

**Contents**

--------------------------------------------------------------------------------------------

Author: Andrea Minuto (andrea.minuto@it.ibm.com)  
IBM Cloud Integration Expert Labs Europe

IBM Italia S.p.A.  
Date: January 26, 2025

[1. Overview 3](#_Toc138070711)

[2. Features 4](#_Toc138070712)

[3. Compatibility Table 6](#_Toc138070713)

[4. Disclaimer 7](#_Toc138070714)

[5. Versions 8](#_Toc138070715)

[6. Configuration directives 14](#_Toc138070716)

[7. Algorithm and record layouts 29](#_Toc138070717)

[8. Examples 33](#_Toc138070718)

# Overview

Web Tracking Apache Module is a module for Apache Web Server 2.4.x (IBM HTTP Server 9.0.x) 64-bit.

The supported platforms are Red Hat Enterprise Linux 8.x and 9.x

The main functionality is to track the input (requests) and the output (responses) of HTTP/HTTPS roundtrips inside an Apache Web Server.

WARNING: The supported protocols/standards are only HTTP/1.0 and HTTP/1.1.

Its use has a dual purpose:

1. Tracking for legal and security control purposes of all HTTP transactions.
2. Tracking for debugging purposes of specific web transactions that present anomalies or whose behavior is not fully understood.

Most likely it may run smoothly on other distributions provided they have the correct version of the required library, that is glibc 2.28.

The source code, not part of the solution, is written in compliance with the C17 specifications and is compatible with all the distributed platforms for which a supported version of the IBM HTTP Server exists.

The used compiler is gcc version 12.2.1 20221121 (Red Hat 12.2.1-7).

The module is based on a proprietary extension of the open-source library zlib V1.2.11 (<https://www.zlib.net/>), fully included and compiled within the module for portability and security.

In this context the supported compression algorithms are gzip and deflate.

The module does not manage security issues or intercepts web threats by itself, that oversees web server administrators and other more specific modules.

# Features

The module is called Web Tracking Apache Module and the shared library, that is the executable associated with the solution, has the name: mod\_web\_tracking\_rh8\_24.so

The distribution package is a compressed file: WebTrackingRT-RH8.zip.

It consists of a main directory – “webtracking” – with two children directories - bin (executable), lib (the modules for the web server).

Through the configuration of directives provided by the module it will be possible:

* + - 1. Disabling the web tracking of all the requests.
      2. Defining a unique identifier for the web server instance (strongly recommended).
      3. Defining for which URIs web tracking is enabled.
      4. Defining which URIs to exclude from those defined in point 3.
      5. Defining one or more request headers whose presence disables the tracking for a single request.
      6. Defining for which values ​​of the Host header the tracking must be enabled.
      7. Defining whether to enable or to disable the tracking of the requests based on the scheme (HTTP or HTTPS).
      8. Defining for which remote IPs or source addresses the tracking must be disabled.
      9. Defining whether to enable the real client IP tracking when there is a proxy in front of the web server.
      10. Defining which header indicates that the real incoming request has an HTTPS scheme although the forwarded request arrived at the web server showed an HTTP scheme (Proxy SSL Offloading).
      11. Defining which headers must be excluded from the request and response records.
      12. Defining which headers won’t report their values in the request and response records.
      13. Defining for which URIs the tracking of the request/response body is enabled/disabled among those defined with the points 3 and 4.
      14. Defining for which URIs the tracking of the request body is disabled when the request method is POST among those derived applying the points 3 and 4.
      15. Defining a size limit for the tracking of the request/response body.
      16. Defining the POST parameters not tracked on the request record.
      17. Defining the file paths where to save the records of the tracking output.
      18. Defining which response headers should be deleted from the real response to the client but preserved on the record saved.
      19. Defining whether to inflate the response deflated with the gzip or deflate algorithm when the tracking record is saved.
      20. Defining which environment Apache variables are to be included as extra headers.
      21. Defining for which uris and token names the users will be extracted and included as extra header.

