WEB TRACKING APACHE MODULE

Version 2025.5.27.1

Author: Andrea Minuto (andrea.minuto@it.ibm.com)
IBM Cloud Integration Expert Labs Europe
IBM Italia S.p.A.
Date: June 12, 2025

Contents

1. Overview

Web Tracking Apache Module

The Web Tracking Apache Module is designed for the Apache Web Server 2.4.x (IBM HTTP Server 9.0.x) 64-bit. It supports platforms such as Red Hat Enterprise Linux 8.x, 9.x, and later versions.

Main Functionality

The primary function of this module is to track the input (requests) and output (responses) of HTTP/HTTPS roundtrips within an Apache Web Server.

WARNING: The supported protocol version is HTTP/1.1. Requests using different protocol versions will not be tracked.

Dual Purpose

The module serves two main purposes:

1. **Legal and Security Control:** It tracks all HTTP transactions for legal and security control purposes.
2. **Debugging:** It helps in debugging specific web transactions that exhibit anomalies or whose behavior is not fully understood.

Source Code and Compatibility

The source code, which is not part of the solution, adheres to the C17/C++23 specifications. It is compatible with all distributed platforms that support the IBM HTTP Server. The module is compiled using gcc version 14.2.1 20250110 (Red Hat 14.2.1-7).

Compression Algorithms

The module is based on a proprietary extension of the open-source library zlib V1.2.11 (https://www.zlib.net/). This library is fully included and compiled within the module for portability and security. The supported compression algorithms are gzip and deflate.

Security Management

The module itself does not manage security issues or intercept web threats. These tasks are overseen by web server administrators and other specialized modules.

2. Requirements

Web server machines requirements

The most important requirements are:

- **The owner of the web server processes must not be *root***.
- **The record folders must be inside the same Linux filesystem** – preferably ext4 type.

All the other kinds of requirements depend on *load number*, that is the number of hits for hour to be tracked, and are described via tables.

The storage size requirement table

Load number	Storage size
> 15,000,000	100 GB
> 12,500,000	80 GB
> 10,000,000	60 GB
> 7,500,000	40 GB
> 5,000,000	25 GB
-	15 GB

WARNING: It is strongly suggested that in case of a system with a load number greater than 10 million, the record log folders are configured to a separated and isolated file system.
Such a setting is necessary to avoid temporarily unavailability of the service to be tracked until the record files will be either processed or manually removed.

The RAM size requirement table

Load number	RAM
> 15,000,000	20 GB
> 12,500,000	16 GB
> 10,000,000	12 GB
> 7,500,000	8 GB
> 5,000,000	6 GB
-	4 GB

The CPU – number of sockets – requirement table

Load number	Sockets
> 15,000,000	16
> 12,500,000	12
> 10,000,000	8
> 7,500,000	6
> 5,000,000	4
-	2

Apache Web Server / IBM HTTP Server

The configuration of MPM directives is very important to avoid instability of web server processes.

If you use the mpm_event module, you don't need to change anything in particular.

If, on the other hand, the module is the most classic mpm_worker, the associated directives must be configured to limit the generation and closure of processes as much as possible.

An example configuration is as follows:

```
# ThreadLimit: maximum setting of ThreadsPerChild
# ServerLimit: maximum setting of StartServers
# StartServers: initial number of server processes to start
# MaxClients: maximum number of simultaneous client connections
# MinSpareThreads: minimum number of worker threads which are kept spare
# MaxSpareThreads: maximum number of worker threads which are kept spare
# ThreadsPerChild: constant number of worker threads in each server process
# MaxRequestsPerChild: maximum number of requests a server process serves
ThreadLimit          400
# After 9.0.0.3, it's important for the event MPM to have some slack space for
ServerLimit
ServerLimit            4
StartServers           1
MaxClients          1600
MinSpareThreads       40
# PI74200: When using the event MPM, discourage process termination during
runtime.
MaxSpareThreads      540
ThreadsPerChild      400
MaxRequestWorkers   1600
MaxRequestsPerChild    0
ListenBacklog       2048
MaxMemFree          4096
```

Splunk Forwarder requirements

The requirements are beyond the scope of this guide.

For such information, refer to other and more specific documents.

3. Features

Web Tracking Apache Module (web_tracking)

The Web Tracking Apache Module (web_tracking) is a shared library, with the executable named mod_web_tracking.so. The distribution package is a compressed file named webtracking-bin.zip.

