# Table of Contents

**WhaTap .NET Agent Guide**  

1. Install ................................................. 2  
   1.1. 에이전트 실행 및 모니터링 개요 .................. 2  
   1.2. 프로젝트 생성 및 라이센스 발급 .................. 3  
   1.3. 에이전트 설치 .................................. 8  
   1.4. 설치 오류 해결 ................................ 10  

2. Data .................................................. 13  
   2.1. 트랜잭션 성능 추적 ....................... 13  
   2.2. 동계(리스트 데이터) ....................... 15  
   2.3. 성능 카운터 ................................ 19  

3. 에이전트 제어와 상태 조회 .............................. 22  
   3.1. 부트 환경 ................................ 22  
   3.2. 환경 변수 ................................ 22  
   3.3. 에이전트 상태 ................................ 22  
   3.4. 설정 ........................................ 22  
   3.5. 에이전트 로그 ................................ 22  

4. 설정 .................................................. 23  
   4.1. 에이전트 네이밍 ................................ 23  
   4.2. 에이전트 기능제어 ....................... 23  
   4.3. 에이전트 통신 ................................ 24  
   4.4. 에이전트 성능 ................................ 25  
   4.5. 에이전트 로그 관리 ....................... 27  
   4.6. 에이전트 알림 ................................ 27  
   4.7. 트랜잭션 프로파일링 ....................... 28  
   4.8. HTTP 트랜잭션 추적 ....................... 29  
   4.9. DB, SQL ........................................ 30  
   4.10. HTTPC, API Call .............................. 31  
   4.11. 사용자 수 ................................ 31  
   4.12. 트랜잭션 추적 옵션 ....................... 32  
   4.13. 통계 ........................................ 32
WhaTap .NET Agent Guide

제목: .NET Agent Guide
작성자: WhaTap Support
이메일: support@whatap.io
날짜: 2020-01-30
버전: 1.0.0

설명: 본 문서는 WhaTap .NET Monitoring 에이전트에 대해 설명합니다.
Chapter 1. Install

1.1. 에이전트 실행 및 모니터링 개요

와탭 애플리케이션 모니터링은 닷넷 기반 웹 애플리케이션 서버 모니터링 서비스를 제공합니다.

1.1.1. 에이전트 동작 방식 개요

닷넷 프로그램의 경우 컴파일 후에 MSIL (중간언어) 형태인 exe나 dll로 변환됩니다. 이것을 CLR(공통 언어 런타임)에 의하여 네이티브 코드로 번역되어 동작합니다. 와탭은 닷넷 프레임워크에서 제공하는 프로파일링 API를 통하여 실행중인 애플리케이션을 모니터링 합니다.

다트넷 모니터링 서비스를 사용하기 위해서는 모니터링 대상 애플리케이션에 모니터링 에이전트를 설치해야 합니다. 에이전트는 윈도우에 설치 가능한 인스톨러로 제공됩니다. 설치되면 윈도우 서비스에 와탭 에이전트가 자동등록 됩니다.

1.1.2. 지원환경

운영체제 : Windows Server 2012 이상 (IIS 8.0 이상)
다트넷버전 : .NET Framework 4.5

Copyright © 2019 WhaTap Labs Inc. All rights reserved.
지원 라이브러리:
* WebClient / WebRequest
* Http/Client / HttpClientHandler
* ASP.NET (MVC5, Web API2, Web Forms)
* ADO.NET (Database)

### 1.1.3. 구성 파일
모니터링 정보를 수집하기 위한 트레이서, 수집된 정보를 서버에 전송하기 위한 에이전트로 구성됩니다.

<table>
<thead>
<tr>
<th>파일명</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>WhaTap.ClrProfiler.dll</td>
<td>.NET CLR Profiler입니다. .NET 프로파일링 API와 연결됩니다.</td>
</tr>
<tr>
<td>WhaTap.Trace.dll</td>
<td>프로파일러에서 전달되는 데이터를 처리하고 에이전트로 전송합니다.</td>
</tr>
<tr>
<td>whatap_dotnet.exe</td>
<td>에이전트, 트레이서에서 전달된 정보를 수집서버로 전송하는 프로그램입니다.</td>
</tr>
<tr>
<td>whatap.conf</td>
<td>에이전트 환경 설정 파일입니다.</td>
</tr>
</tbody>
</table>

### 1.1.4. 에이전트 이름 식별
와탭은 모니터링 정보 수집 대상인 애플리케이션 서버 식별을 위한 정보로 기본적으로 애플리케이션 서버로부터 수집한 정보를 활용합니다. 기본적으로 활용하는 정보는 애플리케이션 서버 종류, 애플리케이션 서버의 IP, 서비스 포트를 포함하여 애플리케이션 서버를 고유 식별자로 사용하게 되며 필요에 따라 사용자가 지정한 명칭을 사용하거나 패턴을 변경하여 사용하는 것도 가능합니다. 이때에는 꼭 고유한 값이어야 합니다.

애플리케이션 서버로부터 추출한 정보를 활용하는 이유는 애플리케이션 서버 정보와 네트워크 단절 또는 에이전트 문제로 인한 수집 서버와 에이전트의 통신 단절 상태가 복구되었을 경우, 재 접속된 에이전트로부터 송신되는 정보가 기존 에이전트로부터 송신된 정보와의 연속성을 유지하기 위해서 입니다. 와탭이 애플리케이션 서버를 식별하기 위해 사용하는 기본 패턴은 다음과 같습니다.

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

기본 패턴에 대한 변경은 whatap.ini에서 설정에서 가능합니다.