The module for each request to be tracked injects a header - the name depends on directive WebTrackingUuidHeader - with a unique value, so that all back-end applications can record this value in their application logs, thus ensuring an effective correlation between the tracking data and those applications.

If this header is already present as a request header the value will be held and prefixed with the instance ID – it depends on directive WebTrackingID.

This header can also be used as an indicator that the web tracking is enabled for the incoming request.

# Compatibility Table

|  |  |
| --- | --- |
| **WEB TRACKING APACHE MODULE** | WEB TRACKING COLLECTOR |
| **[1.0.0, 1.0.7]** | [1.0.0, 1.3.8] |
| **[1.1.0, 1.1.3]** | [1.4.0, 1.4.3] |
| **[1.1.4, 1.1.6,**  **2021.9.21.2,**  **2022.3.16.1,**  **2022.4.4.1]** | [1.4.4, 1.5.4,  2021.9.24.1,  2021.9.28.1,  2022.3.16.1,  2022.4.4.1] |
| **[2022.6.21.1, 2023.6.7.1]** | [2022.7.7.1, 2022.7.18.1, 2022.26.1, 2022.9.1.1, 2022.9.14.1, 2023.2.7.1, 2023.6.19.1] |

# Disclaimer

The use of the artifacts present in the distribution packages is allowed only within the perimeter of the IBM WebSphere Application Server Network Deployment license.

The directive WebTrackingPrintWASUser directive requires an additional and explicit agreement with IBM to be used.

Any violation of these rules will be punished according to the local current laws.

# Versions

The version to which the documentation refers is:

**Web Tracking Apache Module 2023.3.1.1 (C17)**

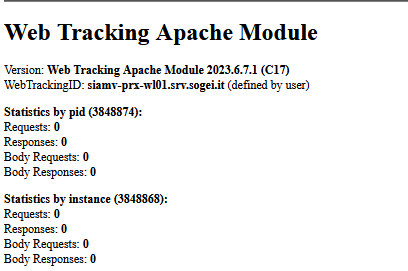
To check the module version, use the command:

strings <Web Tracking .so path> | grep -E "Web Tracking Apache Module [0-9]{4}"

for example:

strings /prod/tracciamento/webtracking/lib/mod\_web\_tracking\_rh8\_24.so | grep -E "Web Tracking Apache Module [0-9]{4}"

The module adds live usage statistics to the server status info.



