Java Application
Monitoring

Agent Configure

이 문서는 와탭 APM 서비스 사용자가 에이전트 설치를 돕기 위해 작성된 문서 입니다.
이 문서는 와탭 랩스의 고유한 자산으로 재배포 또는 사용을 위해서는
와탭랩스(support@whatap.io) 에 연락주시기 바랍니다.

목차

목차

목차

목차

목차

1.1. Java 애플리케이션 모니터링

1.1.1. 에이전트 기능제어

1.1.1.1. enabled

default: true
전체 기능을 활성화합니다. 단 false 가 되어도 서버와 최소한의 통신을 유지하기 위한 정보는 전송됩니다.

1.1.1.2. transaction_enabled

default: true 단 enabled==false 이면 무시됨(false)
트랜잭션 추적 기능을 활성화합니다.

1.1.1.3. counter_enabled

default: true 단 enabled==false 이면 무시됨(false)
성능 카운터(트랜잭션, 리소스 등) 추적을 활성화합니다.

1.1.1.4. stat_enabled

default: true 단 enabled==false 이면 무시됨(false)
통계 정보 추적을 활성화합니다. 5 분마다 트랜잭션, SQL, HTTPCALL, UserAgent, Client IP 등의 통계 데이터가 수집되는데 이들 정보의 수집이 중단됩니다.

1.1.1.5. sigar_enabled

default: true 단 enabled==false 이면 무시됨(false)
sigar library 를 통한 OS 정보 수집을 활성화합니다. 5 초 마다 CPU, Memory, Disk 등의 자원 데이터를 sigar library 를 통해 수집할 지의 여부를 결정합니다.

1.1.1.6. active_stack_enabled

default: true 단 enabled==false 또는 counter_enabled==false 이면 무시됨(false)
액티브 스택 추적을 활성화합니다.

1.1.1.7. license

default:
에이전트가 속한 프로젝트의 PCODE 서버와 보안 통신을 위한 암호키를 포함하고 있는 보안 문자열을 설정합니다.

1.1.1.8. cypher_level

default: 128
AES 보안 알고리즘에 대한 암호 레벨을 지정한다 128,256 중 하나를 사용할 수 있다.

1.1.1.9. encrypt_level

default: 2 단, 값의 범위는 1,2,3 중에 하나를 지정할 수 있다.
와탭은 데이터에 따라 다른 암호화 알고리즘을 적용합니다. 주로 Text 는 보안레벨을 높게 적용하고 단순 숫자 데이터는 보안레벨을 낮게 적용합니다.

이러한 보안레벨을 적용할 때 일괄적으로 전체적인 보안 레벨을 보다 높게 혹은 보다 낮게 적용할 수 있는데 그것을 지정하는 옵션이다.

1.1.1.10. dbcp_pool_enabled

default: false

데이터 소스 풀 추적을 활성화하는 기능이다. datasource 를 직접 추적하고자 하는 경우 true 로 설정한다.

- JMX 비활용 (datasource connection pool) 정보를 수집
- ANALYSIS > Daily Counter 의 Resource - DB Connection Total/Active/Idle 지표 수집에 관여합니다.

1.1.1.11. hikari_pool_enabled

default: false

hikari 데이터 소스 풀 추적을 활성화하는 기능이다. hikari dbcp 를 직접 추적하고자 하는 경우 true 로 설정한다.

- JMX 비활용 (datasource connection pool) 정보를 수집
- ANALYSIS > Daily Counter 의 Resource - DB Connection Total/Active/Idle 지표 수집에 관여합니다.

1.1.1.12. tomcat_ds_enabled

default: false

톰캣 데이터 소스 추적을 활성화하는 기능이다. 실질적으로 tomcat datasource 를 추적하고자 하는 경우 true 로 설정한다.

- JMX 활용
- ANALYSIS > Daily Counter 의 Resource - DB Connection Total/Active/Idle 지표 수집에 관여합니다.

1.1.1.13. weblogic_ds_enabled

default: false

weblogic 데이터 소스 추적을 활성화하는 기능이다. 실질적으로 weblogic datasource 를 추적하고자 하는 경우 true 로 설정한다.

- JMX 활용
- ANALYSIS > Daily Counter 의 Resource - DB Connection Total/Active/Idle 지표 수집에 관여합니다.

1.1.1.14. weblogic_pool_enabled

default: false

weblogic 데이터 소스 풀 추적을 활성화하는 기능이다. 실질적으로 weblogic datasource 를 추적하고자 하는 경우 true 로 설정한다.

- JMX 비활용 (datasource connection pool) 정보를 수집
- ANALYSIS > Daily Counter 의 Resource - DB Connection Total/Active/Idle 지표 수집에 관여합니다.

1.1.2. 에이전트 네트워크 설정

1.1.2.1. whatap_server_host

default: 127.0.0.1, 127.0.0.1

수집서버 아이피를 지정합니다. 콤마(,)로 분리하여 하나 혹은 2 개를 지정할 수 있다. 단 여기서 지정하는 서버에는 proxy 서버가 리스닝하고 있어야 합니다.

1.1.2.2. whatap_server_port

default: 6600

수집서버 PORT 를 지정합니다. 포트는 하나만 지정할 수 있다. 따라서 whatap_server_host 에 지정된 수집서버들은 동일 PORT 를 사용해야 합니다.