Configuration Directives

The module provides various configuration directives that allow you to:

1. **Disable Web Tracking:** Disable tracking for all requests.
2. **Unique Identifier:** Define a unique identifier for the web server instance (strongly recommended).
3. **Enable Tracking for Specific URIs:** Specify which URIs should have web tracking enabled.
4. **Exclude Specific URIs:** Define URIs to exclude from tracking, even if they are included in the enabled list.
5. **Disable Tracking Based on Headers:** Specify request headers whose presence will disable tracking for individual requests.
6. **Host Header Values:** Define values of the host header for which tracking must be enabled.
7. **Scheme-Based Tracking:** Enable or disable tracking based on the scheme (HTTP or HTTPS).
8. **Remote IPs:** Specify remote IPs or source addresses for which tracking should be disabled.
9. **Real Client IP Tracking:** Enable tracking of the real client IP when a reverse proxy is in front of the web server.
10. **Proxy SSL Offloading:** Define headers indicating that the real incoming request has an HTTPS scheme, even if the forwarded request shows an HTTP scheme.
11. **Exclude Headers:** Specify headers to exclude from the request and/or response.
12. **Non-Reporting Headers:** Define headers that should not report their values in the request and/or response.
13. **Body Tracking for URIs:** Enable or disable tracking of the request/response body for specific URIs.

14. **Size Limit:** Define a size limit for tracking the request/response body.
15. **POST Parameters:** Specify POST parameters that should not be tracked in the request record.
16. **Data File Path:** Define the folder path where tracking data files should be saved.
17. **Response Headers:** Specify response headers to be deleted from the response while preserving them in the tracking data.
18. **Inflate Response:** Enable inflating the response when deflated with gzip before saving it to the tracking data.
19. **Apache Environment Variables:** Define which Apache environment variables should be included as extra headers.

Unique Tracking Header

For each request, a header is injected—its name depends on the directive WebTrackingUuidHeader—with a unique value. This allows all back-end applications to record this value in their application logs to correlate the tracking data with the application logs. If this header is already present in the incoming request, the value will be retained, and only the last character will be incremented in a circular way (‘1’ - ‘9’, ‘A’ – ‘Z’, ‘a’ – ‘z’). The first time, the last character will always be ‘0’.

Sentinel Header

There is also a sentinel header, x-wt-request-to-be-tracked, which can be used as an indicator that web tracking is enabled for the current request. The value of this header does not matter; its presence is sufficient.

4. Version

The version to which the documentation refers is:

Web Tracking Apache Module 2025.5.27.1 (C17/C++23)

To check the module version, use the command:

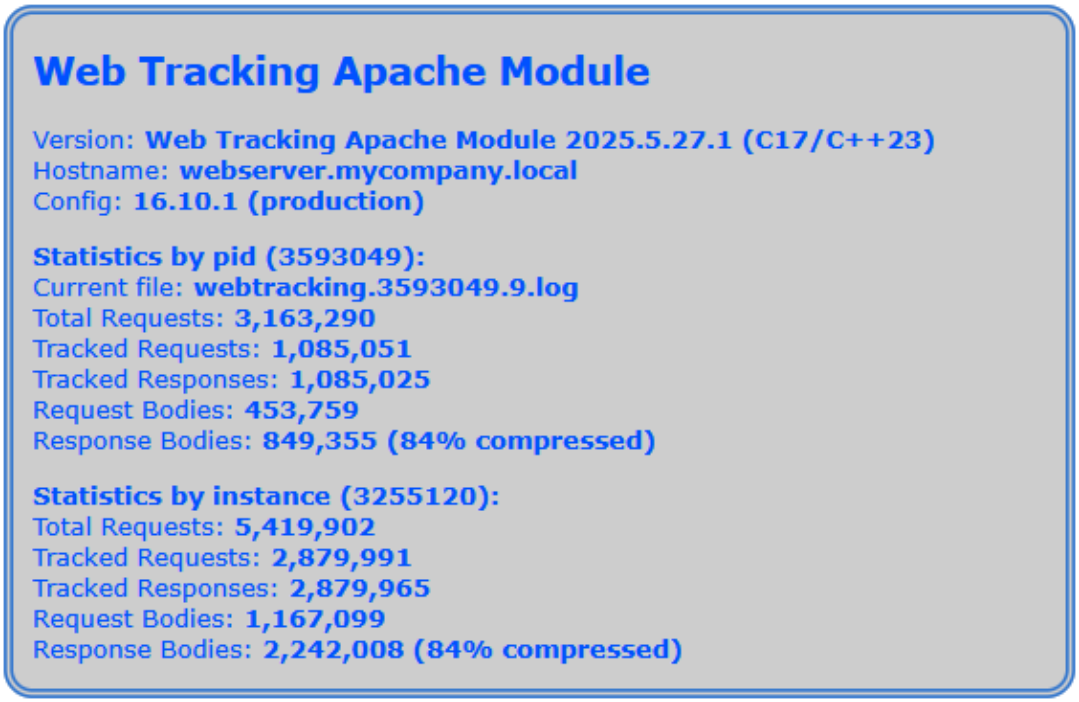
```
strings <Web Tracking .so path> | grep -E -o 'Web Tracking Apache
Module .*?\)'
```

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.