#### Version history

| **VERSION** | **DATE** | **DESCRIPTION** |
| --- | --- | --- |
| **2023.6.7.1** | 2023-06-07 | Fixed some miscasting and warnings  Moved to GNU Compiler Collection 12.2.1  Fixed lock management for directive WebTrackingRecordFile  Added process mutex along with thread mutex |
| **2023.3.1.1** | 2023-03-01 | Added lock management before writing to WebTrackingRecordFile |
| **2022.6.21.1** | 2022-06-21 | Removed directive WebTrackingRequestFile  Removed directive WebTrackingResponseFile  Removed directive WebTrackingPipesPerInstance  Added directive WebTrackingRecordFile  Changed semantic and syntax of directive WebTrackingID  Fixed method DELETE in order not to enable the input filter  Fixed WebTrackingID evaluation  Removed support for Apache Http Server 2.2  Removed support for Windows Server  Removed support for Red Hat Enterprise Linux 7.x  Removed support for Apache 2.2  Removed support for 32 bit architectures  Moved to GNU Compiler Collection 11.2.1 |
| **2022.4.4.1** | 2022-04-04 | Added directive WebTrackingPipesPerInstance  Moved to Visual Studio 2022 - 17.1.3 |
| **2022.3.16.1** | 2022-03-16 | Moved to Visual Studio 2022 - 17.1.1 |
| **2021.9.21.2** | 2021-09-21 | Changed version pattern  Added check for invalid characters to directive WebTrackingID  Added a stronger check to verify the result of record writes  Added BASE64 NOPAD encoding for instance ID  Moved to GNU Compiler Collection 11.2.0  Moved to Visual Studio 2019 - 16.11.3 |
| **1.1.6** | 2021-02-11 | Fixed input filter when only delay\_print is set  Moved to GNU Compiler Collection 10.2.0  Moved to Visual Studio 2019 - 16.8.5 |
| **1.1.5** | 2020-07-15 | Fixed directive WebTrackingApplicationId  Fixed directive WebTrackingPrintWASUser  Changed version format  Moved to Visual Studio 2019 - 16.6.4 |
| **1.1.4** | 2020-06-17 | Fixed request filter when content-length is missing  Improved request and response filter performances and memory usage  Added request headers tracking to request filter  Added exceeded body limit check to input filter  Fixed regression: POST data are not printed anymore in request access log  Moved to Visual Studio 2019 - 16.6.2 |
| **1.1.3** | 2020-06-08 | Added support for environment variables in directive WebTrackingID  Changed shared memory name: now is prefixed with logs/.shm\_  Fixed the elapsed time calculation for request and response filters  Moved to Visual Studio 2019 - 16.6.1 |
| **1.1.2** | 2020-06-04 | Fixed directive WebTrackingPrintWASUser definition  Fixed directive WebTrackingApplicationId definition  Fixed directive WebTrackingHost to be no case sensitive |
| **1.1.1** | 2020-05-25 | Fixed directive WebTrackingPrintWASUser  Added host filter for directive WebTrackingPrintWASUser  Added host filter for directive WebTrackingApplicationId  Changed UUID header behaviour: it is not generated if already present  Fixed input and output filter  Added directive   * WebTrackingUuidHeader |
| **1.1.0** | 2020-05-13 | Added directive   * WebTrackingPrintRequestHeader   Changed body requests and responses track record  Moved to GNU Compiler Collection 10.1.0  Moved to Visual Studio 2019 - 16.5.5 |
| **1.0.7** | 2020-03-31 | Added directive   * WebTrackingPrintWASUser   Fixed behavior of directive WebTrackingOutputHeader  Fixed version info output  Moved to GNU Compiler Collection 9.3.0  Moved to Visual Studio 2019 - 16.5.1 |
| **1.0.6** | 2019-09-06 | Added directive   * WebTrackingPrintEnvVar   Moved to GNU Compiler Collection 9.2.0 |
| **1.0.5** | 2019-05-15 | Moved to GNU Compiler Collection 9.1.0 |
| **1.0.4** | 2018-11-14 | Added ISO8601 request time stamp for the request and response body records  Modified the access records to print the time stamp in UTC and to include the time zone  Fixed some minor issues |
| **1.0.3** | 2018-09-08 | Rewritten request and response body filters |
| **1.0.2** | 2018-09-03 | Changed the timestamp format  Added the POST parameters to the request access format  Added server status extra content implementation  Added directive   * WebTrackingExcludeFormParameter |
| **1.0.1** | 2018-05-29 | Added directive   * WebTrackingExcludeCookie   Changed the directive WebTrackingID to be no longer mandatory |
| **1.0.0** | 2017-10-16 | Initial release including the following directives (in alphabetical order):   * WebTrackingApplicationId * WebTrackingBodyLimit * WebTrackingClientIpHeader * WebTrackingContentType * WebTrackingDisable * WebTrackingDisablingHeader * WebTrackingEnablePostBody * WebTrackingEnableProxy * WebTrackingExcludeHeader * WebTrackingExcludeHeaderValue * WebTrackingExcludeIP * WebTrackingExcludeURI * WebTrackingExcludeURIBody * WebTrackingExcludeURIPost * WebTrackingHost * WebTrackingHttpEnabled * WebTrackingHttpsEnabled * WebTrackingID * WebTrackingInflateResponse * WebTrackingOutputHeader * WebTrackingRequestFile * WebTrackingResponseFile * WebTrackingSSLIndicator * WebTrackingTraceURI * WebTrackingURI |

# Configuration directives

The following table shows all the directives provided by the web tracking module and the relative syntax (in alphabetical order).