<table>
<thead>
<tr>
<th>설정</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>app_name</td>
</tr>
<tr>
<td>Ip#</td>
<td>Ip을 오로 나누었을 때 #번째 자리(0부터)</td>
</tr>
<tr>
<td>Process</td>
<td>app_process_name</td>
</tr>
<tr>
<td>hostname</td>
<td>호스트 명</td>
</tr>
</tbody>
</table>

### 1.2. 프로젝트 생성 및 라이센스 발급

#### 1.2.1. 프로젝트 생성
“프로젝트 생성” 버튼을 누릅니다.
플랫폼 중에 ".NET"을 선택합니다.
• 프로젝트 이름: 용도에 맞게 인식할 수 있는 이름으로 지정합니다.
• 데이터 서버 지역: 모니터링 데이터가 저장되는 서버가 위치한 지역입니다.
• 타임존: 모니터링 데이터가 화면에 표시될 때 적용되는 시간대를 지정합니다. (사용자가 위치한 지역의 타임존)
• 프로젝트 그룹: 사전에 프로젝트 그룹을 생성해 놓은 경우 용도에 따라 그룹에 속하도록 지정할 수 있습니다.

설정 후에 저장을 눌러서 프로젝트를 생성 합니다.

1.2.2. 라이센스 발급
새 프로젝트가 생성되면 기본 화면으로 에이전트 추가 화면이 보여집니다. 라이선스를 생성합니다. 라이선스는 에이전트 프로그램에 환경설정으로 입력하게 됩니다. 에이전트는 설치된 라이선스 키를 사용하여 해당 프로젝트에 자동 등록 됩니다. 라이선스 키는 프로젝트별로 귀속되기 때문에, 유출되거나 배포되어서는 안됩니다. 반드시 본인 프로젝트에 서버를 등록할 때에만 이용하시기 바랍니다.

에이전트를 설치할 서버에 인스톨러 파일을 다운로드 합니다. 환경에 맞게 exe 또는 zip 을 다운로드 합니다.
향후 인스톨러 설치시 라이선스와 와탭 서버의 IP 주소를 입력하게 됩니다.
화면을 유지한 채로 인스톨러를 설치합니다.

1.3. 에이전트 설치

1.3.1. 윈도우

인스톨러 설치

향후 인스톨러 설치시 라이선스와 와탭 서버의 IP주소를 입력하게 됩니다. 화면을 유지한 채로 인스톨러를 설치합니다.
다운로드 받은 인스톨러를 실행하고 라이선스와 와탭 서버의 IP 주소를 입력합니다.

설치를 진행합니다.
파일 복사중입니다.

설치가 완료되면 확인을 눌러 설치를 종료합니다.

1.4. 설치 오류 해결
1.4.1. 동작 상태 확인

서비스 동작 확인

services.msc 명령을 입력하여 윈도우 서비스 실행 상태를 확인합니다.

![서비스 동작 상황]

WhaTap .NET 이 정상 등록되어 있고, 동작 상태 인지 확인합니다. WhaTap .NET 서비스는 수집한 모니터링 데이터를 와탭 서버로 전송하는 기능을 담당합니다.

관련된 로그는 C:\Program Files\WhaTap .NET\logs 폴더에 위치합니다.

시스템 환경 변수 확인

sysdm.cpl 명령을 입력하여 시스템 환경 창을 열고 "고급" → "환경변수"를 선택하여 환경 변수 창을 엽니다.
시스템 환경변수는 인스톨러에 의해 자동 등록됩니다. 아래와 같은 내용이 등록되어 있는지 확인합니다.

• COR_ENABLE_PROFILING = 1  → CLR 프로파일러를 활성화 합니다.
• COR_PROFILER = {D76F1D76-A9E0-4C87-874F-C0AD93D4229B}  → 와탭 CLR 프로파일러를 API 에 등록할때 사용하는 GUID 입니다. ( 고정 )
• COR_PROFILER_PATH = C:\Program Files\WhaTap .NET\WhaTap.ClrProfiler.dll  → 와탭 CLR 프로파일러 DLL의 위치입니다.
• WHATAP_APP_TYPE = 7  → 와탭 서버와 통신하는 에이전트 종류로 고정값 입니다.
• WHATAP_DOTNET_HOME = C:\Program Files\WhaTap .NET  → 와탭 프로그램의 홈 디렉토리 입니다.
• WHATAP_DOTNET_TRACE = 1  → 모니터링 데이터 수집 기능을 활성화 합니다.
Chapter 2. Data

와탭 에이전트는 트랜잭션 성능, 주요 성능 통계 그리고 서비스와 자원에 대한 카운터를 수집합니다.

2.1. 트랜잭션 성능 추적

2.1.1. 트랜잭션 시작과 종료

트랜잭션은 사용자 브라우저의 요청을 처리하기 위한 서버 사이드의 LUW(Logical Unit of Work)를 말합니다. 개별 웹서비스(URL) 요청에 대한 처리과정이 바로 트랜잭션인 것입니다. 웹 어플리케이션에서 트랜잭션은 웹서비스(URL)에 대한 HTTP Request를 받아 Response를 반환하는 과정입니다.

애플리케이션의 성능은 이 트랜잭션들의 성능으로 요약할 수 있습니다.

트랜잭션 성능은 트랜잭션 시작에서부터 종료시점, 그리고 응답시간 및 자원 사용량 혹은 트랜잭션 호출자의 속성 등의 정보들을 포함합니다.

기본적으로 트랜잭션 응답 분포와 트랜잭션 통계를 통해서 트랜잭션 성능을 분석할 수 있습니다.

트랜잭션의 이름

트랜잭션의 이름은 URL입니다. 단 Get 파라미터(Query String)는 제외 됩니다.