1.1.2.3. tcp_so_timeout

default: 60000

수집서버와 통신할 때 TCP 세션의 idle 타임아웃 값을 지정합니다.

1.1.2.4. tcp_connection_timeout

default: 5000

수집서버와 통신 세션을 연결할 때 연결 지연 가능 시간을 지정합니다.

1.1.2.5. net_send_max_bytes

default: 5242880

에이전트가 데이터를 수집하고 네트웍으로 한번에 전송할 수 있는 최대 byte 크기이다.

1.1.2.6. net_send_queue1_size

default: 512

에이전트는 두개의 네트워크 큐를 사용합니다. 1 번큐에는 프로파일과 액티브스택을 제외한 모든 데이터 전송 시 사용됩니다.

1.1.2.7. net_send_queue2_size

default: 1024

에이전트는 두개의 네트워크 큐를 사용합니다. 2 번큐에는 프로파일과 액티브스택을 전송하는데 사용됩니다.

1.1.3. 트랜잭션 처리 제한

1.1.3.1. throttle_enabled

default: false

쓰로틀링 기능을 활성화합니다.

1.1.3.2. throttle_limit

default: 10000

최대 동시 처리 개수를 지정합니다. WAS 에서 동시 처리되는 요청(트랜잭션)수가 지정한 값을 넘으면 추가로 도달하는 요청은 reject 됩니다.

1.1.3.3. throttle_passing_url

default:

최대동시 처리 한계를 초과해도 reject 하지않고 처리해야할 URL, 만약 여러 개를 지정해야 하면 콤마(,)를 사용합니다.

1.1.3.4. throttle_passing_url_prefix

default:

최대동시 처리 한계를 초과해도 reject 하지않고 처리해야할 URL 의 prefix, 만약 여러 개를 지정해야 하면 콤마(,)를 사용합니다.

1.1.3.5. throttle_blocking_url

default:

무조건 블럭킹(처리 거부) 해야 할 URL 을 지정합니다. 시스템의 장애가 나는 URL 를 봉쇄하기 위해 사용할 수 있다.

1.1.3.6. throttle_blocking_ip

default:

무조건 요청을 거부해야 할 ip 를 지정합니다. 디도스 공격이나 잘못된 사용자를 IP 기반으로 봉쇄할 때 사용할 수 있다.

1.1.3.7. throttle_rejected_message

default: too many request!!

사용자 요청이 limit 값을 넘어 reject 될 때 사용자에게 전달되는 안내 메시지

1.1.3.8. throttle_rejected_forward

default:

사용자 요청이 limit 값을 넘어 reject 될 때 사용자에게 전달되는 안내 페이지 URL,

주의) 안내 페이지는 static html 페이지로 만들어야 합니다.

dynamic 페이지로 안내를 만드는 경우에는 무한루프에 빠져 장애를 유발할 수 있다.

1.1.3.9. throttle_rejected_forward_ok

default: true

사용자 요청이 limit 값을 넘어 reject 될 때 사용자에게 안내 페이지로 forward 할지 단순 메시지를 전송할지를 결정함.

1.1.3.10. throttle_blocked_message

default: request blocked!!

사용자 요청이 block 되어 메세지로 안내될 때 사용될 메시지

1.1.3.11. throttle_blocked_forward

default:

사용자 요청이 block 되었을 때 포워드 할 안내 페이지

1.1.3.12. throttle_blocked_forward_ok

default: true

사용자 요청이 block 될 때 안내 페이지로 포워드 할지 단순 메세지로 안내할지를 지정함

1.1.3.13. reject_event_enabled

default: false

사용자 요청이 block 될 때 이벤트 알림을 발행할 지를 지정함

1.1.3.14. reject_event_interval

default: 30000

사용자 요청이 block 될 때 이벤트 알림 발행 간격을 지정함

- reject_enabled=true 로 설정된 경우에 한하여 유효함

1.1.4. 프로파일링 옵션

1.1.4.1. profile_http_header_enabled

default: false

http 헤더 정보를 프로파일에 출력하고자 할 때

- 헤더 수집 여부와 무관함, 헤더를 프로파일에 노출 할 지에 대한 설정임

1.1.4.2. profile_http_header_url_prefix

default: /

http 헤더를 프로파일에 출력할 때 대상 URL 에 대한 prefix

1.1.4.3. profile_http_parameter_enabled

default: false

http 파라미터를 프로파일링을 활성화합니다. 단 파라미터는 별도 보안키를 입력해야 조회할 수 있다.

- 보안 키는 에이전트 설치 경로의 paramkey.txt 파일에 6 자리로 지정합니다. paramkey.txt 가 존재하지 않는 경우 자동 생성되며 random 키가 설정됩니다.

1.1.4.4. profile_http_parameter_url_prefix

default: /

http 파라미터를 프로파일링 활성화할 때 적용될 URL prefix 를 설정합니다.

1.1.4.5. profile_connection_open_enabled

default: true

DBConnection 오픈 정보를 프로파일링 할 때 활성화여부를 지정합니다.

1.1.4.6. profile_step_normal_count

default: 800

프로파일 기본 스텝 수를 제한합니다.