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
2025.5.27.1 [R2025.7]	2025-05-27	Fix include header files syntax Add response elapsed time and log file name in metrics records Fix log record length in metrics to be formatted (base 1024)
2025.5.22.1	2025-05-22	Add configuration checks on directive WebTrackingExactURI Add configuration checks on directive WebTrackingStartsWithURI Add configuration checks on directive WebTrackingExcludeExactURI Add configuration checks on directive WebTrackingExcludeStartsWithURI
2025.5.15.1	2025-05-15	Add directive WebTrackingConfigVersion Fix syntax for directive WebTrackingRequestBodyType Fix syntax for directive WebTrackingResponseBodyType
2025.5.12.1	2025-05-12	Add directive WebTrackingRequestBodyType Add directive WebTrackingResponseBodyType Add directive WebTrackingExcludeURIRequestBody Add directive WebTrackingExcludeURIReSponseBody Remove directive WebTrackingEnablePostBody Remove directive WebTrackingExcludeURIPost
2025.4.30.1	2025-04-30	Add current record log file name to server-status handler
2025.4.16.1 [R2025.6]	2025-04-16	Add a new metric: total requests
2025.4.15.1	2025-04-15	Fix some regressions on directive "WebTrackingUuidHeader"

VERSION	DATE	DESCRIPTION
2025.4.14.1	2025-04-14	Create header "WebTrackingUuidHeader" on every request Create header x-wt-request-to-be-tracked = true when the tracking is active for the current request Fix algorithm for chained "WebTrackingUuidHeader" header values
2025.4.10.1	2025-04-10	Add Hostname info to server-status handler
2025.4.7.1	2025-04-07	Add directive WebTrackingExcludeExactURI Add directive WebTrackingExcludeStartsWithURI Add directive WebTrackingExactHost
2025.3.25.1	2025-03-25	Add directive WebTrackingStartsWithURI Fix some minor bugs
2025.3.13.1 [R2025.5]	2025-03-13	Fix cookie removals
2025.3.5.1 [R2025.4]	2025-03-05	Add directive WebTrackingExactURI Improve trace uri implementation Add folder directory creation at startup (it depends on permissions)
2025.2.21.1 [R2025.3]	2025-02-21	Remove tracking of request with protocol different than HTTP/1.1 Add exception guards for then main functions
2025.2.18.1 [R2025.2]	2025-02-18	Remove output headers from response body Fix memory allocations to remove leaks Enhance file management to reduce its overhead Change uuid algorithm Remove directive WebTrackingID Fix encoding POST query string as “*Post” header

VERSION	DATE	DESCRIPTION
2025.2.10.2 [R2025.1]	2025-02-10	Implement request/responce cycle functions using C++23 Implement record file management in C++23 Change tracking data record format and contents Change requirements for directives WebTrackingDisablingHeader and WebTrackingOutputHeader Add styling to server status hook Implement hot debug for specific resources Implement some runtime optimizations and some code enhancements Remove directive WebTrackingPrintWASUser Remove directive WebTrackingPrintRequestHeader 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 The body limit is also compared to inflated bodies
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 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 Add directive WebTrackingRecordArchiveFolder Add directive WebTrackingRecordLifeTime Remove directive WebTrackingRecordFile

VERSION	DATE	DESCRIPTION
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 Fixed log record writing
2023.9.12.1	2023-09-12	Added logging timestamp to record Moved to GNU Compiler Collection 13.2.0
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

VERSION	DATE	DESCRIPTION
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

VERSION	DATE	DESCRIPTION
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 behavior: 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

VERSION	DATE	DESCRIPTION
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

VERSION	DATE	DESCRIPTION
1.0.1	2018-05-29	Added directive ➢ WebTrackingExcludeCookie 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): ➢ 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

5. 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	WebTrackingApplicationId /myroot MyApplication	No	1.0.0 1.1.1 (the host filter)

