



Monitoring makes \_\_\_\_ easy

## Python Application Monitoring Agent Configure

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

# 목차

1.1. Python 애플리케이션 모니터링 .....	5
1.1.1. 소개 .....	5
1.1.2. 에이전트 네트워크 통신에 관한 설정 .....	5
1.1.2.1. license .....	5
1.1.2.2. whatap.server.host .....	5
1.1.2.3. whatap.server.port .....	5
1.1.3. 에이전트 네트워크 내부 통신에 관한 설정 .....	6
1.1.3.1. net_udp_port .....	6
1.1.4. 에이전트 네트워크 외부 통신에 관한 설정 .....	6
1.1.4.1. tcp_so_timeout .....	6
1.1.4.2. tcp_connection_timeout .....	6
1.1.4.3. net_send_max_bytes .....	6
1.1.4.4. net_send_buffer_size .....	7
1.1.5. 애플리케이션 등록에 관한 설정 .....	7
1.1.5.1. object_name .....	7
1.1.5.2. app_name .....	7
1.1.5.3. app_process_name .....	8
1.1.5.4. auto_otype_enabled .....	8
1.1.5.5. auto_otype_prefix .....	8
1.1.5.6. auto_otype_reset .....	8
1.1.6. 에이전트 기능 제어에 관한 설정 .....	8
1.1.6.1. enabled .....	9
1.1.6.2. plugin .....	9
1.1.6.3. transaction_enabled .....	9
1.1.6.4. counter_enabled .....	10
1.1.6.5. stat_enabled .....	10
1.1.6.6. active_stack_enabled .....	10
1.1.7. 보안 설정에 관한 설정 .....	10
1.1.7.1. cypher_level .....	10
1.1.7.2. encrypt_level .....	11
1.1.8. 트랜잭션 프로파일 데이터에 관한 설정 .....	11
1.1.8.1. profile_http_header_enabled .....	11

# 목차

1.1.8.2.	profile_httpheader_urlprefix.....	11
1.1.8.3.	profile_http_parameter_enabled.....	12
1.1.8.4.	profile_http_parameter_urlprefix.....	12
1.1.8.5.	profile_connection_open_enabled.....	12
1.1.8.6.	profile_dbc_close(*).....	12
1.1.8.7.	profile_step_normal_count.....	12
1.1.8.8.	profile_step_heavy_count.....	13
1.1.8.9.	profile_step_max_count.....	13
1.1.8.10.	profile_step_heavy_time.....	13
1.1.8.11.	profile_basetime.....	13
1.1.8.12.	query_string_enabled.....	13
1.1.8.13.	query_string_urls.....	14
1.1.8.14.	profile_sql_param_enabled.....	14
1.1.8.15.	profile_sql_resource_enabled.....	14
1.1.8.16.	profile_method_resource_enabled.....	14
1.1.8.17.	profile_httpc_resource_enabled.....	15
1.1.8.18.	profile_position_sql.....	15
1.1.8.19.	profile_position_httpc.....	15
1.1.8.20.	profile_position_method.....	15
1.1.8.21.	profile_position_depth.....	16
1.1.8.22.	profile_error_sql_fetch_max.....	16
1.1.8.23.	profile_error_sql_time_max.....	16
1.1.9.	사용자 추적을 위한 설정.....	16
1.1.9.1.	trace_user_enabled(*).....	16
1.1.9.2.	trace_user_using_ip(*).....	17
1.1.9.3.	trace_user_header_ticket.....	17
1.1.9.4.	trace_user_cookie_limit.....	17
1.1.9.5.	trace_http_client_ip_header_key.....	17
1.1.10.	백그라운드 트랜잭션 추적을 위한 설정.....	18
1.1.10.1.	trace_auto_transaction_enabled.....	18
1.1.10.2.	trace_auto_transaction_backstack_enabled.....	18
1.1.10.3.	trace_background_socket_enabled.....	18
1.1.11.	트랜잭션 추적을 위한 설정.....	18

