# WEB TRACKING APACHE MODULE

## Version 2025.2.18.1

---

Author: Andrea Minuto (andrea.minuto@it.ibm.com)
IBM Cloud Integration Expert Labs Europe
IBM Italia S.p.A.
Date: February 19, 2025

# Contents

# 1. 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 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.

The source code, not part of the solution, is written in compliance with the C17/C++23 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 14.2.1 20240801 (Red Hat 14.2.1-1).

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.

# 2. Features

The module is called Web Tracking Apache Module (web_tracking) and the shared library, that is the executable associated with the solution, has the name: mod_web_tracking.so

The distribution package is a compressed file: webtracking-bin.zip.

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 reverse proxy in front of the web server.
10. Defining which header indicates that the real incoming request has an HTTPS scheme though the forwarded request arrived at the web server shows an HTTP scheme (Proxy SSL Offloading).
11. Defining which headers must be excluded from the request and/or response.
12. Defining which headers won't report their values in the request and/or response.
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 folder path where to save the tracking data files.
18. Defining which response headers should be deleted from the response while preserved on the tracking data.
19. Defining whether to inflate the response when deflated with gzip before saving it to the tracking data.
20. Defining which environment Apache variables are to be included as extra headers.

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 will be increment last character.

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

# 3. Version

The version to which the documentation refers is:

**Web Tracking Apache Module 2025.2.18.1 (C17/C++23)**

To check the module version, use the command:

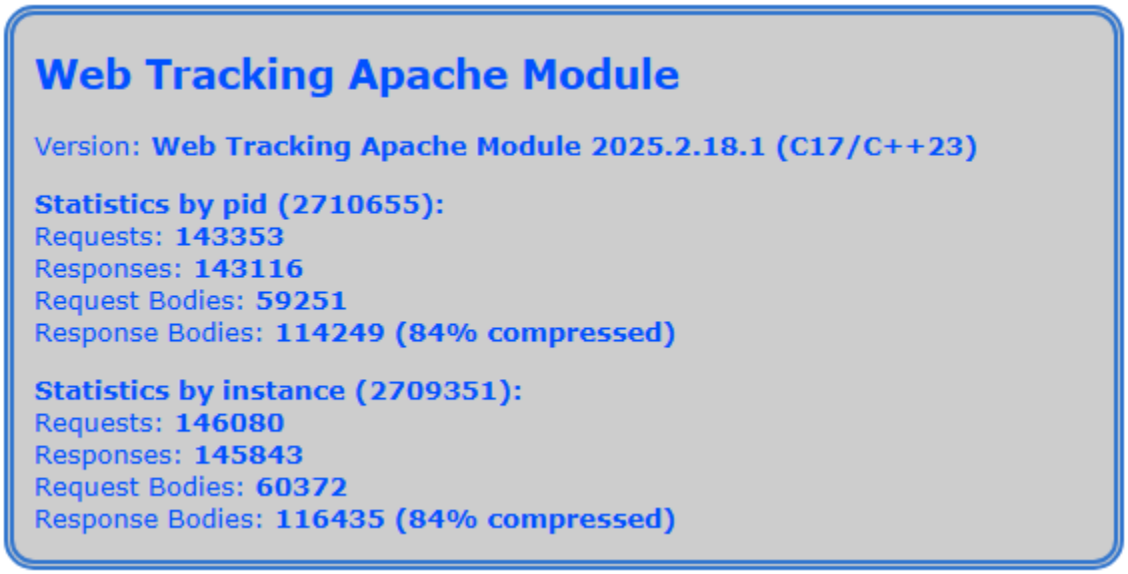
```
strings <Web Tracking .so path> | grep -E -o 'Web Tracking Apache
Module .*?\)'
```

for example:

```
strings /prod/webtracking/lib/mod_web_tracking.so| grep -E -o 'Web
Tracking Apache Module .*?\)'
```

The module current version is written on the error log file – directive ErrorLog – just after the start of a web server instance.

The module adds live usage statistics to the server status info (uri: /server-status).