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		occurrence will be enabled. 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. 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. 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	WebTrackingClientIpHeader ClientIp	No	1.0.0

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		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.			
WebTrackingConfigVersion	String	The version of the defined configuration. Useful to check the active configuration because it is shown along the server status information. It can be defined once for each web server instance.	WebTrackingConfigVersion 16.0.1	No	2025.5.15.1
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	WebTrackingDisablingHeader X-WT-TR-OFF X-WT-TR-NO	No	1.0.0

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		the web racking feature. **All defined headers must start with "X-WT-" or "WT-" (since version 2025.2.5.1).** Header names are case-insensitive. It is a multi-line directive.			
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
WebTrackingExactHost	String	Define for which exact Host header values (including port if necessary) the web tracking is enabled.	WebTrackingExactHost www.mycompany.com	No	2025.4.7.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		It is a multi-line directive.			
WebTrackingExactURI	String	Define for which exact URIs the web tracking is enabled. It is a multi-line directive.	WebTrackingExactURI /my-root/home	No	2025.3.5.1
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
WebTrackingExcludeExactURI	String	Define for which exact URIs the web tracking is disabled. It is a multi-line directive.	WebTrackingExcludeExactURI /my-root/private/	No	2025.4.7.1
WebTrackingExcludeFormParameter	String	It defines which form parameter will be removed from the POST request web tracking records when the Content	WebTrackingFormParameter j_password j_username WebTrackingFormParameter secure* WebTrackingFormParameter *	No	1.0.2

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		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.			
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 SecureHeader	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.	WebTrackingExcludeHeaderValue Set-Cookie	No	1.0.0

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		It is a multi-line directive.			
WebTrackingExcludeIP	PCRE[1]	It defines the source addresses for which the web tracking is disabled. It is a multi-line directive.	WebTrackingExcludeIP ^192\.168\.22 WebTrackingExcludeIP ^10\.	No	1.0.0
WebTrackingExcludeStartsWithURI	String	Define the starting part of a URI that disables the web tracking. It is a multi-line directive.	WebTrackingExcludeStartsWithURI /my-root/	No	2025.4.7.1
WebTrackingExcludeURI	PCRE[1]	It defines for which URIs among the URIs defined by both the WebTrackingURI and WebTrackingExactURI directives the web tracking is disabled. It is a multi-line directive.	WebTrackingExcludeURI \.pdf \.jpg 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 body. It is a multi-line directive.	WebTrackingExcludeURIBody j_security_check$	No	1.0.0

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
WebTrackingExcludeURIRequestBody	PCRE[1]	It defines for which URIs enabled by other directives is disabled the tracking of the request body. It is a multi-line directive.	WebTrackingExcludeURIRequestBody ^/secureinput/	No	2025.5.12.1
WebTrackingExcludeURIResponseBody	PCRE[1]	It defines for which URIs enabled by other directives is disabled the tracking of the response body. It is a multi-line directive.	WebTrackingExcludeURIResponseBod no_response_body$	No	2025.5.12.1
WebTrackingHost	PCRE[1]	Define 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 \.my-company\.com$ 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	WebTrackingOutputHeader X-WT-USER	No	1.0.0

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		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" or "WT-" (since version 2025.2.5.1).** Header names are case-insensitive. 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. Each environment variable will be prefixed with the string "ENV:" 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	WebTrackingRecordArchiveFolder /opt/webtracking/splunk	No	2024.5.21.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		files. If not defined it will be defaulted to WebTrackingRecordFolder/archives. If the folder doesn't exist, it will be created at startup along with the missing parent folders – it depends on permissions. **Warning: The directives WebTrackingRecordFolder and WebTrackingRecordArchiveFolder must reference a folder in the same filesystem.** **Warning: If the 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.**			