# 목차

1.1.11.1.	trace_transaction_name_header_key .....	18
1.1.11.2.	trace_transaction_name_key .....	19
1.1.11.3.	trace_error_callstack_depth .....	19
1.1.11.4.	trace_active_callstack_depth .....	19
1.1.11.5.	trace_active_transaction_slow_time .....	19
1.1.11.6.	trace_active_transaction_very_slow_time .....	19
1.1.11.7.	trace_active_transaction_lost_time .....	20
1.1.11.8.	trace_dbc_leak_enabled .....	20
1.1.11.9.	trace_dbc_leak_fullstack_enabled .....	20
1.1.11.10.	debug_dbc_stack_enabled .....	20
1.1.11.11.	trace_gtx_rate .....	20
1.1.11.12.	web_static_content_extensions(*) .....	21
1.1.12.	운영을 위한 설정 .....	21
1.1.12.1.	active_stack_second .....	21
1.1.12.2.	counter_procfld_enabled .....	21
1.1.12.3.	counter_netstat_enabled .....	21
1.1.12.4.	realtime_user_thinktime_max .....	21
1.1.12.5.	time_sync_interval_ms .....	22
1.1.12.6.	detect_deadlock_enabled .....	22
1.1.12.7.	text_reset .....	22
1.1.12.8.	trace_auto_normalize_enabled(*) .....	22
1.1.12.9.	trace_normalize_enabled .....	22
1.1.12.10.	trace_normalize_urls .....	23
1.1.12.11.	trace_httpc_normalize_enabled .....	23
1.1.12.12.	trace_httpc_normalize_urls .....	23
1.1.12.13.	trace_sql_normalize_enabled .....	23
1.1.13.	로그를 위한 설정 .....	24
1.1.13.1.	log_rotation_enabled .....	24
1.1.13.2.	log_keep_days .....	24

## 1.1. Python 애플리케이션 모니터링

### 1.1.1. 소개

해당 가이드는 Python 애플리케이션 모니터링을 할 때에 설정할 수 있는 옵션에 대해 설명합니다. (\*) 표시가 있는 옵션은 애플리케이션을 재시작 하여야만 적용되는 옵션입니다.

### 1.1.2. 에이전트 네트워크 통신에 관한 설정

#### 1.1.2.1. license

**default:** ""

에이전트가 속한 프로젝트의 PCODE, 서버와 보안 통신을 위한 암호키를 포함하고 있는 보안 문자열을 설정한다. 라이선스 키가 없으면 서버와 통신할 수 없습니다

#### 1.1.2.2. whatap.server.host

**default:** "127.0.0.1", "127.0.0.1"

수집서버주소(아이피)를 지정합니다. 콤마(,)로 분리하여 하나 혹은 두개를 지정할 수 있습니다.

두개를 지정하는 경우, 첫번째 수집서버가 죽는 경우에 대해 대비가 가능합니다.

#### 1.1.2.3. whatap.server.port

**default:** 6600

# WhaTap

수집서버 포트를 지정합니다. 포트는 하나만 지정할 수 있습니다. 따라서 whatap.server.host 에 지정된 수집서버들은 동일 포트를 사용해야합니다.

## 1.1.3. 에이전트 네트워크 내부 통신에 관한 설정

### 1.1.3.1. net\_udp\_port

**default: 6600**

와탭 에이전트는 수집한 데이터를 UDP 를 거쳐 TCP 를 통해 서버로 전송을 하는데,

처음 UDP 서버의 포트를 지정할 수 있습니다. 기본값으로 제공되는 6600 포트가 사용 중일 때 이 옵션을 사용합니다.

## 1.1.4. 에이전트 네트워크 외부 통신에 관한 설정

### 1.1.4.1. tcp\_so\_timeout

**default: 60000**

수집서버와 통신 할 때 TCP 세션의 idle 타임아웃 값을 지정합니다.

### 1.1.4.2. tcp\_connection\_timeout

**default: 5000**

수집서버와 통신 세션을 연결할 때 연결 지연 가능 시간을 지정합니다.