## Version history

The versions with a tag "[R<year>.<sequence>]" are to be considered releases and ready to be deployed in a production environment.

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **2025.2.18.1 [R2025.2]** | 2025-02-18 | Remove output headers from response body<br>Fix memory allocations to remove leaks<br>Enhance file management to reduce its overhead<br>Change uuid algorithm<br>Remove directive WebTrackingID<br>Fix encoding POST query string as “*Post” header |
| **2025.2.10.2 [R2025.1]** | 2025-02-10 | Implement request/responce cycle functions using C++23<br>Implement record file management in C++23<br>Change tracking data record format and contents<br>Change requirements for directives WebTrackingDisablingHeader and WebTrackingOutputHeader<br>Add styling to server status hook<br>Implement hot debug for specific resources<br>Implement some runtime optimizations and some code enhancements<br>Remove directive WebTrackingPrintWASUser<br>Remove directive WebTrackingPrintRequestHeader<br>Move to GNU Compiler Collection 14.2.1 |
| **2025.1.15.1** | 2025-01-15 | Move configuration directives printing out from DEBUG to INFO |
| **2025.1.14.1** | 2024-01-14 | Change WebTrackingBodyLimit meaning and implement it<br>The body limit is also compared to inflated bodies |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **2025.1.9.1** | 2025-01-09 | Simplify algorithm to move current record file |
| **2024.12.20.1** | 2024-12-20 | Change algorithm to copy and delete the current record file |
| **2024.5.29.1** | 2024-05-29 | Fix child exit operations<br>Move to GNU Compiler Collection 14.1.0 |
| **2024.5.28.1** | 2024-05-28 | Add copying and removing record file off-line |
| **2024.5.21.1** | 2024-05-21 | Add directive WebTrackingRecordFolder<br>Add directive WebTrackingRecordArchiveFolder<br>Add directive WebTrackingRecordLifeTime<br>Remove directive WebTrackingRecordFile |
| **2024.1.9.1** | 2024-01-09 | Swapped lock cross-processes and cross-threads management |
| **2023.9.26.1** | 2023-09-26 | Added directive WebTrackingApplicationIdFromHeader<br>Fixed log record writing |
| **2023.9.12.1** | 2023-09-12 | Added logging timestamp to record<br>Moved to GNU Compiler Collection 13.2.0 |
| **2023.6.7.1** | 2023-06-07 | Fixed some miscasting and warnings<br>Moved to GNU Compiler Collection 12.2.1<br>Fixed lock management for directive WebTrackingRecordFile<br>Added process mutex along with thread mutex |
| **2023.3.1.1** | 2023-03-01 | Added lock management before writing to WebTrackingRecordFile |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **2022.6.21.1** | 2022-06-21 | Removed directive WebTrackingRequestFile<br>Removed directive WebTrackingResponseFile<br>Removed directive WebTrackingPipesPerInstance<br>Added directive WebTrackingRecordFile<br>Changed semantic and syntax of directive WebTrackingID<br>Fixed method DELETE in order not to enable the input filter<br>Fixed WebTrackingID evaluation<br>Removed support for Apache Http Server 2.2<br>Removed support for Windows Server<br>Removed support for Red Hat Enterprise Linux 7.x<br>Removed support for Apache 2.2<br>Removed support for 32-bit architectures<br>Moved to GNU Compiler Collection 11.2.1 |
| **2022.4.4.1** | 2022-04-04 | Added directive WebTrackingPipesPerInstance<br>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<br>Added check for invalid characters to directive WebTrackingID<br>Added a stronger check to verify the result of record writes<br>Added BASE64 NOPAD encoding for instance ID<br>Moved to GNU Compiler Collection 11.2.0<br>Moved to Visual Studio 2019 - 16.11.3 |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **1.1.6** | 2021-02-11 | Fixed input filter when only delay_print is set<br>Moved to GNU Compiler Collection 10.2.0<br>Moved to Visual Studio 2019 - 16.8.5 |
| **1.1.5** | 2020-07-15 | Fixed directive WebTrackingApplicationId<br>Fixed directive WebTrackingPrintWASUser<br>Changed version format<br>Moved to Visual Studio 2019 - 16.6.4 |
| **1.1.4** | 2020-06-17 | Fixed request filter when content-length is missing<br>Improved request and response filter performances and memory usage<br>Added request headers tracking to request filter<br>Added exceeded body limit check to input filter<br>Fixed regression: POST data are not printed anymore in request access log<br>Moved to Visual Studio 2019 - 16.6.2 |
| **1.1.3** | 2020-06-08 | Added support for environment variables in directive WebTrackingID<br>Changed shared memory name: now is prefixed with logs/.shm_<br>Fixed the elapsed time calculation for request and response filters<br>Moved to Visual Studio 2019 - 16.6.1 |
| **1.1.2** | 2020-06-04 | Fixed directive WebTrackingPrintWASUser definition<br>Fixed directive WebTrackingApplicationId definition<br>Fixed directive WebTrackingHost to be no case sensitive |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **1.1.1** | 2020-05-25 | Fixed directive WebTrackingPrintWASUser<br>Added host filter for directive WebTrackingPrintWASUser<br>Added host filter for directive WebTrackingApplicationId<br>Changed UUID header behavior: it is not generated if already present<br>Fixed input and output filter<br>Added directive<br>➢ WebTrackingUuidHeader |
| **1.1.0** | 2020-05-13 | Added directive<br>➢ WebTrackingPrintRequestHeader<br>Changed body requests and responses track record<br>Moved to GNU Compiler Collection 10.1.0<br>Moved to Visual Studio 2019 - 16.5.5 |
| **1.0.7** | 2020-03-31 | Added directive<br>➢ WebTrackingPrintWASUser<br>Fixed behavior of directive WebTrackingOutputHeader<br>Fixed version info output<br>Moved to GNU Compiler Collection 9.3.0<br>Moved to Visual Studio 2019 - 16.5.1 |
| **1.0.6** | 2019-09-06 | Added directive<br>➢ WebTrackingPrintEnvVar<br>Moved to GNU Compiler Collection 9.2.0 |
| **1.0.5** | 2019-05-15 | Moved to GNU Compiler Collection 9.1.0 |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **1.0.4** | 2018-11-14 | Added ISO8601 request time stamp for the request and response body records<br><br>Modified the access records to print the time stamp in UTC and to include the time zone<br><br>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<br><br>Added the POST parameters to the request access format<br><br>Added server status extra content implementation<br><br>Added directive<br><br>➢ WebTrackingExcludeFormParameter |
| **1.0.1** | 2018-05-29 | Added directive<br><br>➢ WebTrackingExcludeCookie<br><br>Changed the directive WebTrackingID to be no longer mandatory |