1.1.4.7. profile_step_heavy_count

default: 1000

프로파일의 기본스텝을 초과하여 최대 heavy 스텝수를 제한합니다.
profile_step_normal_count 에서 profile_step_heavy_count 사이에서는 profile_step_heavy_time 을 초과하는 스텝만 수집됩니다.

1.1.4.8. profile_step_max_count

default: 1024

프로파일 스텝의 최대 수, 수집된 프로파일 스텝 수가 이 값을 초과하면 이후 수집되는 스텝들은 모두 버려진다.

1.1.4.9. profile_step_heavy_time

default: 100

profile_step_normal_count 에서 profile_step_heavy_count 사이에서는 profile_step_heavy_time 을 초과하는 스텝만 수집됩니다.

1.1.4.10. profile_basetime

default: 500

트랜잭션의 처리 시간이 이 값에 미치지 못하는 경우 프로파일 정보는 수집되지 않는 다. 단 5 분당 최초 호출된 URL, 에러 트랜잭션은 수집됩니다.

1.1.4.11. profile_sql_param_enabled

default: false

SQL 파라미터를 수집을 활성화합니다. 단 파라미터는 별도 보안키를 입력해야 조회할 수 있다.

- 보안 키는 에이전트 설치 경로의 paramkey.txt 파일에 6 자리로 지정합니다. paramkey.txt 가 존재하지 않는 경우 자동 생성되며 random 키가 설정됩니다.

1.1.4.12. profile_sql_resource_enabled

default: false

프로파일에서 SQL 스텝이 수집될 때 트랜잭션이 시작부터 해당 스텝까지 사용한 CPU 와 메모리 사용량을 추적합니다.

1.1.4.13. profile_method_resource_enabled

default: false

프로파일에서 METHOD 스텝이 수집될 때 트랜잭션이 시작부터 해당 스텝까지 사용한 CPU 와 메모리 사용량을 추적합니다.

1.1.4.14. profile_httpc_resource_enabled

default: false

프로파일에서 HTTP Call 스텝이 수집될 때 트랜잭션이 시작부터 해당 스텝까지 사용한 CPU 와 메모리 사용량을 추적합니다.

1.1.4.15. profile_dbc_close

default: false

DB Connection 이 close 될때 프로파일 스텝으로 추가할 때 설정합니다. 단 open connection 이 출력되어야 이 옵션이 동작합니다.

- trace_dbc_leak_enabled=true 인 경우에만 활성화 됩니다.

1.1.4.16. profile_position_sql

default:

이 옵션에서 지정한 SQL 이 수행되면 어디서 수행되었는지 스택을 같이 프로파일에 출력합니다.

1.1.4.17. profile_position_httpc

default:

알 수 없는 HTTPC 가 프로파일링 될 때 그 위치를 찾아낼 때 사용합니다.

1.1.4.18. profile_position_method

default:

프로파일링 되는 메소드가 어떻게 호출되는지 디버깅 하고자 할 때 true 설정합니다.

1.1.4.19. profile_position_depth

default: 50

position 추적을 위해 스택을 덤프 할 때 스택 라인 수를 지정합니다.

1.1.4.20. profile_update_count

default: false

update sql 의 건수를 프로파일 정보에 출력합니다. excuteUpdate 메소드를 호출한 경우에 한하여 출력합니다.

1.1.5. 사용자 추적 옵션

1.1.5.1. trace_user_enabled

default: true

실시간 사용자를 추적할지 결정합니다.

- default 가 true 이므로 일반적인 경우 설정하지 않는 것을 추천합니다.

1.1.5.2. trace_user_using_ip

default: false

IP 를 기반으로 실시간 사용자를 추적합니다.

- user_header_ticker 와 배타적 설정으로 동시에 적용할 수 없습니다.
- Realtime User 를 count 하기 위한 정보로 사용됩니다.
- trace_http_client_ip_header_key 와 동시에 적용 시, 헤더로부터 추출한 정보를 기반으로 Realtime User 및 Client IP 를 추적 할 수 있습니다.
  - ex) x-forwarded-for

1.1.5.3. trace_user_using_jsession

default: false

실시간 사용자를 추적할 때 사용자 구분을 JSESSION 쿠키 값으로 합니다.

1.1.5.4. trace_user_cookie_limit

default: 2048

사용자 구분을 쿠키로 하는 경우 새로운 사용자가 접속하면 UUID 를 쿠키로 지정하여 사용자를 구분합니다. 그런데 기존의 쿠키가 너무 많은 경우 쿠키 오버플로어가 날 수 있다. 이것을 피하기 위해 limit 를 지정합니다.

1.1.5.5. user_header_ticket_enabled

default: false

사용자 아이디를 http 헤더의 특정 값으로 구분하고 싶을 때 사용합니다. 모바일에서 접속할 때 전달되는 경우가 많다.

1.1.5.6. user_header_ticket

default:

사용자 아이디를 http 헤더의 특정 값으로 구분하고 싶을 때 사용하는 키 명칭을 지정합니다.

- 설정 시, user_header_ticker_enabled 가 true 로 설정된 것으로 간주됩니다.
- Realtime User 를 count 하기 위한 정보로 사용됩니다.
- trace_user_using_ip 와 배타적 설정으로 동시에 적용할 수 없다.