### 1.1.4.3. net\_send\_max\_bytes

**default: 5 \* 1024 \* 1024**

# WhaTap

에이전트가 데이터를 수집하고 네트워크로 한번에 전송할 수 있는 최대 바이트 크기입니다.

## 1.1.4.4. net\_send\_buffer\_size

**default: 1024**

데이터 전송을 하기 위해 가지고 있는 최대 바이트 크기입니다.

## 1.1.5. 애플리케이션 등록에 관한 설정

### 1.1.5.1. object\_name

**default: {type}-{ip2}-{ip3}-{process}**

애플리케이션을 식별하는 에이전트 이름(ONAME)체계입니다. ONAME 을 토대로 OID 가 생성됩니다. 자동으로 부여된 이름 대신 직접 애플리케이션 서버 별로 이름을 부여하고 싶은 경우에는 옵션 설정을 통해 패턴을 지정하거나, 정적으로 애플리케이션 명을 지정할 수도 있습니다.

설정	설명
type	app_name
ip#	Ip 를 .으로 나누었을 때 #번째 자리 (시작은 0 부터)
process	app_process_name
hostname	호스트명

### 1.1.5.2. app\_name

**default: ""**

애플리케이션을 식별하는 에이전트 이름(ONAME)체계에 사용되는 애플리케이션 명.

object\_name 의 {type}에 해당하는 값입니다.

# WhaTap

## 1.1.5.3. app\_process\_name

**default:** ""

애플리케이션을 식별하는 에이전트 이름 체계에 사용되는 애플리케이션 프로세스 명 멀티 프로세스 방식에서 애플리케이션 서버의 CPU, Heap Memory 등을 계산할 때 사용됩니다.. object\_name 의 {process}에 해당하는 값입니다.

## 1.1.5.4. auto\_otype\_name\_enabled

TODO: 지원 예정입니다.

**default:** false

에이전트 이름(ONAME)을 서버로부터 자동 부여 받는 기능을 활성화합니다.

## 1.1.5.5. auto\_otype\_name\_prefix

TODO: 지원 예정입니다.

**default:** "agent"

에이전트 이름(ONAME)을 서버로부터 자동 부여할 때, 에이전트 이름의 prefix.

이 값을 설정하는 경우 에이전트의 이름은 prefix + 일련번호(1~)로 부여됩니다.

## 1.1.5.6. auto\_otype\_name\_reset

TODO: 지원 예정입니다.

**default:** 0

에이전트 이름을 자동부여 하면, whatap.otype\_name 이라는 시스템 환경 변수가 설정되는데, 이 환경변수를 리셋하고 싶은 경우 설정합니다.

## 1.1.6. 에이전트 기능 제어에 관한 설정



# WhaTap

## 1.1.6.1. enabled

**default: true**

데이터 추적에 대한 전체 기능(트랜잭션, 성능 카운터, 통계 데이터 수집)을 활성화 합니다. 단, 값이 비활성화 되어도 서버와 최소한의 통신을 유지하기 위한 정보는 전송됩니다. 서버와 최소한의 통신을 유지하기 위한 정보에는 환경 정보, 설정 정보 등이 있습니다.

## 1.1.6.2. plugin

**default: false**

기본적으로 제공하는 트랜잭션 추적 모듈 이외에 사용자가 원하는 모듈 추적에 대한 기능을 플러그인 형태로 제공한다. 옵션을 활성화 하면 \$WHATAP\_HOME 으로 지정한 경로에 plugin.json 파일이 자동 생성되며, 형식에 맞게 내용을 추가하면 사용자가 메뉴얼하게 모듈을 추가하여 특정 모듈의 데이터를 추적할 수 있다. 사용자 별로 원하는 모듈을 추적하고자 할 때 활용한다. 기본 모듈과 중복이 되는 경우, 사용자의 설정이 우선시 된다. [module\_name]: 추적하고자 하는 대상의 모듈 명. import 하는 모듈 명 이기도 하다.