Note: REQ = Required

| **NAME** | **SYNTAX** | **DESCRIPTION** | **EXAMPLE** | **REQ** | **FROM** |
| --- | --- | --- | --- | --- | --- |
| **WebTrackingApplicationId** | String String [String] | It defines an association between a context root or the initial part of a URI and an application ID.  The first string represents the uri prefix and must necessarily start with a slash ('/').  The third string represents a host filter and can be optional – the default value is \*.  The host filter is case insensitive.  It is a multi-line directive.  In case the uri prefix and the host filter are repeated only the first occurrence will be enabled.  If multiple directives are selectable for a single request, the more specific will be selected. | WebTrackingApplicationId /myroot MyApplication | No | 1.0.0  1.1.1 (the host filter) |
| **WebTrackingBodyLimit** | Number | It defines the maximum size in MB that the body can contain when tracked.  The default value is 5 MB.  The range of values ​​is [1, 100] | WebTrackingBodyLimit 10 | No | 1.0.0 |
| **WebTrackinClientIpHeader** | String | Name of the header indicating where to find the real address of the client when a proxy is enabled and put in front of the web server.  The header name is case-insensitive.  The directive is unique for a web server instance and if it is present more than once, only the first one takes effect.  In case it is not defined, the default value is X-Forwarded-For. | WebTrackingClientIpHeader ClientIp | No | 1.0.0 |
| **WebTrackingContentType** | PCRE1 | It defines the Content-Type value for which the request/response body will be recorded.  In cases where the Content-Type header is not present, it is always considered a negative match.  It is a multi-line directive. | WebTrackingContentType html text json  WebTrackingContentType application/x-www-form-urlencoded multipart/form-data | No | 1.0.0 |
| **WebTrackingDisable** | On | **Off** | It disables the web tracking feature for all the requests. | WebTrackingDisable On | No | 1.0.0 |
| **WebTrackingDisablingHeader** | String | Name of the headers that if present in the request disable the web racking feature.  All defined headers must start with  "X-WT".  Header names are case-insensitive.  It is a multi-line directive. | WebTrackingDisablingHeader X-WT-TR-OFF X-WT-TR-NO | No | 1.0.0 |
| **WebTrackingEnablePostBody** | On | **Off** | It enables the tracking of the request body, if any, when the method is POST regardless of the value of the Content Type header.  **Enabling the web tracking feature regardless of the value of the Content-Type header can be a security exposure, so it should be used only if expressly required.** | WebTrackingEnablePostBody On | No | 1.0.0 |
| **WebTrackingEnableProxy** | On | **Off** | It enables the management of the source address as in the presence of a proxy in front of the web server.  The source address becomes the value of the X-Forwarded-For header, or the header specified by the WebTrackingClientIpHeader directive. | WebTrackingEnableProxy On | No | 1.0.0 |
| **WebTrackingExcludeCookie** | String | It defines which cookies will be removed from the request web tracking record (headers cookies and cookie2) and / or the response web tracking record (headers set-cookie and set-cookie2).  It is a multi-line directive. | WebTrackingExcludeCookie JSESSIONID | No | 1.0.1 |
| **WebTrackingExcludeFormParameter** | String | It defines which form parameter will be removed from the POST request web tracking records when the Content Type is application/x-www-form-urlencoded.  To disable the form parameter tracking use the special value “\*”  The character ‘\*’ could be used also as a trailing wildcard.  It is a multi-line directive. | WebTrackingFormParameter j\_password j\_username  WebTrackingFormParameter secure\*  WebTrackingFormParameter \* | No | 1.0.2 |
| **WebTrackingExcludeHeader** | String | It defines which headers will be removed from the request and response web tracking records.  Header names are case-insensitive.  It is a multi-line directive. | WebTrackingExcludeHeader LtpaToken2 | No | 1.0.0 |
| **WebTrackingExcludeHeaderValue** | String | It defines for which headers will be put only the header name on the request and response web tracking records.  Header names are case-insensitive.  It is a multi-line directive. | WebTrackingExcludeHeaderValue Set-Cookie | No | 1.0.0 |
| **WebTrackingExcludeIP** | PCRE1 | It defines the source addresses for which the web tracking is disabled.  It is a multi-line directive. | WebTrackingExcludeIP ^192\.168\.95  WebTrackingExcludeIP ^10\. | No | 1.0.0 |
| **WebTrackingExcludeURI** | PCRE1 | It defines for which URIs among the URIs defined by the WebTrackingURI directive the web tracking is disabled.  It is a multi-line directive. | WebTrackingExcludeURI \.pdf \.jpg  WebTrackingExcludeURI ^/secure/ | No | 1.0.0 |
| **WebTrackingExcludeURIBody** | PCRE1 | It defines for which URIs enabled by other directives is disabled the tracking of the request and response bodies.  It is a multi-line directive. | WebTrackingExcludeURIBody j\_security\_check$ | No | 1.0.0 |