1.1.5.7. trace_http_client_ip_header_key

default:

사용자의 실제 접속 아이피가 header 에 별도로 전달되는 경우 해당 header key 를 지정합니다.

- Client IP 를 특정하기 위한 정보로 활용됩니다. 취득된 값으로 remote address 를 대체합니다

1.1.6. 트랜잭션 추적 옵션

1.1.6.1. trace_auto_transaction_enabled

default: false

프로파일링(메소드 스텝) 메소드에서 트랜잭션이 시작되지 않았다면 무시되는데 이때 자동으로 트랜잭션을 시작 시키도록 함. 프로덕션 보다는 주로 개발환경에서 백그라운드 트랜잭션의 END POINT 를 찾아낼 때 사용합니다.

1.1.6.2. trace_auto_transaction_backstack_enabled

default: true

trace_auto_transaction_enabled=true 상태에서 자동으로 정의된 트랜잭션의 시작 지점에서 스택을 남김으로 진입점이 어디인지를 추적하고자 할 때 사용합니다.

1.1.6.3. trace_background_socket_enabled

default: true

소켓(TCP) 연결이 오픈될 때 트랜잭션이 시작된 상황에서만 오픈을 추적하는데 트랜잭션이 아닌 백그라운드 쓰레드에 의한 소켓이 오픈될 때도 추적한다

1.1.6.4. trace_transaction_name_header_key

default: null

트랜잭션의 이름을 header 에서 전달되는 값을 URL 에 추가한다

1.1.6.5. trace_transaction_name_key

default: null

트랜잭션의 이름을 http request parameter 의 값을 URL 에 추가합니다.

1.1.6.6. trace_error_callstack_depth

default: 50

트랜잭션에서 에러의 콜스택을 수집할 때 지정한 라인 수(default 50) 라인만 수집합니다. (Error 통계에서 확인)

1.1.6.7. trace_active_callstack_depth

default: 50

액티브스택의 수집되는 콜스택 최대 라인수를 지정합니다

1.1.6.8. trace_active_transaction_yellow_time

default: 3000

액티브 트랜잭션의 아크이퀄라이저에서 노란색 구간의 응답기준

1.1.6.9. trace_active_transaction_red_time

default: 8000

액티브 트랜잭션의 아크이퀄라이저에서 빨간색 구간의 응답기준

1.1.6.10. trace_intertx_enabled

default: false

멀티 티어 트랜잭션을 연결 추적하는 기능을 활성화합니다.

1.1.6.11. trace_dbc_leak_enabled

default: false

DBConnection Leak 을 추적하는 기능을 활성화합니다. Connection Wrapper 를 통해서 Leak 을 추적하기 때문에 일부 WAS(ex 웹로직) 에서는 미리 테스트후 적용해야 합니다.

1.1.6.12. trace_dbc_leak_fullstack_enabled

default: false

DB Connection 를 사용 후 반환하지 않는 트랜잭션의 경우 Connection Leak 위치를 확인하기 위해서 Fullstack 이 필요할 수 있다. 부분적인 CPU 사용량이 몇% 정도 증가할 수 있음으로 CPU 사용량이 높은 시스템에서는 Peak 타임을 피해서 활성화하는 것을 권고합니다.

1.1.6.13. trace_httpc_normalize_urls

default:

트랜잭션이 외부 HTTP 호출하는 URL 을 정규화합니다. 호출 URL 패턴을 파싱하여 패스파라미터를 제거합니다.

- ex) /a/{v}/b 라고 선언하면 a/123/b => a/{v}/b 로 치환한다
- 여러 개를 등록할 때는 콤마(,)를 사용합니다.

1.1.6.14. trace_httpc_normalize_enabled

default: true

HTTP Call URL 을 파싱하여 정규화하는 기능을 활성화합니다.

1.1.6.15. trace_normalize_urls

default:

트랜잭션 URL 을 파싱하여 정규화합니다. 호출 URL 패턴을 파싱하여 패스파라미터를 제거합니다.

- ex) /a/{v}/b 라고 선언하면 a/123/b => a/{v}/b 로 치환한다
- 여러개를 등록할 때는 콤마(,)를 사용합니다.

1.1.6.16. trace_normalize_enabled

default: true

트랜잭션 URL 을 파싱하여 정규화하는 기능을 활성화합니다. False 로 변경시 패스파라미터 파싱이 비활성화됩니다.

- 다만 이 경우 통계데이터의 의미가 약화됨으로 디버그 용도로만 잠시 사용하는 것이 좋습니다.

1.1.6.17. trace_auto_normalize_enabled

default: true

트랜잭션 URL 정규화할때 패턴값을 어노테이션에서 추출하여 자동으로 파싱하는 기능을 활성화합니다.

1.1.6.18. trace_sql_normalize_enabled

default: true

SQL 문에서 리터럴 부분을 축출하여 SQL 문을 정규화하는 기능을 활성화합니다.

1.1.6.19. web_static_content_extensions

default: js, htm, html, gif, png, jpg, css, swf, ico

스태틱 컨텐츠를 판단하는 확장자를 설정합니다. 여기에 설정된 확장자를 가진 트랜잭션들은 프로파일 추적과 카운팅이 제외됩니다.