설정	설명
[class_name]	추적하고자 하는 대상의 클래스 명이 없다면 “으로 사용한다.
[def_name]	추적하고자 하는 대상이다.
args_indexes	추적하고자 하는 대상의 아규먼트 인덱스. 여러 개일 경우로 구분한다.
kwargs	추적하고자 하는 대상의 키워드 명. 여러 개일 경우로 구분한다.

## 1.1.6.3. transaction\_enabled

**default: true**

# WhaTap

트랜잭션 추적 기능을 활성화 합니다. 트랜잭션 데이터는 Hitmap 에서 확인 가능합니다. 단, enable 이 false 면 무시됩니다.

## 1.1.6.4. counter\_enabled

**default: true**

성능 카운터 추적을 활성화 합니다. 성능 카운터는 액티브 트랜잭션, 사용자, CPU, Heap Memory, SQL, 외부 HTTP 호출 등의 데이터를 포함합니다. 단, enable 이 false 면 무시됩니다.

## 1.1.6.5. stat\_enabled

**default: true**

통계 정보 추적을 활성화 합니다. 5 분마다 트랜잭션, SQL Query, 외부 HTTP 호출, User Agent, Client IP 등의 통계 데이터가 수집되는데 이들 정보의 수집이 중단됩니다. 단, enable 이 false 면 무시됩니다.

## 1.1.6.6. active\_stack\_enabled

TODO: 지원 예정입니다.

**default: true**

액티브 스택 추적을 활성화 합니다. 참고로, counter\_enabled 값이 비활성화 된 경우 액티브 트랜잭션 데이터가 수집되지 않으므로 액티브 스택 또한 수집되지 않습니다.

## 1.1.7. 보안 설정에 관한 설정

### 1.1.7.1. cypher\_level

TODO: 지원 예정입니다.

**default: 128**

AES 보안 알고리즘에 대한 암호 레벨을 지정합니다.

# WhaTap

(값은 128, 256 중 하나를 사용할 수 있습니다.)

## 1.1.7.2. encrypt\_level

TODO: 지원 예정입니다.

**default: 2**

와탭은 데이터에 따라 다른 암호화 알고리즘을 적용합니다. 주로 텍스트 데이터는 보안 레벨을 높게 적용하고 단순 숫자 데이터는 보안 레벨을 낮게 적용합니다. 이러한 보안 레벨을 적용할 때 일괄적으로 전체적인 보안 레벨을 보다 높게 혹은 보다 낮게 적용할 수 있는데 그것을 지정하는 옵션입니다.

(값은 1, 2, 3 중에 하나를 사용합니다.)

## 1.1.8. 트랜잭션 프로파일 데이터에 관한 설정

### 1.1.8.1. profile\_http\_header\_enabled

TODO: 지원 예정입니다.

**default: false**

프로파일에 HTTP 헤더 정보를 출력합니다.

헤더 수집의 여부와 무관하게 헤더를 프로파일에 노출할 것인가에 대한 설정입니다.

### 1.1.8.2. profile\_httpheader\_urlprefix

TODO: 지원 예정입니다.

**default: "/"**

HTTP 헤더 정보를 프로파일에 출력 할 때, 출력하고자 하는 대상 URL 의 prefix.

# WhaTap

## 1.1.8.3. profile\_http\_parameter\_enabled

TODO: 지원 예정입니다.

**default: false**

프로파일에 HTTP 파라미터 정보를 출력한다. 단, 파라미터는 별도 보안키를 입력해야 조회 할 수 있습니다. 보안키는 \$WHATAP\_HOME/paramkey.txt 파일의 내용을 변경하여 수정이 가능합니다.

## 1.1.8.4. profile\_http\_parameter\_urlprefix

TODO: 지원 예정입니다.

**default: "/"**

HTTP 파라미터 정보를 프로파일에 출력 할 때, 출력하고자 하는 대상 URL 의 prefix.