| **WebTrackingExcludeURIPost** | PCRE1 | It defines for which URIs the tracking of the request and response bodies is disabled if the request method is POST.  It is a multi-line directive. | WebTrackingExcludeURIPost /login.jsp$ | No | 1.0.0 |
| **WebTrackingHttpEnabled** | **On** |Off | Flag to enable/disable the web tracking if the scheme is HTTP. | WebTrackingHttpEnabled Off | No | 1.0.0 |
| **WebTrackingHttpsEnabled** | **On** |Off | Flag to enable/disable the web tracking if the scheme is HTTPS. | WebTrackingHttpsEnabled Off | No | 1.0.0 |
| **WebTrackingHost** | PCRE1 | It defines for which Host header values (including port if necessary) the web tracking is enabled.  The regular expression is case insensitive.  It is a multi-line directive. | WebTrackingHost \.agenziaentrate\.gov\.it  WebTrackingHost ^www\. | No | 1.0.0 |
| **WebTrackingID** | String | ID of the web server instance.  It can be defined once for each web server instance.  **The directive was mandatory up to version 1.0.0.**  Since version 1.0.1 it has become optional, because it would be generated by the module.  **It is strongly suggested to define it manually using environment variables.**  Since version 1.1.3 it is possible to use environment variables with the syntax ${<environment variable>}.  The final value will be encoded with BASE64 NOPAD code.  Since version 2022.6.21.1 if the value is prefixed by a exclamation mark the real value won’t be encoded. | WebTrackingID Precomp01 | No | 1.0.0 |
| **WebTrackingInflateResponse** | On |**Off** | Flag to force the inflating of the response body if it has been compressed with the gzip or deflate algorithms. | WebTrackingInflateResponse On | No | 1.0.0 |
| **WebTrackingOutputHeader** | String | It defines the response headers whose value is put in the web tracking record but deleted from the real response to the client.  The header name must have the prefix "X-WT".  Header names are case-insensitive.  It is a multi-line directive. | WebTrackingOutputHeader X-WT-USER | No | 1.0.0 |
| **WebTrackingPrintEnvVar** | String | It defines which Apache environment variables would be put in the web tracking record at the end of the HEADERS part.  Each environment variable will be prefixed with the string “ENV:”  It is a multi-line directive. | WebTrackingPrintEnvVar WAS | No | 1.0.6 |
| **WebTrackingPrintRequestHeader** | String | It defines which request headers would be put in the web tracking record at the end of the (response) HEADERS part.  Each request header will be prefixed with the string “REQ:”.  Be aware that a request header could be not still valid after the completion of the response.  It is a multi-line directive. | WebTrackingPrintRequestHeader Accept | No | 1.1.0 |
| **WebTrackingPrintWASUser [[1]](#footnote-1)** | String String String String [String] | It defines an association between a context root or the initial part of a URI and the data needed to extract the user from an LTPA token.  The first string represents the uri prefix and must necessarily start with a slash ('/').  The second input string is the password for the LTPA keys file.  The third input string is the unescaped LTPA 3DES key property in the LTPA keys file.  The fourth input string is the LTPA token cookie name.  The fifth string represents a host filter and can be optional – the default value is \*.  The host filter is case insensitive.  It is a multi-line directive.  In case the uri prefix and the host filter are repeated only the first occurrence will be enabled.  If multiple directives are selectable for a single request, the more specific will be selected.  The found user will be prefixed with the string “USER:”. | WebTrackingPrintWASUser /wtrest/ WebAS yLTZ1B/rRnEwOivPkWhrp+qZu9tRy1Mr3/cGjIn+mR4= WTSecureToken .mydomain.org | No | 1.0.7  1.1.1 (the host filter) |
| **WebTrackingRecordFile** | Apache Log File | Path of the web tracking file.  It can be expressed as a pipe (strongly recommended) | WebTrackingRecordFile “|/home/liberty/ihs/bin/rotatelogs -p '/home/liberty/webtracking/bin/wt\_collector.sh' -l -f -c /home/liberty/ihs/logs/webtracking-%Y.%m.%d-%H.%M.log 900”  WebTrackingRecordFile "/home/liberty/ihs/logs/webtracking.log" | Sì | 2023.3.1.1 |
| **WebTrackinSSLIndicator** | String | The name of the header indicating that the correct scheme is HTTPS. although the request has been forwarded with the HTTP scheme (SSL Offloading)  The header name is case-insensitive  If defined more than once, only the first directive is enabled. | WebTrackingSSLIndicator SSL-ON | No | 1.0.0 |
| **WebTrackingTraceURI** | PCRE1 | It defines for which URIs the web tracking is enabled for debug purpose.  That directive enables the web tracking for the given URIs independently of the other directives with the only exception of the WebTrackingDisable directive.  It is strongly suggested not to set this directive for production environments.  It is a multi-line directive. | WebTrackingTraceURI ^/test/snoop$ | No | 1.0.0 |
| **WebTrackingURI** | PCRE1 | Define for which URIs the web tracking is enabled.  It is a multi-line directive. | WebTrackingURI /PrecomWeb/.+ | No | 1.0.0 |
| **WebTrackingUuidHeader** | String | The header where the request uuid will be stored.  The default value is X-WT-UUID.  It can be defined once for each web server instance. | WebTrackingUuidHeader X-APP1-UUID | No | 1.1.1 |