1.1.6.20. profile_error_jdbc_fetch_max

default: 10000

SQL 수행후 패치건수가 여기서 지정한 값을 초과하면 TOO MANY 조회 에러로 처리됩니다.

1.1.6.21. profile_error_sql_time_max

default: 30000

SQL 수행후 수행시간이 여기서 지정한 값을 초과하면 TOO SLOW 에러로 처리됩니다.

1.1.6.22. prepared_sql_max

default: 7001

attach 나 watcher 방식으로 추적할 때 PreparedStatement 에서 수행되는 SQL 을 캐싱하는데 캐시의 크기이다. javaagent 방식에서는 설치 시 사용되지 않습니다.

1.1.6.23. mtrace_rate

default: 100

type:%

최초 트랜잭션이 발생할 때 발급받는 MTID(Multi Transaction ID)의 발급 비율을 설정하는 옵션이다. MTID 를 추적하면 등록된 모든 애플리케이션 간의 호출을 확인 할 수 있습니다. 같은 프로젝트에 속한 애플리케이션은 Caller & Callee 기능을 통해 트랜잭션의 프로파일을 바로 확인 가능합니다.

1.1.6.24. recursive_max

default: 1000000

unit: count

트랜잭션의 재귀호출 여부 검출을 위한 옵션으로, 단일 트랜잭션으로 부터 파생되는 재귀호출 횟수를 카운트하여 이벤트 알림을 발행하기 위한 기준을 지정합니다.

- HTTP URL 재귀호출을 대상으로 함
- jsp:forward 를 통해 재호출 되는 케이스도 카운트에 포함됨

1.1.7. 로그 옵션

1.1.7.1. log_datasource_lookup_enabled

default: true

InitialContext Lookup 시에 DataSource 라 Lookup 되면 로깅합니다.

1.1.7.2. log_rotation_enabled

default: true

에이전트 로그파일을 매일 변경합니다.

1.1.7.3. log_keep_days

default: 7

로그파일 보관기간을 설정합니다.

1.1.8. 바이트코드 INSTRUMENTATION 옵션

1.1.8.1. hook_connection_open_patterns

default:

Connection Open 시 호출되는 메소드를 등록합니다. 미리 지정되지 않는 Connection Pool 의 getConnection 을 등록하는 것이 일반적입니다.

- ex) hook_connection_open_patterns=mypool.ConPool.getConnection

1.1.8.2. hook_method_patterns

default:

특정 메소드의 응답시간을 측정하고 싶을 때 사용합니다.
마지막 (.)가 구분자이며, 그 앞쪽은 클래스 뒤쪽은 메소드입니다.

- ex) hook_method_patterns=a.b.C1.*

1.1.8.3. hook_method_ignore_prefixes

default: get,set

메소드 프로파일을 설정할 때 지정한 문자열로 시작하는 메소드들은 응답을 추적하지 않습니다.

1.1.8.4. hook_method_ignore_classes

default:

메소드 프로파일을 설정할 때 배제 하고 싶은 클래스들을 설정합니다.

1.1.8.5. hook_method_access_public_enabled

default: true

메소드 프로파일을 설정할 때 public 메소드에 대해서만 별도로 대상으로 할지를 결정합니다.

1.1.8.6. hook_method_access_private_enabled

default: false

메소드 프로파일을 설정할 때 private 메소드에 대해서만 별도로 대상으로 할지를 결정합니다.

1.1.8.7. hook_method_access_protected_enabled

default: true

메소드 프로파일을 설정할 때 protected 메소드에 대해서만 별도로 대상으로 할지를 결정합니다.

1.1.8.8. hook_method_access_none_enabled

default: true

메소드 프로파일을 설정할 때 no access indicated 메소드에 대해서만 별도로 대상으로 할지를 결정합니다.

1.1.8.9. hook_service_access_public_enabled

default: true

Non Http Demon 프로세스의 트랜잭션을 지정할 때 public 메소드에 대해서만 Access 권한을 기준으로 on/off 를 지정합니다

1.1.8.10. hook_service_access_private_enabled

default: true

Non Http Demon 프로세스의 트랜잭션을 지정할 때 private 메소드에 대해서만 Access 권한을 기준으로 on/off 를 지정합니다

1.1.8.11. hook_service_access_protected_enabled

default: true

Non Http Demon 프로세스의 트랜잭션을 지정할 때 protected 메소드에 대해서만 Access 권한을 기준으로 on/off 를 지정합니다

1.1.8.12. hook_service_access_none_enabled

default: true

Non Http Demon 프로세스의 트랜잭션을 지정할 때 no access indicated 메소드에 대해서만 Access 권한을 기준으로 on/off 를 지정합니다

1.1.8.13. hook_serivce_ignore_methods

default:

Non Http Demon 프로세스의 트랜잭션을 지정할 때 제외할 메소드 이름을 지정한다, 콤마(,) 구분자를 사용하여 멀티로 지정합니다.

- example: hook_serivce_ignore_methods=run,start

1.1.8.14. hook_service_patterns

default:

NON HTTP 트랜잭션의 END POINT 를 지정합니다.

1.1.8.15. hook_httpservlet_classes

default:

HTTP 트랜잭션의 END POINT 를 추가로 지정한다 메소드의 첫번째 2 개의 파라미터는 HttpServletRequest 와 HttpServletResponse 만 지정 가능합니다.