| VERSION | DATE | DESCRIPTION |
|---|---|---|
| **1.0.0** | 2017-10-16 | Initial release including the following directives (in alphabetical order):<br>➢ WebTrackingApplicationId<br>➢ WebTrackingBodyLimit<br>➢ WebTrackingClientIpHeader<br>➢ WebTrackingContentType<br>➢ WebTrackingDisable<br>➢ WebTrackingDisablingHeader<br>➢ WebTrackingEnablePostBody<br>➢ WebTrackingEnableProxy<br>➢ WebTrackingExcludeHeader<br>➢ WebTrackingExcludeHeaderValue<br>➢ WebTrackingExcludeIP<br>➢ WebTrackingExcludeURI<br>➢ WebTrackingExcludeURIBody<br>➢ WebTrackingExcludeURIPost<br>➢ WebTrackingHost<br>➢ WebTrackingHttpEnabled<br>➢ WebTrackingHttpsEnabled<br>➢ WebTrackingID<br>➢ WebTrackingInflateResponse<br>➢ WebTrackingOutputHeader<br>➢ WebTrackingRequestFile<br>➢ WebTrackingResponseFile<br>➢ WebTrackingSSLIndicator<br>➢ WebTrackingTraceURI<br>➢ WebTrackingURI |

# 4. 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.<br><br>The first string represents the uri prefix and must necessarily start with a slash ('/').<br><br>The third string represents a host filter and can be optional – the default value is *.<br><br>The host filter is case insensitive.<br><br>It is a multi-line directive.<br><br>In case the uri prefix and the host filter are repeated only the first | WebTrackingApplicationId /myroot MyApplication | No | 1.0.0<br><br>1.1.1 (the host filter) |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | occurrence will be enabled.<br><br>If multiple directives are selectable for a single request, the more specific will be selected. | | | |
| **WebTrackingApplicationIdFromHeader** | String | Defines which response header sets the application id value for the current request.<br><br>It can only be defined once within the directive file. | WebTrackingApplicationIdFromHeader application-id | No | 2023.9.26.1 |
| **WebTrackingBodyLimit** | Number | It defines the maximum size in MB that the body can contain when tracked.<br><br>The default value is 5 MB.<br><br>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 | WebTrackingClientIpHeader ClientIp | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | the web server.<br><br>The header name is case-insensitive.<br><br>The directive is unique for a web server instance and if it is present more than once, only the first one takes effect.<br><br>In case it is not defined, the default value is X-Forwarded-For. | | | |
| **WebTrackingContentType** | PCRE[1] | It defines the Content-Type value for which the request/response body will be recorded.<br><br>In cases where the Content-Type header is not present, it is always considered a negative match.<br><br>It is a multi-line directive. | WebTrackingContentType html text json<br><br>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 | WebTrackingDisablingHeader X-WT-TR-OFF X-WT-TR-NO | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | request disable the web racking feature.<br><br>**All defined headers must start with "X-WT-" or "WT-" (since version 2025.2.5.1).**<br><br>Header names are case-insensitive.<br><br>It is a multi-line directive. | | | |
| **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.<br><br>**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 | WebTrackingEnableProxy On | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | of a proxy in front of the web server.<br><br>The source address becomes the value of the X-Forwarded-For header, or the header specified by the WebTrackingClientIpHeader directive. | | | |
| **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).<br><br>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- | WebTrackingFormParameter j_password j_username<br><br>WebTrackingFormParameter secure*<br><br>WebTrackingFormParameter * | No | 1.0.2 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | urlencoded.<br><br>To disable the form parameter tracking use the special value "*"<br><br>The character '*' could be used also as a trailing wildcard.<br><br>It is a multi-line directive. | | | |