브라우저 요청

http://www.whatap.io/hr/apply.do?name='kim'

트랜잭션 이름

/hr/apply.do

와탭에서는 "웹서비스 이름 + "과 "트랜잭션 이름"을 혼용해서 사용합니다. 서비스 특정 URL과 그에 대한 요청을 처리하기 위한 모듈로 볼수있고 트랜잭션 그 요청에 대한 처리 하나를 의미하기 때문에 둘의 이름은 동일하게 URL이라고 할 수 있습니다.

트랜잭션 이름 정규화

MSA 기반의 시스템이 발전하면서 URL?파라미터 형식보다는 URL 파스에 파라미터를 넣는 방식을 많이 사용하게 됩니다.

http://www.whatap.io/hr/kim/apply.do

이렇게 파라미터를 그대로 트랜잭션 이름으로 사용하게 되면 통계적 관점의 성능 분석이 어렵게 됩니다. 따라서 정규화할 필요가 있습니다. 와탭은 이때 정규화를 위한 옵션과 기능을 제공하고 있습니다.

whatap.ini

trace_normalize_urls=/hr/{name}/apply.do

위와 같이 설정하면 트랜잭션 이름이 /hello/kim → /hello/name 이렇게 치환되어 수집됩니다. 만약 대상 url설정은 그대로 두고 기능만 off하고자 한다면 다음과 같이 옵션을 지정할 수 있습니다. 기본값은 true입니다.

whatap.ini

trace_normalize_enabled=false

Copyright © 2019 WhaTap Labs Inc. All rights reserved.
2.1.2. 트랜잭션 프로파일

트랜잭션 성능이 트랜잭션 시작과 종료사이의 요약 지표들이나 속성을 의미한다면 트랜잭션 프로파일은 트랜잭션이 수행되는 과정중이 스텝들을 추적하는 것입니다.

와입이 수집하는 스텝의 종류에는 크게 SQL 스텝, HTTP CALL 스텝, 메세지 스텝, SOCKET 오픈 스텝, DB 연결 스텝, 메소드 스텝들이 있습니다.

**DB연결 스텝**
수집방식: END
RDB에 대한 연결에 대한 성능을 포함합니다. 스텝 정보에는 이름, 응답시간을 포함합니다.

**SQL 스텝**
수집방식: END
SQL에 대한 성능을 포함합니다. 스텝정보에는 연결정보,SQL문이 포함되어있습니다.

**HTTP Call 스텝**
수집방식: END
외부 http서비스 호출에 대한 성능을 포함합니다. 스텝 정보에는 url, host, port, 응답시간이 포함됩니다.

**Message 스텝**
수집방식: ADD
메세지를 수집하는 스텝을 포함하는 과정에서 비정형적인 모든 구간에 대한 이력을 수집할때 사용됩니다. file 오픈 등등 혹은 사용자도 입의의 위치를 저장하는데 사용 할 수 있습니다.

**SOCKET 스텝**
수집방식: ADD
Socket 오픈을 표현하는 스텝입니다.

**METHOD 스텝**
수집방식: END
메소드 응답시간을 추적합니다. 1초 이상의 메소드를 자동으로 추적합니다.

2.1.3. 트랜잭션 스텝 수집 방식

에이전트는 선택 수집 방식과 환경 수집방식을 제공합니다.

**선형 수집**
개별 트랜잭션 프로파일은 무한정 스텝을 수집할 수 없습니다. 트랜잭션 별로 길이가 제한된 버퍼에 프로파일 스텝을 저장합니다. 각 트랜잭션은 최대로 수집할 수 있는 스텝의 수가 옵션으로 저장되어 있습니다.

```
profile_step_max_count
default: 1024
최대로 수집가능한 스텝수
```

```
profile_step_normal_count
default: 800
일반적으로 아무런 제약없이 수집되는 스텝수
```

Copyright ⓒ 2019 WhaTap Labs Inc. All rights reserved.
profile_step_heavy_count
default: 1000
normal count를 초과한 경우에 스텝은 응답시간이 느린 스텝과 액티브 스텝만 수집합니다.

profile_step_heavy_time
default: 100
heavy count이내에서의 수집되는 스텝의 기준 시간은 profile_step_heavy_time입니다.

profile_step_max_count
profile_step_normal_count
profile_step_heavy_count
profile_step_heavy_time

Figure 1. Profile Step Buffer

수집되는 프로파일의 스텝 수가 heavy count를 초과하는 경우에는 액티브 스텝만이 수집됩니다. 단 이경우에도 최대 스텝 수는 profile_step_max_count를 넘을 수 없습니다.

2.2. 통계(리스트 데이터)

와탐 에이전트는 트랜잭션이나 SQL처럼 중요한 서비스 수행이력 통계를 수집합니다. 매 5분마다 목록을 만들고 서버로 전송합니다.

0분, 5분, 15분 등 매 5분마다 통계를 수집하고 서버로 전송합니다.

2.2.1. 트랜잭션 통계

트랜잭션 통계를 수집합니다. 매 5분마다 최대 5000개의 URL별 수행 통계를 수집하여 서버에 전송합니다. 만약 서로다른 URL의 수가 5분동안 5000개가 넘으면 무시됩니다.