Note 1: PCRE = Perl Compatible Regular Expression  
(<http://www.pcre.org>, <http://perldoc.perl.org/perlre.html>).

When the directive value is a PCRE string, the function used for the comparison is the "search" (and not the best known "match").

This choice was made for two fundamental reasons.

1. It is always possible to write a PCRE such that the "search" function works as the "match" function were used, while the opposite would not be possible.
2. With this choice it is easier to write a functional PCRE because it requires fewer characters.

To give an example of the difference between the two functions, here is a comparison table.

|  |  |  |  |
| --- | --- | --- | --- |
| **PCRE** | **URI** | **MATCH** | **SEARCH** |
| **/mycontext** | /mycontext | **OK** | **OK** |
| **/mycontext** | /mycontext/myresource | **KO** | **OK** |
| **/mycontext** | /mypre/mycontext | **KO** | **OK** |
| **^/mycontext$** | /mycontext | **OK** | **OK** |
| **^/mycontext$** | /mycontext/myresource | **KO** | **KO** |
| **^/mycontext$** | /mypre/mycontext | **KO** | **KO** |

From the previous table we understand that for transforming the "search" function in the "match" function it is sufficient to include the PCRE between the characters ^ (caret) and $ (dollar sign).

# Algorithm and record layouts

The module core algorithm is based on the following main points:

1. Reading and analysis of the request to check whether the web tracking should be enabled.
2. Reading and analysis of the request to check whether the tracking of the request and / or the response body should be enabled.
3. Writing the output records to the defined streams.