1.1.8.16. hook_httpc_patterns

default:

HTTP Call 을 수행하는 클래스를 지정합니다.

1.1.8.17. hook_future_classes

default:

java.util.concurrent.Future 인터페이스를 implment 한 클래스를 설정하여 비동기 클래스를 추적하고자 할 때 활용합니다. full package class 명을 컴마(,) 구분자를 사용하여 복수의 클래스를 지정할 수 있습니다.

1.1.8.18. hook_future_prefix

default:

java.util.concurrent.Future 인터페이스를 implment 한 클래스를 설정하여 비동기 클래스를 추적하고자 할 때 활용합니다. full package class 명의 prefix 를 지정하며, 컴마(,) 구분자를 사용하여 복수의 prefix 를 지정할 수 있습니다.

1.1.8.19. hook_runnable_classes

default:

java.lang.Runnable 인터페이스를 implment 한 클래스를 설정하여 비동기 클래스를 추적하고자 할 때 활용합니다. full package class 명을 컴마(,) 구분자를 사용하여 수의 클래스를 지정할 수 있습니다.

1.1.8.20. hook_runnable_prefix

default:

java.lang.Runnable 인터페이스를 implment 한 클래스를 설정하여 비동기 클래스를 추적하고자 할 때 활용합니다. full package class 명의 prefix 를 지정하며, 컴마(,) 구분자를 사용하여 복수의 prefix 를 지정할 수 있습니다.

1.1.8.21. hook_jdbc_pstmt_classes

default:

example : oracle.jdbc.driver.OraclePreparedStatement

미등록 되었던 jdbc PreparedStatement 클래스를 설정합니다. 주의할 점은 생성자 파라미터에 SQL 문자열이 전달되는 구조여야 합니다.

- ex) Derby : org.apache.derby.impl.jdbc.EmbedPreparedStatement

1.1.8.22. hook_jdbc_stmt_classes

default:

미등록 되었던 JDBC Statement 클래스를 설정합니다.

- ex) Derby : org.apache.derby.impl.jdbc.EmbedStatement

1.1.8.23. hook_jdbc_rs_classes

default:

미등록되었던 JDBC ResultSet 클래스를 설정합니다.

- ex) Derby : org.apache.derby.impl.jdbc.EmbedResultSet

1.1.8.24. hook_jdbc_wrapping_driver_patterns

default:

DB2 드라이버처럼 난독 처리된 JDBC 드라이버는 hook_jdbc_xxx 옵션으로 직접 BCI 가 어렵다 이런 경우 Wrapper 방식으로 SQL 추적할 수 있는데 이때 Driver.connect 를 지정하여 추적하게 됩니다.

1.1.8.25. hook_jsp_patterns

default: org.apache.jasper.servlet.JspServlet.serviceJspFile (자동 추가)

JSP 파일을 로딩하는 메소드를 지정합니다.

트랜잭션 호출 결과로 반환되는 JSP 정보를 프로파일에 표시합니다.

본 옵션을 통해 추가한 설정에 default 설정이 자동으로 추가됩니다.

1.1.8.26. hook_trace_helper_patterns

default:

메소드 실행 및 종료 부분에서 프로파일 플러그인을 삽입할 포인트(클래스 및 메소드명)를 지정합니다

- * plugin 을 활용한 커스터마이즈 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입됩니다.
  - $WHATAP_HOME/plugin/TraceHelperStart.x
  - $WHATAP_HOME/plugin/TraceHelperEnd.x

1.1.8.27. hook_trace_helper_end_patterns

default:

메소드 종료 부분에서 프로파일 플러그인을 삽입할 포인트(클래스 및 메소드명)를 지정합니다.

- *plugin 을 활용한 커스터마이즈 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입됩니다.
  - $WHATAP_HOME/plugin/TraceHelperEnd.x

1.1.8.28. hook_trace_helper_start_patterns

default:

메소드 시작 부분에서 프로파일 플러그인을 삽입할 포인트(클래스 및 메소드명)를 지정합니다.

- * plugin 을 활용한 커스터마이즈 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입됩니다.
  - $WHATAP_HOME/plugin/TraceHelperStart.x

1.1.8.29. hook_direct_patch_classes

default:

직접적으로 특정 클래스를 로딩타임에 바꿔치기 하고자 할 때 사용합니다. 클래스를 컴파일한 후에 별도 파일로 만들고 그 파일의 풀패스를 설정합니다.

1.1.9. 운영 설정

1.1.9.1. active_stack_second

default: 10

액티브 스택을 추적하는 간격을 설정합니다. (주의: 값을 바꾸는 것을 권장하지 않습니다.)

1.1.9.2. boot_redefine_size

default: 100

Attach 방식이나 Watcher 방식으로 설치했을 때 이미 로딩된 클래스 중에 추적을 위해 BCI 를 새로 수행하게 됩니다. 이때 동시 redefine 하는 클래스 수

1.1.9.3. counter_procfd_enabled

default: false

파일 디스크립트 수를 추적하는 기능을 활성화합니다.

1.1.9.4. counter_netstat_enabled

default: false

NET STAT 상태별 건수를 모니터링합니다. ESTABLISH, CLOSE WAIT FIN WAIT 등 상태별 건수를 수집합니다.