| **WebTrackingExcludeHeader** | String | It defines which headers will be removed from the request and response web tracking records.<br><br>Header names are case-insensitive.<br><br>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.<br><br>Header names are case-insensitive.<br><br>It is a multi-line directive. | WebTrackingExcludeHeaderValue Set-Cookie | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| **WebTrackingExcludeIP** | PCRE[1] | It defines the source addresses for which the web tracking is disabled.<br><br>It is a multi-line directive. | WebTrackingExcludeIP ^192\.168\.95<br><br>WebTrackingExcludeIP ^10\. | No | 1.0.0 |
| **WebTrackingExcludeURI** | PCRE[1] | It defines for which URIs among the URIs defined by the WebTrackingURI directive the web tracking is disabled.<br><br>It is a multi-line directive. | WebTrackingExcludeURI \.pdf \.jpg<br><br>WebTrackingExcludeURI ^/secure/ | No | 1.0.0 |
| **WebTrackingExcludeURIBody** | PCRE[1] | It defines for which URIs enabled by other directives is disabled the tracking of the request and response bodies.<br><br>It is a multi-line directive. | WebTrackingExcludeURIBody j_security_check$ | No | 1.0.0 |
| **WebTrackingExcludeURIPost** | PCRE[1] | It defines for which URIs the tracking of the request and response bodies is disabled if the request method is POST.<br><br>It is a multi-line directive. | WebTrackingExcludeURIPost /login.jsp$ | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| **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** | PCRE[1] | It defines for which Host header values (including port if necessary) the web tracking is enabled.<br><br>The regular expression is case insensitive.<br><br>It is a multi-line directive. | WebTrackingHost \.agenziaentrate\.gov\.it<br><br>WebTrackingHost ^www\. | No | 1.0.0 |
| **WebTrackingInflateResponse** | On \| **Off** | Flag to force the inflating of the response body if it has been compressed with the gzip 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. | WebTrackingOutputHeader X-WT-USER | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | **The header name must have the prefix "X-WT" or "WT-" (since version 2025.2.5.1).**<br><br>Header names are case-insensitive.<br><br>It is a multi-line directive. | | | |
| **WebTrackingPrintEnvVar** | String | It defines which Apache environment variables would be put in the web tracking record at the end of the HEADERS part.<br><br>Each environment variable will be prefixed with the string "ENV:"<br><br>It is a multi-line directive. | WebTrackingPrintEnvVar WAS | No | 1.0.6 |
| **WebTrackingRecordArchiveFolder** | Path | Path of the web tracking folder where to archive tracking data files.<br><br>If not defined it will be defaulted to WebTrackingRecordFolder/archives.<br><br>**Warning**: **If the** | WebTrackingRecordArchiveFolder /prod/webtracking/splunk | No | 2024.5.21.1 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | **value is equal, as string, to the directive WebTrackingRecordFolder, the move of the record files, right after their closure, is disabled. So, mind the file system free space in such a case.** | | | |
| **WebTrackingRecordFolder** | Path | Path of the web tracking folder where to save tracking data files.<br><br>If not defined it will be defaulted to the current directory for the apache web server instance. | WebTrackingRecordFolder /prod/IBM/HTTPServer/logs | No | 2024.5.21.1 |
| **WebTrackingRecordLifeTime** | Number | Defines the time a single tracking data file must accept new records.<br><br>It should be in the range [5, 120] and is expressed in minutes.<br><br>The default value is 30.<br><br>**WARNING: A tracking data file will be closed when** | WebTrackingRecordLifeTime 15 | No | 2024.5.21.1 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | **the size is greater than 1 GB, regardless of the time interval that has already passed.** | | | |
| **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)<br><br>The header name is case-insensitive<br><br>If defined more than once, only the first directive is enabled. | WebTrackingSSLIndicator SSL-ON | No | 1.0.0 |
| **WebTrackingTraceURI** | PCRE[1] | It defines for which URIs the web tracking is enabled for debug purpose.<br><br>That directive enables the web tracking for the given URIs independently of the other directives with the only exception of the | WebTrackingTraceURI ^/test/snoop$ | No | 1.0.0 |