The fundamental phase for the operations of the module are the points 1 and 2, while what defines the artifacts of the solution are the records written to the streams as described on point 3.

Here are the rules to apply for fulfilling the points 1 and 2 (in order of priority):

1. Check whether the web tracking is enabled as a whole (WebTrackingDisable)
2. Check whether the host (more specifically the value of the Host header) enables the web tracking (WebTrackingHost)
3. Check whether the request URI enables the web tracking (WebTrackingURI)
4. Check whether the request URI disables the web tracking (WebTrackingExcludeURI)
5. Check whether the SSL Offloading header is present among the request headers (WebTrackingSSLIndicator)
6. Check whether the scheme of the request enable the web tracking (WebTrackingHttpsEnabled, WebTrackingHttpEnabled)
7. Check whether one of the request headers disables the web tracking (WebTrackingDisablingHeader)
8. Check whether the real source IP disables the web tracking (WebTrackingExcludeIP)  
   Note: the source IP address is based also on the value of the WebTrackingEnableProxy directive.
9. Check which headers must be removed from the response but written to the web tracking records (WebTrackingOutputHeader).
10. Check for which headers the value must be removed from the web tracking records (WebTrackingExcludeHeaderValue)
11. Check which headers must be removed from the web tracking records (WebTrackingExcludeHeader)
12. Check which cookies present on the headers cookie, cookie2, set-cookie and set-cookie2 must be removed from the web tracking records (WebTrackingExcludeCookie)
13. Check which POST form parameters must be removed for the request web tracking records (WebTrackingExcludeFormParameter)

In addition, the following rules are checked for the tracking of the request and / or response body:

1. Check whether the URI disables the tracking of the request and / or response body (WebTrackingExcludeURIBody)
2. Check whether the URI disables the tracking of the request body if the method is POST (WebTrackingExcludeURIPost)
3. Check whether the request content-type header enables the tracking of the request body and whether the response content-type header enables the tracking of the response body (WebTrackingContentType)
4. Check whether the response size is less or equal to the maximum size defined.   
   If it were greater, the tracking would be disabled (WebTrackingBodyLimit)

The directives that can enable / disable the web tracking are:

* WebTrackingDisable
* WebTrackingExcludeIP
* WebTrackingExcludeURI
* WebTrackingHost
* WebTrackingHttpEnabled
* WebTrackingHttpsEnabled
* WebTrackingTraceURI
* WebTrackingURI

The record layout of the web tracking as follows (between parenthesis the directives that can impact the value of the single field):

* UUID  
  (WebTrackingID if defined)
* Application Id   
  (WebTrackingApplicationId)
* “"\*\*REQUEST\*\*”
* BASE64:
  + Request Timestamp (UTC)
  + Local Time zone (ISO8601)
  + Remote IP   
    (WebTrackingEnableProxy, WebTrackingClientIpHeader)
  + Protocol
  + Method
  + URL
  + Content-Type
  + Bytes Read
  + “HEADERS”
  + Request Headers  
    (WebTrackingExcludeCookie, WebTrackingExcludeHeader, WebTrackingExcludeHeaderValue, WebTrackingPrintEnvVar, WebTrackingPrintWASUser)
  + In case of a POST request whose Content-Type is “application/x-www-form-urlencoded” the final header will be:  
    \*Post=<allowed form parameters in the query\_string format> (that is: <parameter>=<value>{&<parameter>=<value>}  
    (WebTrackingExcludeFormParameter)
* “\*\*REQUEST\_BODY\*\*”
* BAS64(REQUEST BODY) |<EMPTY>
* “"\*\*RESPONSE\*\*”
* BASE64:
  + Local Time zone (ISO8601)
  + Remote IP   
    (WebTrackingEnableProxy, WebTrackingClientIpHeader)
  + Protocol
  + Method
  + URL
  + Status Code
  + Elapsed Time  
    Expressed in microseconds
  + Content-Type
  + Bytes Sent
  + “HEADERS”
  + Response Headers  
    (WebTrackingExcludeCookie, WebTrackingExcludeHeader, WebTrackingExcludeHeaderValue, WebTrackingOutputHeader, WebTrackingPrintEnvVar, WebTrackingPrintRequestHeader, WebTrackingPrintWASUser)
* “\*\*RESPONSE\_BODY\*\*”
* BAS64(RESPONSE BODY) |<EMPTY>