1.1.9.5. realtime_user_thinktime_max

default: 300000

실시간 사용자를 측정할 때 사용자로 인정되는 호출간격을 지정합니다.

1.1.9.6. time_sync_interval_ms

default: 300000

에이전트는 이 옵션에서 지정한 시간에 한번씩 서버와 통신하면서 시간을 맞춘다.

1.1.9.7. detect_deadlock_enabled

default: false

실행중인 쓰레드간 deadlock 이 있는지를 확인하고 이벤트를 발생합니다. deadlock 발생여부는 5 초마다 체크하지만 deadlock 이벤트는 한시간에 한번만 발생시킨다.

1.1.9.8. text_reset

default: 0

에이전트는 한번 보낸 텍스트는 다음날까지는 재전송하지 않는다. 그런데 즉시 재전송하고자 할 때 text_reset 값을 임으로 지정하면 전송 기록이 리셋되어 다시 텍스트가 전송됩니다. 이전 값과 다른 int 값을 설정하면 됩니다.

1.1.9.9. auto_oname_enabled

default: false

에이전트 이름(oname)을 서버로부터 자동 부여 받는 기능을 활성화 합니다.

- 적용 시, -Dwhatap.name, -Dwhatap.oname 옵션은 무시됩니다.
- 수집 서버와의 통신을 통해 oname 을 부여 받은 이후, 에이전트의 일반적인 동작을 개시합니다.

1.1.9.10. auto_oname_prefix

default: agent

에이전트 이름을 서버로부터 자동 부여할 때 에이전트 이름의 prefix 보통 업무 명을 사용합니다.

- prefix 일련번호 1~) 부여됩니다.

1.1.9.11. auto_oname_reset

default: 0

에이전트 이름을 자동 부여하면 what.oname 이라는 시스템 환경 변수에 셋트됩니다. 한번 셋트되면 자바 인스턴스가 재기동 될 때까지 유지 되는데 리셋을 원할 때 auto_oname_reset 값을 수정합니다.

* 현재 설정 값과 다른 값으로 변경하면 적용됩니다.

1.1.10. 알림 설정 옵션

1.1.10.1. 재귀적으로 forward 되는 요청에 대한 경고 알림 설정

재귀적으로 forward 되는 요청에 대한 경고 알림 설정

트랜잭션의 재귀 호출 여부 검출을 위한 옵션으로, 단일 트랜잭션으로 부터 파생되는 재귀 호출 횟수를 카운트하여 이벤트 알림을 발행하기 위한 기준을 지정합니다.

- HTTP URL 재귀 호출을 대상으로 함
- jsp:forward 를 통해 재호출 되는 케이스도 카운트에 포함됩니다.
- *recursive_max* = 1000000 (default) 재귀호출 카운트 임계치
- *recursive_event_interval* = 30000 (default) 이벤트 알림 발행 간격

1.1.10.2. 서비스 거절(호출 부하 제한/거절))시 경고

사용자 요청이 block 될 때 이벤트 알림 발행 여부와 간격을 지정합니다.

- *reject_event_enabled* = false (default) 활성화 여부
- *reject_event_interval* = 30000 (default) 이벤트 알림 발행 간격

1.1.10.3. 외부 API호출에서 에러 발생시 경고

외부 API 호출 시 에러가 발생할 경우 이벤트 알림 발행 여부와 간격을 지정함

- *httpc_event_enabled* = false (default) 활성화 여부
- *httpc_event_interval* = 30000 (default) 이벤트 알림 발행 간격

1.1.10.4. 힙 사용량 경고

힙 사용량 임계 도달 시 이벤트 알림 발행 여부와 간격을 지정합니다.

- *heap_event_enabled* = false (default) 활성화 여부
- *heap_event_percent* = 90 (default) 임계치(%)
- *heap_event_duration* = 30000 (default) 임계치(지속시간 밀리초)
- *heap_event_interval* = 300000 (default) 이벤트 알림 발행 간격
- *heap_event_action* = 이벤트 발생 시 실행할 동적 로딩 코드 ($WHATAP_HOME/plugin/ActionScript.x 에 작성한 Java 코드)에 전달할 ID ($id 로 전달됨)

1.1.10.5. 디스크 사용량 경고

디스크 사용량 임계 도달 시 이벤트 알림 발행 여부와 간격을 지정합니다.

- *disk_event_enabled* = false (default) 활성화 여부
- *disk_event_percent* = 90 (default) 임계치(%)
- *disk_event_interval* = 3600000 (default) 이벤트 알림 발행 간격

- *disk_event_action* = 이벤트 발생 시 실행할 동적 로딩 코드($WHATAP_HOME/plugin/ActionScript.x 에 작성한 Java 코드)에 전달할 ID ($id 로 전달됨)

1.1.10.6. CPU 사용량 경고

CPU 사용량 임계 도달 시 이벤트 알림 발행 여부와 간격을 지정합니다.