<table>
<thead>
<tr>
<th>폴러</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>hash</td>
<td>URL 해쉬</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>건수</td>
<td>u4</td>
</tr>
<tr>
<td>error</td>
<td>에러건수</td>
<td>u4</td>
</tr>
<tr>
<td>time_sum</td>
<td>응답시간의 합</td>
<td>u8</td>
</tr>
<tr>
<td>time_max</td>
<td>최대 응답시간</td>
<td>u4</td>
</tr>
<tr>
<td>sql_count</td>
<td>SQL 수행 건수</td>
<td>u4</td>
</tr>
<tr>
<td>sql_time</td>
<td>SQL 수행 시간의 합</td>
<td>u8</td>
</tr>
<tr>
<td>httpc_count</td>
<td>HTTP Call 건수</td>
<td>u4</td>
</tr>
<tr>
<td>httpc_time</td>
<td>HTTP Call 시간의 합</td>
<td>u8</td>
</tr>
</tbody>
</table>
### 2.2.2. SQL 수행 통계

5분 동안의 SQL 수행 통계를 수집합니다. 5분 동안 서로 다른 SQL문장이 최대 5000개까지 허용이 됩니다.

<table>
<thead>
<tr>
<th>항목</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>malloc_sum</td>
<td>메모리 사용량 SUM</td>
<td>u8</td>
</tr>
<tr>
<td>cpu_sum</td>
<td>CPU사용량 SUM</td>
<td>u8</td>
</tr>
</tbody>
</table>

서비스(URL) hash는 5분 동안 해당 SQL문장 호출한 URL중 하나(마지막 호출 URL)를 분석 활용을 위해 수집합니다.

### 2.2.3. HTTPCall 수행 통계

5분 동안의 Http Call 수행 통계를 수집합니다. 5분 동안 서로 다른 Http Call문장이 최대 5000개까지 허용이 됩니다.

<table>
<thead>
<tr>
<th>항목</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>타겟 URL hash</td>
<td>u4</td>
</tr>
<tr>
<td>host</td>
<td>Host or ip</td>
<td>u4</td>
</tr>
<tr>
<td>port</td>
<td>Tcp Port</td>
<td>u4</td>
</tr>
<tr>
<td>count_total</td>
<td>수행 건수</td>
<td>u4</td>
</tr>
<tr>
<td>time_sum</td>
<td>응답시간의 합</td>
<td>u8</td>
</tr>
<tr>
<td>time_max</td>
<td>최대 응답시간</td>
<td>u4</td>
</tr>
<tr>
<td>service</td>
<td>Http Call을 수행한 service중에 하나</td>
<td>u4</td>
</tr>
</tbody>
</table>

### 2.2.4. 에러 통계

5분 동안 발생한 서비스 에러에 대한 통계입니다. 서로 다른 에러+트랜잭션 이름을 키로 발생 건수를 수집합니다. 5분 동안 최대 1000가지 서로 다른 에러를 통계화합니다.

수집하는 에러 타입은 아래와 같습니다.

- E_CORE_ERROR, E_PARSE, E_COMPILE_ERROR, E_ERROR, E_USER_ERROR, E_RECOVERABLE_ERROR
- Exception

<table>
<thead>
<tr>
<th>항목</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>classHash</td>
<td>에러 클래스</td>
<td>u4</td>
</tr>
</tbody>
</table>

Copyright © 2019 WhaTap Labs Inc. All rights reserved.
2.2. 통계(리스트 데이터)

<table>
<thead>
<tr>
<th>칼럼</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>service</td>
<td>트랜잭션 HASH</td>
<td>u4</td>
</tr>
<tr>
<td>msg</td>
<td>메세지 hash</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>발생 건수</td>
<td>u4</td>
</tr>
</tbody>
</table>

2.2.5. IP별 호출 건수

IP별로 호출한 트랜잭션 건수를 통계적으로 수집합니다. 5분당 수집가능한 서로 다른 IP수는 인스턴스당 최대 70000개입니다.

<table>
<thead>
<tr>
<th>칼럼</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>주소</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>건수</td>
<td>u4</td>
</tr>
</tbody>
</table>

2.2.6. UserAgent별 호출 건수

User Agent 문자열의 Hash별로 호출 건수를 수집합니다. 5분당 수집가능한 서로 다른 UserAgent Hash는 인스턴스당 최대 500개입니다.

<table>
<thead>
<tr>
<th>칼럼</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>hash</td>
<td>hash</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>건수</td>
<td>u4</td>
</tr>
</tbody>
</table>

2.2.7. 트랜잭션 Caller 통계

멀티 서버가 rest 호출로 연결된 경우 Caller와 Callee간의 연관 통계를 수집할수있다. 이 데이터를 수집하기 위해서는 다음의 옵션을 먼저 설정해야 한다.

```ini
whatap.ini

//mtrace_rate=100
mtrace_spec=v1
stat_mtrace_enabled=true
```

Figure 2. MSA 시스템

위와 같은 아키텍처에서 caller&callee되는 api1와 api2에서만 조회할 수있습니다. 사용자 브라우저에서 호출되는 시스템에서는 Caller통계를 조회할 수 없습니다.

하지만 Caller쪽 서버에서 데이터를 전송해야하기 때문에 모든 서버에 적절한설정이 들어가야 합니다.

```ini
whatap.conf of [front]

mtrace_rate=100
mtrace_spec=v1
stat_mtrace_enabled=true
```
2.2. 통계(리스트 데이터)

whatap.conf of [api1] & [api2]

```yaml
mtrace_spec=v1
stat_mtrace_enabled=true
```

수집되는 통계 데이터는 다음과 같습니다. Callee쪽에서 조회되어야합니다.

