



# Java Agent v2.1.0

2021-12-15

## 에이전트 관리 체계 수정

### 에이전트 버전체계 변경

에이전트 버전 체계를 2.1.x로 변경 하였습니다.

### 자바 에이전트 배포 패키지 구성 변경

whatap.agent-2.1.0 부터 와탭랩스 자바 에이전트 패키지 구성을 변경 하였습니다.

- [whatap.agent-2.1.0.jar](#) : 에이전트 코어 jar
- [whatap.agent.proxy-2.1.0.jar](#) : 와탭 서버와 직접연결이 어려워 우회 연결 지원
- [whatap-logsink-lz4-1.7.0.jar](#) : 로그 모니터링(LogSink)에서 데이터를 압축 하기 위한 플러그 모듈 (기본은 JDK에서 기본으로 제공하는 GZIP 사용)

### JAVA 지원 범위 확대

Asm 라이브러리 v9.2를 활용하여 와탭랩스 자바 에이전트가 지원하는 범위를 기존 java14에서 java17로 확대 하였습니다.

### 멀티서버 추적(mtrace) 자동화

#### 멀티 서버 연결추적 기본정책 변경

에이전트간 호출 성능 추적 기본값을 on으로 변경 하였습니다.

```
whatap.conf
```

```
tx_caller_meter_enabled = true  
sql_dbc_meter_enabled = true
```

```
httpc_host_meter_enabled = true
actx_meter_enabled = true
actx_slice_meter_enabled = true

tx_caller_meter_pkind_enabled = true
actx_meter_pkind_enabled = true
```

## 멀티 서버 추적 자동 적용

User Agent를 기반으로 브라우저가 호출하는 트랜잭션에 한하여 자동으로 처리하도록 변경 하였습니다. (User Agent가 'Moz'로 시작하는 경우에 한하여 mtrace\_rate가 동작하는 방식)

```
whatap.conf
```

```
mtrace_auto_enabled=true (기본값)
mtraceable_user_agent=Moz (기본값)
```

## 멀티서버 추적 비율 자동 조절

명시적으로 mtrace\_rate를 지정하지 않은 경우 500tps 이상의 환경에서는 자동으로 rate를 낮추는 기능을 추가 하였습니다. 단일 프로세스에서 500tps 기준으로 mtrace\_rate는 기본 10%이며 tps가 높아질수록 동일 비율로 mtrace\_rate를 자동 조절 합니다.

## 서버(에이전트)간 의존성 정보 수집내용 변경

서로 다른 프로젝트에 속한 에이전트간의 호출이 발생한 경우 caller 정보를 OID 단위로 수집하도록 수정하였습니다.

이전 버전에서는 PKIND(pcode+okind) 단위로 수집하던 것을 POID(pcode+oid)단위로 수집하도록 수정하였습니다.

## 추적 정보 개선

### DB 연결URL 에서 '?' 이하를 제거

SQL 수행 성능을 추적을 위해 JDBC 연결정보를 수집할때 '?' 이하를 제거 하였습니다.

### URL 정규화 작업에 ContextPath 반영

URL 정규화 작업을 수행할 때 ContextPath를 반영할 수 있도록 변경 하였습니다.

## URL 패턴을 서버에 설정하도록 변경

와탭랩스에서는 Service 혹은 HttpClient URL의 패턴을 와탭서버에서 설정하여 수집하도록 변경하였습니다.

URL 패턴 설정 대상

- Service URL
- HttpClient URL

ⓘ 기존 에이전트 설정 방식으로는 쿠버네티스 환경에서 변경에 어려움

## 에이전트 로그에 ONAME 출력

에이전트 로그의 출력 내용을 (날짜)(PID)(메세지) 포맷을 (날짜)(에이전트이름)(메세지) 포맷으로 변경하였습니다.

```
20211224 05:12:31(GMT) [JA-161-111][UrlNorm] Service PathTree done. PathTree=0
20211224 05:12:31(GMT) [JA-161-111][HttpClientNorm] PathTree build done. PathTree=0
```

## 사용자 정의 카운터 추가

App Extra Counter를 추적하기 플러그인을 사용할 수 있도록 수정 하였습니다. 단, 에이전트에서 플러그인 ExtraCounter.x 구현이 필요합니다.