## 1.1.8.5. profile\_connection\_open\_enabled

TODO: 지원 예정입니다.

**default: true**

프로파일에 DB Connection 오픈에 대한 정보를 출력합니다.

## 1.1.8.6. profile\_dbc\_close(\*)

**default: false**

profile\_connection\_open\_enabled 이 활성화 되어 있는 경우, 프로파일에 DB Connection 이 close 될 때의 정보를 출력합니다.

## 1.1.8.7. profile\_step\_normal\_count

**default: 800**

프로파일 기본 스텝 수를 제한합니다.

# WhaTap

## 1.1.8.8. profile\_step\_heavy\_count

**default: 1000**

프로파일 기본 스텝을 초과한 경우, 최대 heavy 스텝 수를 제한합니다. profile\_step\_normal\_count 에서 profile\_step\_heavy\_time 을 초과하는 스텝을 카운팅합니다. 따라서, 응답시간이 없는 스텝들은 수집되지 않습니다.

## 1.1.8.9. profile\_step\_max\_count

**default: 1024**

프로파일 스텝의 최대 수를 설정합니다. 수집된 프로파일 스텝 수가 이 값을 초과하면 이후 수집되는 스텝들은 모두 버려집니다.

profile\_step\_heavy\_count 을 최대 1000 으로 설정한다면 profile\_step\_max\_count 만큼 액티브 스택이 수집됩니다.

## 1.1.8.10. profile\_step\_heavy\_time

**default: 100**

프로파일 기본 스텝의 수가 profile\_step\_normal\_count 를 초과 하는 경우, 이 값 이상의 응답시간을 가지는 스텝들을 profile\_step\_heavy\_count 만큼 수집합니다.

## 1.1.8.11. profile\_basetime

**default: 0**

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

## 1.1.8.12. query\_string\_enabled

**default: false**

# WhaTap

트랜잭션 URL의 쿼리 스트링을 함께 수집하는 기능을 활성화합니다.

## 1.1.8.13. query\_string\_urls

**default:** “

트랜잭션에서 쿼리 스트링을 수집할 URL들을 등록합니다. 여러 개를 등록할 때는 콤마(,)를 사용합니다.

## 1.1.8.14. profile\_sql\_param\_enabled

TODO: 지원 예정입니다.

**default:** false

SQL 파라미터를 수집합니다. 단, 파라미터는 암호화 되어 있어 별도의 보안키가 필요합니다. 보안키는 \$WHATAP\_HOME/paramkey.txt 파일의 내용을 변경하여 수정이 가능합니다.

## 1.1.8.15. profile\_sql\_resource\_enabled

**default:** false

프로파일에서 SQL 스텝이 수집될 때,

트랜잭션 시작부터 해당 스텝까지 사용한 CPU와 Memory 사용량을 추적합니다.

트랜잭션에서 리소스들의 사용량을 확인할 때 구간으로 나누어 추적이 용이합니다.

## 1.1.8.16. profile\_method\_resource\_enabled

TODO: 지원 예정입니다.

**default:** false

프로파일에서 Method 스텝이 수집될 때,

트랜잭션 시작부터 해당 스텝까지 사용한 CPU와 Memory 사용량을 추적합니다.

# WhaTap

트랜잭션에서 리소스들의 사용량을 확인할 때 구간으로 나누어 추적이 용이합니다.

## 1.1.8.17. profile\_httpc\_resource\_enabled

**default:** false

프로파일에서 외부 HTTP 호출(HTTP Call) 스텝이 수집될 때, 트랜잭션 시작부터 해당 스텝까지 사용한 CPU와 Memory 사용량을 추적합니다. 트랜잭션에서 리소스들의 사용량을 확인할 때 구간으로 나누어 추적이 용이합니다.

## 1.1.8.18. profile\_position\_sql

TODO: 지원 예정입니다.

**default:** ""

이 옵션에서 지정한 SQL이 수행되면 어디서 수행되었는지에 대한 스택을 프로파일에 같이 출력합니다.