<table>
<thead>
<tr>
<th>열</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>caller_pcode</td>
<td>Caller의 프로젝트(와탭) 코드</td>
<td>u8</td>
</tr>
<tr>
<td>caller_spec</td>
<td>Caller의 버전 문자열 hash</td>
<td>u4</td>
</tr>
<tr>
<td>caller_url</td>
<td>Caller의 URL hash</td>
<td>u4</td>
</tr>
<tr>
<td>spec</td>
<td>Callee의 버전 문자열 hash</td>
<td>u4</td>
</tr>
<tr>
<td>url</td>
<td>Callee URL hash</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>수행 건수</td>
<td>u4</td>
</tr>
<tr>
<td>error</td>
<td>에러건수</td>
<td>u4</td>
</tr>
<tr>
<td>time</td>
<td>응답시간의 합</td>
<td>u8</td>
</tr>
</tbody>
</table>

2.2.8. 트랜잭션 도메인 통계

와탭에이전트는 도메인별 트랜잭션 통계를 수집할 수 있습니다. 하나의 서버에 비즈니스적인 이유등으로 여러개의 도메인을 분리하여 서비스 하는 시스템에서는 도메인 별 분석이 필요할 수 있습니다. 그래서 front서버에서 유호하다.

whatap.conf

```yaml
stat_domain_enabled=true
```

수집되는 데이터는 도메인별 URL의 처리 현황을 파악할 수 있습니다.

<table>
<thead>
<tr>
<th>열</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>domain</td>
<td>서비스 도메인 hash</td>
<td>u4</td>
</tr>
<tr>
<td>url</td>
<td>트랜잭션 URL hash</td>
<td>u4</td>
</tr>
<tr>
<td>count</td>
<td>수행 건수</td>
<td>u4</td>
</tr>
<tr>
<td>error</td>
<td>에러건수</td>
<td>u4</td>
</tr>
<tr>
<td>time</td>
<td>응답시간의 합</td>
<td>u8</td>
</tr>
</tbody>
</table>

2.2.9. 트랜잭션 REFERER 통계

와탭에이전트는 Referer별 트랜잭션 통계를 수집할 수 있습니다. 업무에 맞서는 Referer별 분석이 필요할 수 있습니다.

whatap.conf

```yaml
stat_domain_enabled=true
```

수집되는 데이터는 도메인별 URL의 처리 현황을 파악할 수 있습니다.

<table>
<thead>
<tr>
<th>열</th>
<th>설명</th>
<th>타입</th>
</tr>
</thead>
<tbody>
<tr>
<td>referer</td>
<td>Referer hash</td>
<td>u4</td>
</tr>
</tbody>
</table>
2.3. 성능 카운터

와탭 에이전트가 수집하는 성능카운터는 크게 3가지로 분류할 수 있습니다.

Counters
- User: 실시간 사용자 혹은 방문사용자
- Service: 트랜잭션, SQL, 외부호출 건수 및 응답, 에러율 등
- Resource: 시스템, 프로세스 자원 사용량

2.3.1. User Counter

사용자는 모니터링 대상 시스템을 사용하는 클라이언트를 말한다. 클라이언트에는 다른시스테드 시스템도 포함될 수 있지만 일반적으로는 브라우저로 기존으로 사용자를 카운팅한다.

웹시스템 성능에서 사용자는 부하를 발생시키는 시작기 때문에 중요하다 사용자 추적을 위해서는 사용자는 어떤 기준으로 구분할 것이며 어떻게 카운팅 할 것인가에 대한 고려가 필요하다

사용자 구분

와탭 에이전트 사용자를 구분하기 위해 다양한 옵션을 제공합니다. 기본 값은 IP를 사용합니다.

RemoteIP

가장 기본은 remote ip를 사용하여 사용자를 구분하는 것입니다. 하지만 remote ip 실제 사용자를 구분하는데 한계가 있습니다.

Cookie(WHATAP)

쿠키를 사용하여 사용자를 구분할 수 있습니다.

whatap.conf

```bash
whatap.trace_user_using_ip=false
whatap.trace_user_set_cookie=true
```

- 모든접속 클라이언트에 대한 UUID가 "WHATAP"이라는 쿠키에 셋팅합니다.

Header Key

http header에 전달되는 값으로 사용자를 구분할 수 있습니다. .whatap.conf

```bash
whatap.trace_user_header_ticket=USER
```

Copyright ⓒ 2019 WhaTap Labs Inc. All rights reserved.
사용자 카운팅

사용자를 카운팅하는 방법에 따라서 다른 목적으로 사용될 수 있다. 동시에 사용자는 현재 시스템을 사용하는 사용자의 수를 알기 위해서 측정하고 일일 액티브 사용자는 하루동안 해당 서비스에 관심을 갖는 사용자가 몇명인지에 대한 비즈니스적인 관리를 위해 측정한다.

실시간 사용자

최근 5분 동안 사용자 수를 카운팅합니다. 매5초마다 shifting하면 사용자를 카운팅합니다. 각 서버에서 카운팅된 숫자는 HyperLogLog 알고리즘을 통해서 미지워집니다.

일일 방문(액티브) 사용자(DAU)

하루동안 시스템에 접속한 사용자를 카운팅합니다. 24시간 동안 발생한 사용자를 hyperloglog로 통해서 계산합니다.

와탭에서는 장기간 사용자를 카운팅하기 위해서 사용자데이터에 대한 byte block을 서버로 수집합니다. 이 데이터를 hyperloglog로 미지하면 이론적으로 한달 이상의 액티브 사용자를 계산 할 수 있습니다.

2.3.2. Service Counter

서비스 카운터에는 트랜잭션과 트랜잭션이 사용하는 SQL 혹은 외부호출등에 대한 건수 응답시간 에러건수 등에 대한 성능지표가 포함된다.

Transaction Counter

트랜잭션을 수행하면 측정하는 카운터입니다.

- 건수
- 응답시간
- 에러건수

Active Transaction Counter

진행중인 트랜잭션의 수를 카운팅합니다.