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
WebTrackingRecordFolder	Path	Path of the web tracking folder where to save tracking data files. If not defined it will be defaulted to the current directory for the apache web server instance. If the folder doesn't exist, it will be created at startup along with the missing parent folders – it depends on permissions. **Warning: The directives WebTrackingRecordFolder and WebTrackingRecordArchiveFolder must reference a folder in the same filesystem.**	WebTrackingRecordFolder /opt/IBM/HTTPServer/logs	No	2024.5.21.1
WebTrackingRecordLifeTime	Number	Defines the time a single tracking data file must accept new records. It should be in the range [5, 120] and is	WebTrackingRecordLifeTime 15	No	2024.5.21.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		expressed in minutes. The default value is 30. **WARNING: A tracking data file will be closed when the size is greater than 1 GB, regardless of the time interval that has already passed.**			
WebTrackingRequestBodyType	String	Specify the type of request body handling. Possible values are (not case-sensitive): "**Content**": Enables tracking depending on the value of the Content-Type header. "**Always**": Always enables tracking. "**Never**": Never enables tracking. **The default value is "Content".** It is not a multi-line directive: the last configured	WebTrackingRequestBodyType Always	No	2025.5.12.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		directive always wins.			
WebTrackingResponseBodyType	String	Specify the type of response body handling. Possible values are (not case-sensitive): "**Content**": Enables tracking depending on the value of the Content-Type header. "**Always**": Always enables tracking. "**Never**": Never enables tracking. **The default value is "Content".** It is not a multi-line directive: the last configured directive always wins.	WebTrackingResponseBodyType Never	No	2025.5.12.1
WebTrackingStartsWithURI	String	Define the starting part of a URI that enables the web tracking. It is a multi-line directive.	WebTrackingStartsWithURI /my-root/	No	2025.3.25.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
WebTrackingTraceURI	PCRE[1]	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	PCRE[1]	Define for which URIs the web tracking is enabled. It is a multi-line directive.	WebTrackingURI /my-root/public/	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	WebTrackingUuidHeader X-APP1-UUID	No	1.1.1

NAME	SYNTAX	DESCRIPTION	EXAMPLE	REQ	FROM
		for each web server instance.			

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).

6. Algorithm

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

1. **Reading and analysis of the request** to determine whether web tracking should be enabled.
2. **Reading and analysis of the request** to decide whether tracking of the request and/or the response body should be enabled.
3. **Writing the record** to the defined stream.

The fundamental phases for the operations of the module are points 1 and 2, while point 3 defines the artifacts of the solution through the records written to the stream.

Rules for Fulfilling Points 1 and 2 (in order of priority):

1. **Check whether web tracking is enabled as a whole** (WebTrackingDisable).
2. **Check whether the host enables web tracking** (based on the value of the host header: WebTrackingExactHost, WebTrackingHost).
3. **Check whether the request URI enables web tracking** (WebTrackingExactURI, WebTrackingStartsWithURI, WebTrackingURI).
4. **Check whether the request URI disables web tracking** (WebTrackingExcludeExactURI, WebTrackingExcludeStartsWithURI, WebTrackingExcludeURI).
5. **Check whether the SSL Offloading header is present** among the request headers (WebTrackingSSLIndicator).
6. **Check whether the scheme of the request enables web tracking** (WebTrackingHttpsEnabled, WebTrackingHttpEnabled).
7. **Check whether any request headers disable web tracking** (WebTrackingDisablingHeader).
8. **Check whether the real source IP disables web tracking** (WebTrackingExcludeIP). Note: The source IP address is also based 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 that are present in the cookie and set-cookie headers must be removed from the web tracking records** (WebTrackingExcludeCookie).
13. **Check which POST form parameters must be removed from the request web tracking record** (WebTrackingExcludeFormParameter).

Additional Rules for Tracking the request and/or response body:

1. **Check the value of request body type** (WebTrackingRequestBodyType)
2. **Check the value of response body type** (WebTrackingResponseBodyType)
3. **Check whether the URI disables tracking of the request and/or response body** (WebTrackingExcludeURIBody, WebTrackingExcludeURIRequestBody, WebTrackingExcludeURIResponseBody).
4. **Check whether the request content-type header enables tracking of the request body and whether the response content-type header enables tracking of the response body** (WebTrackingContentType).
5. **Check whether the response size is less than or equal to the maximum size defined**. If it exceeds the limit, tracking is disabled (WebTrackingBodyLimit).

7. Record Layout

The web tracking is enabled when defined by the following directives or rules.

WebTrackingDisable, WebTrackingExcludeIP, WebTrackingExcludeExactURI, WebTrackingExcludeStartsWithURI, WebTrackingExcludeURI, WebTrackingExactHost, WebTrackingHost, WebTrackingHttpEnabled, WebTrackingHttpsEnabled, WebTrackingTraceURI, WebTrackingExactURI, WebTrackingStartsWithURI, WebTrackingURI

Enabling or disabling the tracking of the request body or the response body, or both, follows these rules where the rule with the lowest number takes priority:

1. If the request URI matches one of the WebTrackingTraceURI directives, tracking of the request and response body is always enabled regardless of the values of other directives and rules.
2. If the WebTrackingRequestBodyType directive is set to **Never**, tracking of the request body is disabled.
3. If the WebTrackingResponseBodyType directive is set to **Never**, tracking of the response body is disabled.
4. If the request URI matches one of the WebTrackingExcludeURIBody directives, tracking of the request and response body is disabled.
5. If the request URI matches one of the WebTrackingExcludeURIRequestBody directives, tracking of the request body is disabled.
6. If the request URI matches one of the WebTrackingExcludeURIResponseBody directives, tracking of the response body is disabled.
7. If the WebTrackingRequestBodyType directive is set to **Always**, tracking of the request body is enabled.
8. If the WebTrackingResponseBodyType directive is set to **Always**, tracking of the response body is enabled.
9. If the WebTrackingRequestBodyType directive is set to **Content**, tracking of the request body depends on the WebTrackingContentType and WebTrackingBodyLimit directives. The absence of the request header content-type disables the tracking of the request body.

10. If the WebTrackingResponseBodyType directive is set to **Content**, tracking of the response body depends on the WebTrackingContentType and WebTrackingBodyLimit directives.

The record layout of the web tracking is defined 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
- [webtracking.server.local]
- 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)
  [APPLICATION_20241221]
- "****REQUEST****"
- Request Timestamp
  [2025-01-28 10:46:57.618 CET]
- Remote IP
  (WebTrackingEnableProxy, WebTrackingClientIpHeader)
  [10.10.198.115]
- Protocol[6]
  [6] At the moment is the only supported protocol version.
  [HTTP/1.1]
- Method
  [POST]
- URL
  [https://www.mycorp.com/public/home]
- "**HEADERS**"
- Request Headers
  (WebTrackingExcludeCookie, WebTrackingExcludeHeader, WebTrackingExcludeHeaderValue, WebTrackingPrintEnvVar, WebTrackingExcludeFormParameter)

[Host: private.mycorp.com]
[PrivateRequestHeader]
[*Post: domain=.mycorp.com&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, WebTrackingExcludeURIBody, WebTrackingExcludeURIRequestBody, WebTrackingRequestBodyType, 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=app1.server.local:9101]
- "****RESPONSE_BODY****"[4]
- BAS64(RESPONSE BODY)[4]
  (WebTrackingBodyLimit, WebTrackingExcludeURIBody, WebTrackingExcludeURIResponseBody, WebTrackingResponseBodyType, 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.

8. 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

# Version
WebTrackingConfigVersion "16.0.1 (production)"

# Web Tracking Header
WebTrackingUuidHeader X-WT-UUID

# Application Id
WebTrackingApplicationIdFromHeader application-id
WebTrackingApplicationId / WEBTRACKING

# Web Tracking Directives
WebTrackingHost \.mycorp\.com$
WebTrackingExactURI /wlptest/snoop
WebTrackingStartsWithURI /mycontext/
WebTrackingURI Precom
WebTrackingExcludeExactURI /mycontext/login
WebTrackingExcludeURI \.(pdf|jpg|css|png|js|gif|ico|eot|woff$|woff2|map|ttf|svg)$
WebTrackingExcludeURI ^/server-status/
```

```
WebTrackingContentType html|json|text\/(?!csv)|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 /webtracking/logs
WebTrackingRecordArchiveFolder /webtracking/splunk
WebTrackingRecordLifeTime 15
```

To disable the tracking of both the request and the response bodies configure WebTrackingRequestBodyType and WebTrackingResponseBodyType to "**Never**".

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 /webtracking/logs/webtracking*.log /webtracking/splunk
rm -fv /prod/webtracking/lib/*
unzip -uo ~/webtracking-bin.zip -d /prod/
```

9. 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> |<response time> | <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>#formatted written-bytes<else>KO | <elapsed time to write to file> | <log file name>]
```

Sample of metrics record log:

```
[Wed Feb 05 17:36:50.248970 2025] [web_tracking:info] [pid 3819381:tid 140265348957952] [WT-METRICS: webtracking.server.local:Z6OToQuMcAc4W-gG8aTQ9wAAAeE | APP_23| /private/getuser | 200 | 1.295 ms | 934 us | NO | RESPONSE| 7.815 KB | 57 us | webtracking.3819381.9.log]
```

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:

```
# Private
https://private.mycorp.com/private/v1/

# Public
```

```
https://www.mycorp.com/html/
```

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
0,30 8-20 * * * find /webtracking/logs/ -name "webtracking*.log" -type f -mmin +30 -exec mv {}
/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 WebTrackingExcludeExactURI or similar.
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.