## 1.1.8.19. profile\_position\_httpc

TODO: 지원 예정입니다.

**default:** ""

이 옵션에서 지정한 외부 HTTP 호출이 수행되면 어디서 수행되었는지에 대한 스택을 프로파일에 같이 출력합니다. 알 수 없는 외부 HTTP 호출(HTTPCall)을 프로파일링하여, 그 위치를 찾아 낼 수 있습니다.

## 1.1.8.20. profile\_position\_method

TODO: 지원 예정입니다.

**default:** ""

# WhaTap

이 옵션에서 지정한 Method 가 수행되면 어디서 수행되었는지에 대한 스택을 프로파일링에 같이 출력합니다. 프로파일링 되는 메소드가 어떻게 호출되는지 디버깅 하고자 할 때 설정합니다.

## 1.1.8.21. profile\_position\_depth

TODO: 지원 예정입니다.

**default: 50**

포지션 추적을 위해 스택을 덤프할 때 스택 라인수를 지정합니다.

## 1.1.8.22. profile\_error\_sql\_fetch\_max

TODO: 지원 예정입니다.

**default: 10000**

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

## 1.1.8.23. profile\_error\_sql\_time\_max

TODO: 지원 예정입니다.

**default: 30000**

SQL 수행 후 수행시간이 여기서 지정한 값을 초과하면 TOO SLOW 조회 에러로 처리됩니다.

## 1.1.9. 사용자 추적을 위한 설정

### 1.1.9.1. trace\_user\_enabled(\*)

**default: true**

실시간 사용자를 추적할지를 결정합니다. 이 값을 활성화 하는 경우 기본값은 IP 가 됩니다.



# WhaTap

## 1.1.9.2. trace\_user\_using\_ip(\*)

**default: true**

실시간 사용자의 구분을 IP 로 하고자 하는 경우 설정합니다. 이 데이터는 Real Time User 에서 확인 가능합니다. 단, user\_header\_ticket 옵션과 배타적으로 설정을 동시에 적용할 수 없습니다.

## 1.1.9.3. trace\_user\_header\_ticket

TODO: 지원 예정입니다.

**default: ""**

사용자의 구분을 HTTP 헤더의 특정 값으로 구분하고 싶을 때 사용 할 HTTP 키를 지정합니다. 단, trace\_user\_using\_ip 배타적으로 설정을 동시에 적용할 수 없습니다.

## 1.1.9.4. trace\_user\_cookie\_limit

TODO: 지원 예정입니다.

**default: 2048**

사용자 구분을 기본 값인 WHATAP COOKIE 로 할 때, 쿠키의 크기가 너무 많이 지는 경우 오버플로우를 피하기 위해 쿠키의 크기를 제한합니다.

## 1.1.9.5. trace\_http\_client\_ip\_header\_key

TODO: 지원 예정입니다.

**default: ""**

사용자의 실제 접속 아이피가 HTTP 헤더에 별도로 전달 되는 경우, 해당 헤더 키를 지정하여 사용자를 구분합니다. Client IP 를 특정하기 위한 정보로 활용됩니다. 취득된 값으로 remote address 를 대체합니다.

# WhaTap

## 1.1.10. 백그라운드 트랜잭션 추적을 위한 설정

### 1.1.10.1. trace\_auto\_transaction\_enabled

TODO: 지원 예정입니다.

**default: false**

백그라운드 호출을 하나의 트랜잭션으로 수집하여 데이터를 추적하고 싶을 때, 트랜잭션의 END POINT 를 설정 할 수 있습니다. 프로덕션 보다는 주로 개발 환경에서 사용합니다.

### 1.1.10.2. trace\_auto\_transaction\_backstack\_enabled

TODO: 지원 예정입니다.

**default: true**

trace\_auto\_transaction\_enabled 설정이 활성화 되어 있을 때, 정의된 트랜잭션의 시작점에서 스택을 남김으로 진입점이 어디인지 추적이 가능합니다.

### 1.1.10.3. trace\_background\_socket\_enabled

TODO: 지원 예정입니다.

