numeric values, while the request operation attributes relate to textual data.

4. **Choose where the value should be reported**
   The options available for this step depend on the selection made in the **Choose how the value should be reported** option.

   - **Custom metric (1, 2, 3, 4, 5)**

     Select a metric category from 1 through 5, where each number corresponds to one of five custom metric categories. The values extracted will be reported by CAS in the same custom metrics category.
   - **Grouping attribute (1, 2, 3)**

     You can define up to three grouping attributes. Each matching rule in each of the set of rules is taken into account regardless of its order, but the rules applied to the HTTP requests always takes precedence over the response rules and it is not advised to use both of them to extract one attribute. The extracted Grouping attributes should not be longer than 255 bytes. The order of entries is irrelevant because each matching hit is used to extract the grouping attributes.
   - **Miscellaneous parameter (1, 2, 3, 4, 5, 6, page name)**

     You can define up to six parameters. The number of available Miscellaneous parameters, however, is limited by the number of defined URL with parameter definitions at the URL level. For example, if you use up all four parameter definitions available for a particular URL, you can define only two more Miscellaneous parameters.
   - **Operation attribute (1, 2, 3, 4, 5)**

     Select a category from 1 through 5, where the selection from the drop-down list corresponds to one of five operation attribute categories. Values extracted will be reported by the CAS in the same category.
5. **Choose where to search for the value**
   You can retrieve the values from a number of entities:

   - Request or Response Headers
   - Request or Response Body
   - Request parameter
6. Apply additional search and transformation rules.
   Choose one of the available search methods to detect the values. For more information, see Choosing method of searching within the payload
7. **Advanced settings**
   Define the matching based on pattern or absence of a pattern.

a. Select a condition
Pattern presence
The response for a defined category is reported if a given pattern was detected, which is the default setting.

Pattern absence
The response for a defined category is reported if a given pattern was *not* detected.

b. Enter a **Host Pattern**.
Enter the pattern to match in the *host* field in HTTP requests. The pattern should consist of a case-sensitive string that is expected to be found in the host name and may also contain an optional wild-card character "*" to signify any number of any characters. If spaces are included in the pattern, the pattern must be enclosed in a pair of double or single quotation marks. Otherwise, quoting the pattern is optional.

> **Note:**
>
> The eligible hosts are selected by limiting the size of the host group to the one defined by the narrowest condition.
>
> In other words, for a particular sample of monitored traffic data, if one host pattern defines a set of hosts that is included in the set of hosts defined by another pattern, a match will be attempted on the *smaller group first*. If the monitored traffic data does match the application response definition for the smaller group of hosts, there is no attempt to match the same traffic data to the application response definition for the larger set.
>
> For example, the pattern **`*.abc`** defines a larger set of hosts than the pattern **`*.myhost.abc`** . In this case, for any given sample of monitored traffic data, first, an attempt to match it to the response definition for **`*.myhost.abc`** and, if successful, there is no attempt to match it to the response definition for **`*.abc`** .
>
> Within a given host group, all path patterns that match are taken into consideration while searching for responses.

To ensure meaningful results, no two different patterns defining host names should be matched by a single host, except for the pattern "*". This means that *the same* patterns can repeat in the configuration file, but for any two *different* patterns, the search will not find a host that matches *both* of them.

For example, if there are two patterns such as "*t*" and "*u*", the host names Jupiter and Saturn both match both of the patterns because both of the names contain the letters "u" and "t". So, if you are monitoring two such hosts, modify your pattern so that no host matches both of them. However, if there are no such hosts in the monitored data, the above pattern will cause no problems.

Similar requirements apply to the patterns for paths and response pattern text.

c. Enter a **Path Pattern**.
Enter a pattern to match the `path` field in HTTP requests, after removing from it the leading portion, up to and including the host name. The pattern should consist of a case-sensitive string that is expected to be found in the path and it may also contain optional wild-card characters "*", to signify any number of any characters. Note that if spaces are included in the pattern, the pattern must be enclosed in double or single quotation marks. Otherwise, quoting the pattern is optional.

For example, if the path in the HTTP request is `http://www.somehost.one/sales/eg/index.jhtml`,

enter **`/sales/eg/index.jhtml`** .

d. **Match only when response has one of the following HTTP status code**
Match only when response has one of the indicated HTTP status codes. The HTTP status codes can be defined by providing the HTTP status code range. Use the official HTTP status codes to narrow down qualifying responses.

- 1xx - Informational
- 2xx - Successful
- 3xx - Redirection
- 4xx - Client error
- 5xx - Server error