| NAME | SYNTAX | DESCRIPTION | EXAMPLE | REQ | FROM |
|---|---|---|---|---|---|
| | | WebTrackingDisable directive.<br><br>**It is strongly suggested not to set this directive for production environments.**<br><br>It is a multi-line directive. | | | |
| **WebTrackingURI** | PCRE[1] | Define for which URIs the web tracking is enabled.<br><br>It is a multi-line directive. | WebTrackingURI /PrecomWeb/.+ | No | 1.0.0 |
| **WebTrackingUuidHeader** | String | The header where the request uuid will be stored.<br><br>The default value is X-WT-UUID.<br><br>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).

# 5. 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 record to the defined stream.

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 record written to the stream 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 and cookie, set-cookie must be removed from the web tracking records (WebTrackingExcludeCookie)
13. Check which POST form parameters must be removed for the request web tracking record (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 (the directives that can impact the value of the single field in round brackets) [examples of values in square brackets]:

- Timestamp
  [2025-01-28 10:46:57.618 CET]
- Web Server Hostname
- [siamv-prx-wl01.srv.sogei.it]
- UUID[5]
  [5]The field UUID must be unique overall.
  For the web_tracking module is a string of 65 characters, the first 64 is a sha256 hash value of a unique string, last character is numeric and is the number of times the same UUID is injected to a request – 0 means is the origin request.
  (WebTrackingUuidHeader)
  [33a0cf36f18ce6bf45feb4aab74586665bc73248363387334e8cceaec3b8acce0]
- Application Id
  (WebTrackingApplicationId, WebTrackingApplicationIdFromHeader)
  [E_0168_ENT-SA-0323]
- "****REQUEST****"
- Request Timestamp
  [2025-01-28 10:46:57.618 CET]
- Remote IP
  (WebTrackingEnableProxy, WebTrackingClientIpHeader)
  [26.0.198.115]