**default: true**

기본적으로 트랜잭션이 시작된 상황에서만 소켓(TCP)연결 오픈에 대한 데이터를 추적하는데, 트랜잭션이 아닌 백그라운드 쓰레드에 의한 소켓이 오픈될 때도 데이터를 추적이 가능합니다.

## 1.1.11. 트랜잭션 추적을 위한 설정

### 1.1.11.1. trace\_transaction\_name\_header\_key

# WhaTap

TODO: 지원 예정입니다.

**default:** ""

설정 된 HTTP 헤더의 키 값을 트랜잭션 URL 에 추가합니다.

동일한 URL 에서 HTTP 헤더의 키를 구분하여 트랜잭션을 세밀하게 나누어 데이터추적이 가능합니다.

## 1.1.11.2. trace\_transaction\_name\_key

TODO: 지원 예정입니다.

**default:** ""

설정 된 HTTP 요청 파라미터의 값을 트랜잭션 URL 에 추가합니다.

동일한 URL 에서 파라미터를 구분하여 트랜잭션을 세밀하게 나누어 데이터추적이 가능합니다.

## 1.1.11.3. trace\_error\_callstack\_depth

**default:** 50

트랜잭션에서 에러 콜 스택을 수집할 때, 콜스택 최대 라인수를 지정합니다.

이 데이터는 에러 통계데이터에서 조회 할 수 있습니다.

## 1.1.11.4. trace\_active\_callstack\_depth

**default:** 50

트랜잭션에서 액티브 스택을 수집할 때, 콜스택 최대 라인수를 지정합니다.

## 1.1.11.5. trace\_active\_transaction\_slow\_time

**default:** 3000

액티브 트랜잭션의 아카이클라이저에서 slow 구간의 응답기준을 지정합니다.

## 1.1.11.6. trace\_active\_transaction\_very\_slow\_time

**default:** 8000

# WhaTap

액티브 트랜잭션의 아카이클라이저에서 very slow 구간의 응답기준을 지정합니다.

## 1.1.11.7. trace\_active\_transaction\_lost\_time

**default: 300000 (5min)**

트랜잭션의 종료를 기다리는 제한 시간. 5 분안에 트랜잭션이 끝나지 않는 경우 트랜잭션을 정보를 더이상 수집하지 않습니다.

## 1.1.11.8. trace\_dbc\_leak\_enabled

TODO: 지원 예정입니다.

**default: false**

DB Connection Leak 을 추적하는 기능을 활성화 합니다.

## 1.1.11.9. trace\_dbc\_leak\_fullstack\_enabled

TODO: 지원 예정입니다.

**default: false**

DB Connection Leak 을 추적하는 기능이 활성화 되어 있을 때, 사용 후 반환하지 않은 트랜잭션의 DB Connection Leak 위치를 확인 하기 위해 풀 스택을 확인 할 수 있습니다. 부분적인 CPU 사용량이 몇 % 정도 증가할 수 있으므로 CPU 사용량이 높은 시스템에서는 Peak 타임을 피해서 활성화 하는 것을 권장합니다.

## 1.1.11.10. debug\_dbc\_stack\_enabled

TODO: 지원 예정입니다.

**default: false**

DB Connection 오픈이 될 때, 스택을 프로파일에 출력 할 수 있습니다.

## 1.1.11.11. trace\_gtx\_rate

# WhaTap

TODO: 지원 예정입니다.

**default: 0**

와탭 에이전트가 등록된 애플리케이션 들의 관계를 확인 하기 위해 발급되는 gtx 에 대해, 관계를 파악하기 위한 용도로 사용될 gtx 사용율.

## 1.1.11.12. web\_static\_content\_extensions(\*)

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

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

## 1.1.12. 운영을 위한 설정

### 1.1.12.1. active\_stack\_second

**default: 10**

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

### 1.1.12.2. counter\_procfld\_enabled

TODO: 지원 예정입니다.