- 건수
- Active Status

진행 상태는 METHOD, SQL, HTTPC, DBC, SOCKET 5가지 상태로 고정되어있습니다.

1. METHOD – 일반 함수를 호출하는 상태
2. SQL – db sql을 수행중인 상태
3. HTTPC – 외부 Http Api(서비스)를 호출중인 상태
4. DBC – DB연결을 요청한 상태, 일반적으로 Pool에서 가져옴
5. SOCKET – TCP세션을 Connecting 중인 상태

SQL

SQL 수행 현황을 카운팅합니다.

- 건수
- 응답시간
- 에러건수
- 패치건수

HTTP Call

외부 Http 호출에 대한 현황을 카운팅 합니다.
2.3. 성능 카운터

- 건수
- 응답시간
- 에러 건수

2.3.3. Resource Counter

서버자원 사용량을 카운팅합니다.

CPU (sys, usr, wait, steal, irq, cores)

CPU 사용량 % 입니다. 각 종류별로 수집됩니다. 가상환경에서만 Steal이 의미가 있습니다. Cput Core 갯수를 같이 수집하고 있습니다.

Process CPU

자바 프로세스가 사용하는 CPU %입니다.

Memory

시스템 메모리 사용율(%) 입니다.

Swap

Swap 메모리 사용율(%) 입니다.

Disk

Disk는 Process의 Current 디렉토리의 사용율(%) 입니다.

Heap (Total, Used)

시스템 전체 메모리를 Total로 표시하며, IIS 프로세스의 메모리 사용량을 합산하여 Used로 표시합니다. 데이터 단위는 KBytes 입니다.
Chapter 3. 에이전트 제어와 상태 조회

각 에이전트 동작을 제어하거나 상태를 조회할 수 있습니다.

3.1. 부트 환경

에이전트는 시작하면 주요환경 정보를 모아 서버로 전송합니다. 이정보는 기동시의 에이전트 환경 상태를 확인할때 사용될 수 있습니다. Yard서버에 저장해 두기 때문에 에이전트가 종료되어도 조회할 수 있습니다.

와탑의 버전, Hostname, uptime 등의 정보들을 조회할 수 있습니다.

3.2. 환경 변수

에이전트에서는 현재의 환경 변수를 조회할 수 있습니다. 화면의 요청을 받았을때 조회됩니다. 따라서 프로세스가 정지된 상태에서는 조회할 수 없습니다.

3.3. 에이전트 상태

서비스로 동작하는 WhaTap .NET 프로세스에 대해서 최근 하루의 상태를 확인합니다. CPU, Memory, Network 상태를 확인할 수 있습니다.

3.4. 설정

에이전트 실행옵션을 설정할 수 있습니다. Configure 헬퍼에서 소개하고 있는 대부분의 옵션은 whatap.conf에 적용할 수 있습니다.

 일반적으로 설정은 런타임에 바로 reload됩니다. 다만 해당 설정의 메커니즘상 IIS가 시작할때 적용되는 옵션들은 재가동해야 할수 있습니다.

3.5. 에이전트 로그

에이전트 로그는 C:\ProgramData\WhaTap 에 기록됩니다.

dotnet-profiler.log ( 프로파일러 동작 로그 )  dotnet-tracer-{프로세서명}-{날짜}.log ( 프로세서 로그 )
Chapter 4. 설정

4.1. 에이전트 네이밍

whatap.object_name
Default : {type}-{ip2}-{ip3}-{process}
Type : String
애플리케이션을 식별하는 에이전트 이름(ONAME)체계입니다. ONAME을 토대로 OID가 생성됩니다.

<table>
<thead>
<tr>
<th>명칭</th>
<th>설명</th>
</tr>
</thead>
<tbody>
<tr>
<td>{type}</td>
<td>whatap.app_name 에 설정된 값을 사용합니다.</td>
</tr>
<tr>
<td>{ip#}</td>
<td>IP를 나누었을 때 #번째 자리의 값을 사용합니다.</td>
</tr>
<tr>
<td>{process}</td>
<td>whatap.app_process_name 에 설정된 값을 사용합니다.</td>
</tr>
<tr>
<td>{hostname}</td>
<td>서버 호스트명을 사용합니다.</td>
</tr>
</tbody>
</table>

IIS 재시작 후 적용됩니다.

whatap.app_name
Default : NONE
Type : String
애플리케이션을 식별하는 에이전트 이름(ONAME)체계에 사용되는 애플리케이션 명. object_name의 {type}에 해당하는 값이다.

IIS 재시작 후 적용됩니다.

whatap.app_process_name
Default : NONE
Type : String
애플리케이션을 식별하는 에이전트 이름(ONAME)체계에 사용되는 애플리케이션 프로세스 명. 애플리케이션 서버의 CPU, Heap Memory등을 수집할 대상 프로세스를 설정합니다. object_name의 {process}에 해당하는 값이다.

IIS 재시작 후 적용됩니다.

4.2. 에이전트 기능제어

whatap.stat_enabled
Default : true
Type : Boolean
통계정보 추적 기능을 활성화합니다. 5 분단위로 수집되는 트랜잭션, SQL, HTTPCALL, UserAgent, Client IP 등의 통계 데이터들이 해당됩니다.

whatap.license
Default : NONE
Type : String
에이전트 설치시 서버로부터 부여받은 라이센스를 지정합니다. 라이센스에는 에이전트가 속한 프로젝트와 보안 통신을 위한 암호키를 포함하고 있습니다.

whatap.realtime_user_thinktime_max
Default : 300000
4.3. 에이전트 통신

**whatap.time_sync_interval_ms**
Default : 300000
Type : Milliseconds
에이전트와 서버간 시간 동기화 주기를 지정합니다. 동기화 하지 않을 경우 0으로 지정합니다.