- Protocol
  [HTTP/1.1]
- Method
  [POST]
- URL
  [https://scrivania.agenziaentrate.it/scrivania-int/scrivania]
- "**HEADERS**"
- Request Headers
  (WebTrackingExcludeCookie, WebTrackingExcludeHeader, WebTrackingExcludeHeaderValue, WebTrackingPrintEnvVar, WebTrackingExcludeFormParameter)
  [Host: sd20.agenziaentrate.it]
  [PrivateRequestHeader]
  [*Post: domain=.agenziaentrate.it&tipo=23][1]
  [1] In case of a method POST whose Content-Type is "application/x-www-form-urlencoded" and the URI is not demanded to be excluded. The value is url encoded. (WebTrackingExcludeURIPost)
- "****REQUEST_BODY****"[2]
- BAS64(REQUEST BODY)[2]
  (WebTrackingBodyLimit, WebTrackingEnablePostBody, WebTrackingExcludeURIBody, WebTrackingExcludeURIPost, WebTrackingContentType)
  [2] Optional (both fields are either present or missing)
- "****RESPONSE****"
- Status Code
  [200]
- Elapsed Time
  [78361][3]
  [3] Expressed in microseconds
- Elapsed Time
  [78.361 ms]
- Bytes Read
  [12834]
- Bytes Sent
  [1275381]
- "**HEADERS**"
- Response Headers
  (WebTrackingExcludeCookie, WebTrackingExcludeHeader, WebTrackingExcludeHeaderValue, WebTrackingOutputHeader, WebTrackingPrintEnvVar)
  [Content-Type: text/html]

[PrivateResponseHeader]
[ENV: WAS=siamv-prx-al01.srv.sogei.it:9101]

- "****RESPONSE_BODY****"[4]
- BAS64(RESPONSE BODY)[4]
  (WebTrackingBodyLimit, WebTrackingExcludeURIBody, WebTrackingContentType)

[4] Optional (both fields are either present or missing)

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 content of the request and response bodies obviously does not have a defined layout because it depends on the requested resource. Anyway, they are stored BASE64 encoded.

# 6. Examples

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

Here is the way to do it:

```
# Web Tracking Module
Include "conf/webtracking.conf"
```

A typical configuration file could be:

```
# Load module web_tracking
LoadModule web_tracking_module /prod/webtracking/lib/mod_web_tracking.so

# Set log level for module web_tracking
LogLevel web_tracking:info

# Web Tracking Header
WebTrackingUuidHeader X-WT-UUID

# Application Id
WebTrackingApplicationIdFromHeader application-id
WebTrackingApplicationId / WEBTRACKING

# Web Tracking Directives
WebTrackingHost \.agenziaentrate\.it
WebTrackingEnablePostBody Off
WebTrackingURI ^/
WebTrackingExcludeURI \.pdf$ \.jpg$ \.css$ \.png$ \.js$ \.gif$ \.ico$
loginPage.jsp$ \.eot$ \.woff$ \.woff2 \.map$ \.ttf$
WebTrackingExcludeURI ^/server-status/
WebTrackingContentType html json text\/(?!csv)
WebTrackingContentType application/x-www-form-urlencoded

WebTrackingInflateResponse On
```

```
WebTrackingDisablingHeader X-WT-OFF
WebTrackingOutputHeader X-WT-USER X-WT-ID-SESSION
WebTrackingOutputHeader X-WT-CAMPI-LIBERI
WebTrackingOutputHeader X-WT-IP-APP-SERVER X-WT-HOSTNAME-APP-SERVER X-WT-APP-
SERVER-PORT X-WT-SERVER-ENCODING
WebTrackingEnableProxy On
WebTrackingClientIpHeader X-Forwarded-For

# WebTracking File Directives
WebTrackingRecordFolder /prod/IBM/HTTPServer/logs
WebTrackingRecordArchiveFolder /prod/tracciamento/webtracking/splunk
WebTrackingRecordLifeTime 15
```

To disable the tracking of the request and the response bodies do not define any WebTrackingContentType directives and set WebTrackingEnablePostBody to Off.