**default: false**

파일 디스크립터 수를 추적하는 기능을 활성화합니다.

### 1.1.12.3. counter\_netstat\_enabled

TODO: 지원 예정입니다.

**default: false**

NETSTAT 상태별 건수를 모니터링 합니다.

### 1.1.12.4. realtime\_user\_thinktime\_max

# WhaTap

**default: 300000**

실시간 사용자를 측정할 때 사용자로 인정되는 호출간격을 지정합니다.

## 1.1.12.5. time\_sync\_interval\_ms

**default: 30000**

에이전트는 이 옵션에 지정한 시간에 한번씩 서버와 통신하면서 시간을 맞춥니다.

## 1.1.12.6. detect\_deadlock\_enabled

TODO: 지원 예정입니다.

**default: false**

실행 중인 스레드간에 deadlock 이 있는지를 확인하고 이벤트를 발생합니다. deadlock 발생여부는 5 초마다 체크하지만 deadlock 이벤트는 한시간에 한번만 발생시킵니다.

## 1.1.12.7. text\_reset

**default: 0**

에이전트는 한번 보낸 텍스트는 다음날까지 재전송하지 않습니다. 그런데 즉시 재전송하고자 할 때, 이 값을 이전과 다른 int 값으로 설정을 하게 되면 전송기록이 리셋되어 텍스트가 다시 전송됩니다.

## 1.1.12.8. trace\_auto\_normalize\_enabled(\*)

**default: true**

트랜잭션 URL 을 정규화 할 때 패턴 값을 추출하여 자동으로 파싱하는 기능을 활성화합니다.

## 1.1.12.9. trace\_normalize\_enabled

**default: true**

# WhaTap

트랜잭션 URL 을 파싱하여 정규화 하는 기능을 활성화합니다.

## 1.1.12.10.trace\_normalize\_urls

**default:**

트랜잭션 URL 을 파싱하여 정규화 합니다. 호출 URL 패턴을 파싱하여 패스 파라미터를 제거합니다. 예를 들어, /a/{v}/b 라고 선언하면 /a/123/b -> /a/{v}/b 로 치환합니다. 여러 개를 등록 할 때는 콤마(,)를 사용합니다. /a/\*/b 로 사용하여 특정 형식의 패스 파라미터를 제거하여 수집할 수 있습니다.

## 1.1.12.11.trace\_httpc\_normalize\_enabled

**default: true**

외부 HTTP 호출(HTTP Call) URL 을 파싱하여 정규화 하는 기능을 활성화합니다.

## 1.1.12.12.trace\_httpc\_normalize\_urls

**default:**

외부 HTTP 호출하는 URL 을 파싱하여 정규화합니다. 호출 URL 패턴을 파싱하여 패스 파라미터를 제거합니다. 예를 들면, /a/{v}/b 라고 선언하면 /a/123/b -> /a/{v}/b 로 치환한다. 여러 개를 등록 할 때는 콤마(,)를 사용합니다. /a/\*/b 로 사용하여 특정 형식의 패스 파라미터를 제거하여 수집할 수 있습니다.

## 1.1.12.13.trace\_sql\_normalize\_enabled

**default: true**

SQL 문에서 리터럴 부분을 추출하여 SQL 문을 정규화 하는 기능을 활성화합니다.

# WhaTap

## 1.1.13. 로그를 위한 설정

### 1.1.13.1. log\_rotation\_enabled

**default: true**

에이전트 로그파일을 매일 변경합니다.

(로그파일 기본 경로: \$WHATAP\_HOME/logs/whatap-xxx.log)

### 1.1.13.2. log\_keep\_days

**default: 7**

로그파일 보관기간을 설정합니다.



# Thank you



[support@whatap.io](mailto:support@whatap.io)

(주)와탭랩스

[www.whatap.io](http://www.whatap.io)

서울특별시 강남구 테헤란로 69 길 5 유기타워 11F (06160)

Tel. 02.565.1803

Fax. 0504.848.1803

**Monitoring makes \_\_\_\_ easy**