**whatap.text_reset**
Default : 0
Type : Int
와탭 에이전트는 한번 보낸 텍스트유형 데이터는 hash 처리되므로 다음날까지 재전송하지 않습니다. 이전 설정값과 다른 값을 설정하는 경우 재전송 합니다.

```
| 트랜잭션 URL, SQL String 등이 텍스트유형 데이터에 해당합니다. |
```

**whatap.profile_method_enabled**
Default : true
Type : Boolean
실행시간이 1초(whatap.profile_method_time)를 초과하는 사용자 함수의 정보를 트랜잭션 상세 정보에서 확인할 수 있습니다.

```
| IIS 재시작 후 적용됩니다. |
```

**whatap.profile_method_time**
Default : 1000
Type : Milliseconds
whatap.profile_method_enabled 설정 수립되는 함수의 실행시간을 설정할 수 있습니다. NOTE: IIS 재시작 후 적용됩니다.

**whatap.profile_method_stack_enabled**
Default : false
Type : Boolean
사용자 함수의 스택정보를 트랜잭션 상세 정보에서 확인할 수 있습니다.

```
| IIS 재시작 후 적용됩니다. |
```

**whatap.server.host**
Default : 127.0.0.1,127.0.0.1
Type : ip_address
에이전트가 수집한 데이터를 전송할 서버를 지정합니다. 수집서버 이중화로 2개 이상의 IP를 가진 경우 락마(,)로 분리하여 지정할 수 있습니다. 지정된 IP에는 수집서버 proxy 대신이 리스닝 상태로 서비스 되어야 합니다.

**whatap.server.port**
Default : 6600
Type : tcp_port
수집서버 PORT 를 지정합니다. 포트는 하나만 지정할 수 있으므로 whatap_server_host 에 지정된 수집서버들은 동일 PORT 를 사용해야 합니다.

**whatap.tcp_so_timeout**
Default : 60000
Type: Millisecond
수집서버와 통신하는 TCP세션의 Socket Timeout 값을 지정합니다.

whatap.tcp_connection_timeout
Default: 5000
Type: Millisecond
수집서버와 통신하는 TCP세션의 Connection Timeout 값을 지정합니다.

whatap.net_send_max_bytes
Default: 5242880
Type: Byte
수집서버로 데이터를 전송할 때 한번에 전송되는 최대 크기를 지정합니다.

whatap.net_udp_port
Default: 6600
Type: tcp_port
와탭 에이전트는 트레이서에서 UDP를 통해 수집한 데이터를 수집서버로 전송합니다. 처음 UDP서버의 포트를 지정할 수 있다. 기본값으로 제공되는 6600포트가 사용 중일 때 이 옵션을 사용합니다.

IIS 재시작 후 적용됩니다.

4.4. 에이전트 성능

메이전트 성능 옵션은 설정 후에 whatap .NET 윈도우 서비스를 반드시 재시작해야 합니다.
메이전트 성능을 확대할 경우 Cpu, Memory 사용율이 증가할 수 있습니다.

whatap.tx_max_count
Default: 5000
Type: Int
트랜잭션을 최대 저장하는 개수 입니다. 동시에 처리되는 트랜잭션이 많을 경우 해당 크기를 더 늘려줍니다.

whatap.tx_default_capacity
Default: 101
Type: Int
트랜잭션을 저장하는 버퍼의 시작 크기 입니다.

whatap.tx_load_factor
Default: "0.75"
Type: String(float)
트랜잭션을 저장하는 버퍼의 크기를 평균적으로 증가시키기 위한 기준입니다. 101개에서 75%이상 저장되면 버퍼의 크기를 두배로 증가시킵니다.

whatap.queue_log_enabled
Default: false
Type: Boolean
queue 관련 로그를 출력합니다.

whatap.queue_tcp_enabled
Default: true
Type: Boolean
Tcp 데이터 송신 처리에 사용하는 Channel을 대신해서 별도의 Queue를 사용합니다.(데이터 우선 순위 Queue를 사용합니다.)
whatap.queue_tcp_sender_thread_count
Default : 2
Type : Int
Tcp 데이터 송신을 처리하는 쓰레드의 개수를 지정합니다. 개수를 늘리면 데이터 송신을 좀 더 빠르게 처리 할 수 있습니다.

whatap.net_send_buffer_size
Default : 1024
Type : Int
Tcp 데이터 처리를 위한 Channel 버퍼를 설정합니다.

whatap.net_send_queue1_size
Default : 256
Type : Int
Tcp 데이터 처리를 위한 Queue 버퍼를 설정합니다. 우선순위가 높은 데이터를 위한 Queue 입니다.

whatap.net_send_queue2_size
Default : 5000
Type : Int
Tcp 데이터 처리를 위한 Queue 버퍼를 설정합니다. 프로파일 데이터를 위한 Queue 입니다.

whatap.queue_udp_enabled
Default : false
Type : Boolean
Udp 데이터 수신 처리에 Channel을 대신해서 별도의 Queue를 사용합니다.

whatap.queue_udp_size
Default : 2048
Type : Int
Udp 데이터 수신 버퍼를 설정합니다. Channel, Queue 버퍼를 설정합니다.

whatap.queue_udp_read_thread_count
Default : 3
Type : Int
Udp 데이터 수신을 처리하는 쓰레드의 개수를 지정합니다. 개수를 늘리면 UDP 버퍼에서 좀 더 빠르게 데이터를 읽어들릴 수 있습니다.

whatap.queue_udp_overflowed_size
Default : 4096
Type : Int
Udp 데이터 수신에 Queue를 사용하는 경우, overflowed Queue 버퍼 크기를 설정합니다.

whatap.queue_udp_process_thread_count
Default : 1
Type : Int
Udp 데이터를 처리하는 쓰레드의 개수를 지정합니다. 개수를 늘리면 UDP에서 읽은 데이터를 좀 더 빠르게 처리할 수 있습니다.

whatap.queue_profile_enabled
Default : false
Type : Boolean
Profile 정보 처리에 Channel을 대신해서 별도의 Queue를 사용합니다.

whatap.queue_profile_size
Default : 512
4.5. 에이전트 로그 관리

whatap.log_keep_days
Default : 7
Type : Day
로그파일 보관기간을 설정합니다.