The fields UUID and APPID and every field present in REQUEST and RESPONSE data are included between a pair of double quotes ("); the separator between the various fields is the pipe character (|).

The tracking of the request / response body obviously does not have a defined layout because it depends on the requested resource and can be multi-line.

# Examples

To simplify the administration of the mod\_web\_tracking module and its configuration, it is strongly recommended to add an include directive within the Apache Web Server master configuration file (httpd.conf).

Here is the way to do it:

# Web Tracking Module

Include “conf/webtracking.conf”

A typical configuration:

LoadModule web\_tracking\_module <web tracking home>/lib/mod\_web\_tracking\_rh8\_24.so

WebTrackingID !${HOSTNAME}

WebTrackingDisable Off

WebTrackingHttpEnabled On

WebTrackingHttpsEnabled On

WebTrackingDisablingHeader X-WT-TR-OFF X-WT-TR-NO

WebTrackingOutputHeader X-WT-USER

WebTrackingExcludeHeader Cookie

WebTrackingExcludeHeaderValue Set-Cookie Set-Cookie2

WebTrackingInflateResponse On

WebTrackingApplicationId /PrecomWeb PRECOMP

WebTrackingApplicationid / DEFAULT

WebTrackingPrintEnvVar WAS

WebTrackingRecordFile "|/home/liberty/ihs/bin/rotatelogs -p '/home/liberty/webtracking/bin/wt\_collector.sh' -l -f -c /home/liberty/ihs/logs/webtracking-%Y.%m.%d-%H.%M.log 900"

WebTrackingHost sogei-dev

WebTrackingURI ^/PrecomWeb/.+$ \.html$ post$

WebTrackingExcludeURI \.pdf$ \.jpg$

WebTrackingExcludeIP ^192\.168\.10\.

WebTrackingExcludeURIBody j\_security\_check$

WebTrackingExcludeURIPost /login.jsp$

WebTrackingContentType html json text

WebTrackingContentType application/x-www-form-urlencoded

Here is an example of configuration in which the tracking of the request and the response bodies is disabled (the trick is to do not define the WebTrackingContentType directive at all):

LoadModule web\_tracking\_module <web tracking home>/lib/mod\_web\_tracking\_rh8\_24.so

WebTrackingID ${HOSTNAME}

WebTrackingDisable Off

WebTrackingHttpEnabled On

WebTrackingHttpsEnabled On

WebTrackingDisablingHeader X-WT-TR-OFF X-WT-TR-NO

WebTrackingOutputHeader X-WT-USER

WebTrackingInflateResponse On

WebTrackingEnableProxy On

WebTrackingClientIpHeader ClientIp

WebTrackingBodyLimit 10

WebTrackingApplicationId /PrecomWeb PRECOMP

WebTrackingApplicationid / DEFAULT

WebTrackingRecordFile "logs/webtracking.log"

WebTrackingHost sogei-dev

WebTrackingURI ^/PrecomWeb/.+$ \.html$ post$

WebTrackingExcludeURI \.pdf$ \.jpg$

To check whether the module has been loaded correctly, set the LogLevel directive to the info level.

LogLevel info

If the module has been loaded correctly the error file will contain a line like:

Web Tracking Apache Module <Version> (<Language Specifications>)

To debug the module, set the LogLevel directive to the debug level.

LogLevel debug

The debug information can be found in the error file.

1. See Disclaimer before using the directive [↑](#footnote-ref-1)