- `${WHATAP}/plugin/ExtraCounter.x`

```
$pack.put("mycount",100);
$pack.put("myrandom",(float)( whatap.util.KeyGen.next()%100));
```

## 애플리케이션 로그 모니터링

애플리케이션 로그를 통합 모니터링 하는 기능을 추가 하였습니다. 자바 애플리케이션에서 로그 모니터링 기능을 사용하기 위해서는 whatap.agent-2.1.0 이후 버전이 필요합니다.

## ! whatap.agent-2.1.0 이후 버전

와탭의 애플리케이션 로그 모니터링의 기능 명칭은 로그싱크(LogSink) 입니다.

### 로그 모니터링 기능 활성화

로그 모니터링을 위해서는 [whatap.conf](#) 설정 파일에서 `logsink_enabled` 를 `true` 로 설정 해야 합니다.

```
whatap.conf
```

```
logsink_enabled=true
```

자바 애플리케이션에서 발생하는 System.out, System.err 그리고 프레임워크에서 구현한 LOGBACK, LOG4J 로그를 수집 합니다. 기본값은 `logsink_enabled` 의 설정과 동일 합니다.

```
whatap.conf
```

```
logsink_stdout_enabled=logsink_enabled  
logsink_stderr_enabled=logsink_enabled  
logsink_logback_enabled=logsink_enabled  
logsink_log4j_enabled=logsink_enabled  
logsink_tomcat_enabled=logsink_enabled  
logsink_custom_enabled=logsink_enabled
```

### 프레임워크에서 로그내용 수집

프레임워크에서 수집하는 로그를 모니터링 하기 위해서는 모듈에서 로그를 가로채기 위한 설정과 플러그인을 추가해야 합니다. 프레임워크에서 수집하는 로그의 기본 카테고리 이름을 'AppLog'로 설정합니다.

기본값은 `hooklog_enabled` 의 설정과 동일 합니다.

```
whatap.conf
```

```
hooklog_logback_enabled=hooklog_enabled;  
hooklog_log4j_enabled=hooklog_enabled;  
hooklog_tomcat_enabled=hooklog_enabled;  
hooklog_custom_methods= (사용자 메소드에서 로그 수집 설정)
```

다만 `hooklog_enabled` 는 기본값으로 `logsink_enabled` 값이 적용되며 기동전에 `true` 로 설정할 필요가

있습니다.

운영중에 `hooklog_enabled=true` 로 변경한다면 재기동 후에 로그 수집이 가능합니다.

! `logsink_enabled=true` 변경 후 재기동

## 사용자 메소드에서 로그 수집

`hooklog_custom_methods` 는 임의의 로그 프레임워크의 내용을 전달합니다. 사이트에서 개별로 만든 로그 모듈의 로그를 추적할때 사용합니다.

```
whatap.conf
```

```
hooklog_custom_methods=com.greatshop.common.Log.print
```

## 로그 전송설정

단일 라인의 최대크기와 전송시 큐 사이즈 설정이 가능합니다.

```
whatap.conf
```

```
logsink_queue_size=1000  
logsink_line_size=512
```

## 로그 압축 전송 활성화/디버깅

`logsink_zip_enabled=true` 설정으로 로그를 압축 전송할 수 있습니다.

```
whatap.conf
```

```
logsink_zip_enabled=false  
debug_logsink_zip_enabled=false
```

모든 데이터를 압축하지는 않으며 압축전송여부를 확인할때 debug 옵션을 활용합니다.

## 로그 압축 전송 최소 사이즈

전송 데이터가 100bytes 이하인 경우 데이터를 압축하지 않습니다.

```
whatap.conf
```

```
logsink_zip_min_size=100
```

## 로그 압축 전송 묶음 크기 및 시간

압축을 위해서는 데이터를 묶어서 전송해야 하는데 무한정 다음 데이터를 기다릴 수 없기 때문에 최대시간을 지정합니다.

```
whatap.conf
```

```
max_buffer_size=65536  
max_wait_time=2000
```

에이전트는 데이터 수집 사이즈가 `max_buffer_size` 를 초과하거나 로그를 묶기 위해 기다리는 시간이 `max_wait_time` 을 초과하면 로그 데이터를 서버로 전송합니다.