- *cpu_event_enabled* = false (default) 활성화 여부
- *cpu_event_percent* = 90 (default) 임계치(%)
- *cpu_event_duration* = 10000 (default) 임계치(지속시간 밀리초)
- *cpu_event_interval* = 300000 (default) 이벤트 알림 발행 간격
- *cpu_event_action* = 이벤트 발생 시 실행할 동적 로딩 코드 ($WHATAP_HOME/plugin/ActionScript.x 에 작성한 Java 코드)에 전달할 ID ($id 로 전달됨)

1.1.10.7. DB Connection 중복 할당 경고

DB Connection 이 중복 할당 되었을 때 이벤트 알림 발행 여부와 간격을 지정합니다.

- *dbc_dup_event_enabled* = false (default) 활성화 여부
- *dbc_dup_event_fullstack_enabled* = false (default) 중복할당 될때 Stack 확보 여부를 지정합니다.

1.1.10.8. Exception 발생시 경고

Exception 발생 시 이벤트 알림 발행 여부와 간격을 지정합니다.

- *exception_event_enabled* = false (default) 활성화 여부
- *exception_event_interval* = 60000 (default) 이벤트 알림 발행 간격
- *exception_event_set* = Exception 을 지정합니다. Exception Set 을 지정할 경우 ','로 구분자를 사용합니다.

- *exception_event_action* = 이벤트 발생 시 실행할 동적 로딩 코드 ($WHATAP_HOME/plugin/ActionScript.x 에 작성한 Java 코드)에 전달할 ID ($id 로 전달됨)

1.1.11. 에이전트 명명 옵션

1.1.11.1. priority

1. [Configuration] auto_oname_enabled
2. [JVM Option] -Dwhatap.name
3. [JVM Option] -Dwhatap.oname

1.1.11.2. JVM Options Only

와탭의 에이전트 식별자 (oname) 는 whatap.name (조합 패턴) + port, ip 로 구성됩니다.

1.1.11.3. -Dwhatap.name

whatap.conf 에 auto_oname_enabled 가 설정되어 있지 않은 경우 적용됩니다.
pattern 으로 지정 시 port 및 ip 를 조합합니다.
조합에 사용 가능한 옵션 parenthesis - {type}, {ip0}, {ip1}, {ip2}, {ip3}, {port}, {cmdN}
패턴으로 지정하지 않고 고정 값으로 설정하는 경우, 지정 값으로 에이전트가 식별됩니다.

패턴 옵션	설명	비고

type	애플리케이션 서버 유형	TC:Tomcat SB:SpringBoot JB:JBoss WL:WebLogic WS:Websphere JU:Jeus JT:Jetty AP:Application
ipN	ip address 의 N 번째 자리	
port	애플리케이션 서비스 포트	
pid	애플리케이션 프로세스 ID	애플리케이션 서버 포트 식별 불가 시 활용
cmdN	Java 명령을 통해 전달된 N 번째 파라미터	애플리케이션 서버가 파라미터를 전달받는 경우에 사용가능

1.1.11.4. -Dwhatap.oname

whatap.conf 에 auto_oname_enabled 가 설정되어 있지 않고, Dwhatap.name 미지정 시 -Dwhatap.oname 에 지정한 값이 oname, 즉 애플리케이션의 식별 값으로 적용됩니다.

1.1.11.5. 예외적 옵션

1.1.11.5.1. object_name

whatap.conf 에 설정, -Dwhatap.name 과 동일한 옵션을 설정 파일에 지정하기 위한 옵션으로 -Dwhatap.name 과 동일하게 pattern 을 지정할 수 있습니다.

- 전제 사항 : 1 application server per 1 VM
- 사용 상황 : VM 복제

1.1.12. 부록

1.1.12.1. 데이터베이스 관련 이슈 추적 옵션

1.1.12.2. tomcat_ds_enabled / weblogic_ds_enabled

- JMX 를 통한 datasource pool 정보를 수집

1.1.12.3. dbcp_pool_enabled / hikari_pool_enabled

- datasource 에 접근하여 pool 정보를 수집

1.1.12.4. profile_dbc_close

- DB 접속 close 시 profile 에 스텝 노출

1.1.12.5. trace_dbc_leak_enabled

- DB connection leak 추적 활성화 (staistics > error 에 노출)

1.1.12.6. trace_dbc_leak_fullstack_enabled

- DB connection leak 발생 시 호출 스택 수집 활성화 (staistics > error 에 노출)

1.1.12.7. 에이전트 설정 분리 옵션

1.1.12.8. -Dwhatap.config

2 개 이상의 애플리케이션 서버가 동일 호스트에 탑재되어 있고, 와탭 에이전트를 단일 경로에 설치하여 운영하는 경우, 필요에 따라 에이전트 configuration 을 상이하게 적용해야 할 경우가 발생합니다.
이와 같은 경우, -Dwhatap.config 설정을 통해 와탭 설정 파일의 경로를 지정할 수 있습니다.

- ex) 와탭 에이전트 설치 경로 내에 설정 파일만 추가로 위치시키는 경우
  - -Dwhatap.config=whatap.conf2
- 절대경로는 사용불가하며, 와탭 설치 경로로부터 상대경로상의 파일을 참조합니다.

Thank you

support@whatap.io
(주)와탭랩스

www.whatap.io
서울특별시 강남구 테헤란로 69 길 5 유기타워 11F (06160)
Tel. 02.565.1803
Fax. 0504.848.1803

Monitoring makes ____ easy