If the module has been loaded correctly the error file should contain a line with the module version:

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

To define the log level, you must use the Apache Web Server directive:
`LogLevel web_tracking:<level>`

The level can be: warn, info (recommended), debug.

An upgrade/deployment procedure (strongly recommended) can be:

1. Stop all IHS/Apache Web Server instances that use the web_tracking module.
2. Move all WebTackingRecordFolder/webtracking*.log files to the WebTrackingRecordArchiveFolder directory
3. Remove files in /prod/webtracking/lib directory
4. Unzip the installation package to the /prod directory

Example script:

```
/prod/IBM/HTTPServer/bin/apachectl stop
mv -v /prod/IBM/HTTPServer/logs/webtracking*.log /prod/webtracking/splunk
rm -fv /prod/webtracking/lib/*
unzip -uo ~/webtracking-bin.zip -d /prod/
```

# 7. Troubleshooting

**Metrics**

If the log level for the module web_tracking is at least set to info, for each tracked request will be written a log record on the web server error log file – directive ErrorLog.

The format for the metrics record log is:

```
[WT-METRICS: <uuid> | <appid> | <uri> | <status code> | <module overhead for request> | <if request body is present>REQUEST<else>NO | <if response body is present>RESPONSE<else>NO | <if the record is successfully written to file>#written-bytes<else>KO | <elapsed time to write to file>]
```

Sample of metrics record log:

```
[Wed Feb 05 17:36:50.248970 2025] [web_tracking:info] [pid 3819381:tid 140265348957952] [WT-METRICS: siamv-prx-wl01.srv.sogei.it:Z6OToQuMcAc4W-gG8aTQ9wAAAeE | SCRIVANIA.INT | /scrivania-int/scrivania | 200 | 934 us | NO | RESPONSE| 7815 | 57 us]
```

**Hot Debug**

It is possible to enable the debug for specific URIs or group of.

It doesn't need to restart the involved web server instances because the web_tracking module is able to read at runtime for what resources must be enabled the debug.

As always, the debug log records will be written on error file as configured by standard IBM HTTP Server or Apache Web Server directives.

The URI to be debugged must be written in a file whose path is: **/tmp/webtracking_debug_uris**.

Each line not staring with the character pound ('#') specifies the URI prefix to be debugged.

Example:

```
# Territorio
https://sitnew.agenziaentrate.it/sit2/public/appnav/index.php/istanzepost/associaplanim

# Scrivania
https://scrivania.agenziaentrate.it/scrivania-rest/
```

**Crontab**

Due to the internal mechanisms of the Apache Web Server / IHS, it may happen that some files with tracking data are not moved from the WebTrackingRecordFolder folder to the WebTrackingRecordArchiveFolder folder.

To prevent these files from not being processed and therefore removed, the suggestion is to activate a script on the user's crontab with which the web server process runs – User directive – which moves the files not moved yet automatically.

An example can be:

```
# record file watchdog
0 9-20 * * * find /prod/IBM/HTTPServer/logs/ -name "webtracking*.log" -type f -mmin +60 -exec mv {} /prod/webtracking/splunk/ \;
```

**Incidents**

In case someone reports an incident where the web_tracking module is either involved or supposed to be, the following procedure must be put into action:

1. Retrieve the URL that experiences the reported issue.
2. Enable the hot debug for that URL, adding it to the file /tmp/webtracking_debug_uris.
3. Once the debug log records have been collected, remove that URL from hot debug and temporarily exclude it via the directive WebTrackingExcludeURI.
4. When the incident will be solved or claimed as a non-error, re-enable the no longer reported URL.

This procedure must be performed for all URLs reported with a problem.

**WARNING: In case is reported either a CPU or memory issue, disable the web_tracking module as soon as possible and collect metrics data from web server error logs.**