## 로그 압축 알고리즘 선택

로그를 압축전송할 때 압축 알고리즘 변경이 가능합니다. 기본 알고리즘은 GZIP이며 lz4도 선택 가능합니다.

```
whatap.conf
```

```
logsink_zip_libpath=/whatap/whatap-logsink-lz4-1.7.0.jar
```

! 알고리즘 선택은 `logsink_zip_libpath` 에 jar파일을 지정하는것으로 처리

## IBM Java6 웹스피어 환경에서 에이전트가 동작하지 않는 문제 수정

IBM Java6 웹스피어 환경에서 WhaTap Agent가 초기화 되지 못하고 데이터를 수집하지 못하는 문제를 수정 하였습니다.

! `ManagementFactory.getPlatformMBeanServer()` 로직이 내부적으로 오류가 발생하면서 WhaTap Agent가 초기화되지 못하는 문제

## CounterPack 디버깅 옵션 추가

에이전트에서 전송하는 실시간 성능 데이터의 전송 여부를 디버깅 하기 위한 옵션을 추가 하였습니다.

```
whatap.conf
```

```
debug_counterpack_enabled=true (기본값)  
debug_counterpack_factor=3 (기본값)  
debug_counterpack_recvtime=9000 (기본값)
```

에이전트의 큐에 전송 대기 상태의 CounterPack이 3 이상 늘어나거나 가장 최근에 받은 서버 데이터의 전송속도가 9000ms를 초과하는 경우 로그로 표시 합니다.

## Text & ActiveStack 정보 수집시 압축 옵션 추가

Text 데이터와 ActiveStack 정보를 수집할 때 압축할 수 있는 옵션을 추가 하였습니다.

```
whatap.conf
```

```
active_stack_zip_enabled=false (기본값)  
text_zip_enabled=false (기본값)
```

## 시스템 메트릭 수집 라이브러리 변경 및 메모리 데이터 수집 오류 수정

### OSHI 추가

OSHI(Operating System and Hardware Information)는 시스템 인프라 메트릭을 수집하기 위한 라이브러리로, 최근에도 활발히 업데이트 되고 있어 시스템 성능 지표를 수집하기 위한 라이브러리로 선택 하였습니다. 단, Java8 이상의 환경에서만 사용 가능합니다.

- `oshi_enabled=false` (기본값)
- jdk8 이상에서만 사용가능
- `disk="."`에 대한 사용량 측정 및 경고 기능 가능

### Sigar 기능 deprecated 처리

이미 5년 이상 업데이트 되고 있지 않아 초기 설정에서 제외 하였습니다.

- sigar\_enabled=false (기본값)
- lib1에 대해 aix만 포함

Sigar 기능을 사용하기 위해서는 명시적으로 `sigar_enabled=true` 설정 후 사용할 수 있습니다.

## Linux memory 수집 오류 수정

리눅스 환경에서 메모리 사용량 및 사용률 수집 오류를 수정 하였습니다.

## 환경변수 조회기능 옵션 처리

에이전트에서 환경변수 조회시 기능 동작 여부를 ON/OFF 할 수 있게 옵션을 추가 하였습니다.

```
whatap.conf
```

```
agent_env_system_enabled=true  
agent_env_jvm_enabled=true
```

프로세스 환경변수와 자바 환경변수에 대해서 조회가능 여부를 옵션으로 통제할 수 있습니다.

## 에이전트 환경변수 조회시 특정 키를 제외하는 옵션

자바 실행환경의 환경변수를 조회하는 기능에서 특정 환경변수를 제외할 수 있는 옵션을 추가 하였습니다.

```
whatap.conf
```

```
agent_env_ignore_keys=USER,Address
```

';' 구분자를 사용하여 옵션지정하며, 대소문자를 구분하지 않습니다.

## TOMCAT10 지원

Tomcat10의 jakarta 라이브러리를 추적할 수 있도록 지원 범위를 확대 하였습니다. weaving 옵션에 tomcat10을 지정하여 추적할 수 있습니다.

```
whatap.conf
```



```
weaving=tomcat10
```

! weaving 옵션은 `weaving_reserved` 옵션을 축약하여 v2.1.1에서 추가

## HttpCall에 대한 상세 정보 추적

트랜잭션 추적에서 HttpCall에 대한 추적 정보를 추가하였습니다.

! OLD 버전의 수집서버에서는 트랜잭션 상세 스텝 정보 확인이 불가능합니다.

만약 OLD 수집서버에 데이터를 보내야 한다면 `trace_http_version` 옵션을 `whatap.conf` 파일에 추가해야 합니다.

```
whatap.conf
```

```
trace_http_version=1
```