4.6. 에이전트 알림

whatap.hitmap_horiz_event_enabled
Default : false
Type : Boolean
이벤트 가로 라인 활성화 여부를 설정합니다.
히트맵의 응답시간 분포상, 응답시간이 지정시간(whatap.hitmap_horiz_event_basetime) 이상인 트랜잭션이 동일 응답시간 구간에 연속하여 지정시간(whatap.hitmap_horiz_event_duration) 동안 발생할 경우 알림이 발생합니다.

• 연속: X축(5초) 단위의 분포가 연속된 선으로 나타날 경우 (중간에 끊김이 있는 경우는 해당되지 않습니다.)

whatap.hitmap_horiz_event_duration
Default : 30000
Type : Millisecond
가로줄 현상의 지속시간 설정합니다.

whatap.hitmap_horiz_event_basetime
Default : 10000
Type : Millisecond
응답시간의 기준을 설정합니다.

whatap.hitmap_horiz_event_error_only
Default : false
Type : Boolean
에러가 발생한 경우만 이벤트 알림을 발생합니다.
whatap.hitmap_horiz_event_interval
Default: 10000
Type: Millisecond
이벤트 알림 발행 시에 중복 이벤트 억제 시간을 지정합니다. 해당 시간 내에는 다시 발행되지 않습니다.

whatap.hitmap_ver_event_enabled
Default: false
Type: Boolean
이벤트 세로 라인 활성화 여부를 설정합니다. 히트맵의 응답시간 분포상, X축 단일 구간(5초)에 세로줄이 나타나는 현상이 지정시간(whatap.hitmap_ver_event_duration)에 해당하는 시간 동안 지속될 경우 알림이 발생합니다.

• 세로 줄 현상: 2.5초 ~ 80초를 표현하는 전체 100개 구간 중, 경고 구간이 마킹된 경우 [경고 알림]이 발생하며, 동일한 규칙으로 위험구간이 마킹된 경우 [위험 알림]이 발생합니다.

whatap.hitmap_ver_event_duration
Default: 30000
Type: Millisecond
세로줄 현상의 지속시간을 설정합니다.

whatap.hitmap_ver_event_error_only
Default: false
Type: Boolean
에러가 발생한 경우만 이벤트 알림을 발행합니다.

whatap.hitmap_ver_event_warning_percent
Default: 60
Type: Int
세로줄 현상의 경고 구간 시작 지점입니다. 종료지점은 위험구간 시작 지점 전까지 입니다.

whatap.hitmap_ver_event_fatal_percent
Default: 80
Type: Int
세로줄 현상의 위험 구간 시작 지점입니다.

whatap.hitmap_ver_event_interval
Default: 300000
Type: Millisecond
이벤트 알림 발행 시에 중복 이벤트 억제 시간을 지정합니다. 해당 시간 내에는 다시 발행되지 않습니다.

4.7. 트랜잭션 프로파일링

whatap.profile_step_normal_count
Default: 1000
Type: Int
트랜잭션 프로파일의 최대 스텝 수를 지정합니다.

whatap.profile_step_max_count
Default: 1024
Type: Int
수집 가능한 프로파일 스텝의 최대 개수를 설정합니다. 수집된 프로파일 스텝 수가 이것을 초과하면 이후 수집되는 스텝들은 모두 버려집니다.
whatap.profile_step_heavy_count
Default: 1000
Type: Int
기본 스텝의 수집 개수가 초과되면, 실행 시간이 profile_step_heavy_time을 초과하는 스텝만 수집합니다. 해당 스텝의 수집 가능한 개수를 설정합니다. Default 설정일 경우 profile_step_normal_count가 800 이면 최대 200개의 스텝이 수집됩니다.

whatap.profile_step_heavy_time
Default: 100
Type: Millisecond
Heavy한 스텝의 기준을 지정 합니다. 지정된 값보다 수행시간이 긴 경우 profile_step_normal_count 를 초과하는 경우라도 profile_step_heavy_count 이내에서 기록됩니다.

whatap.profile_basetime
Default: 500
Type: Millisecond
트랜잭션이 설정된 값 이하의 시간에 종료된 경우 프로파일 정보를 수집하지 않습니다. 단, 5 분단위로 최초 호출된 URL, 에러가 발생한 트랜잭션에 대한 프로파일 정보는 수집됩니다.

whatap.query_string_enabled
Default: false
Type : Boolean
트랜잭션 URL의 쿼리 스트링을 함께 수집하는 기능을 활성화합니다. whatap.query_string_urls 에 등록된 URL만 적용됩니다.

whatap.query_string_urls
Default : NONE
Type : String
트랜잭션에서 쿼리 스트링을 수집 할 URL들을 등록합니다.

whatap.whatap.profile_compile_file_enabled
Default :false
Type : Boolean
compile_file, compile_string 함수의 추적 여부를 설정합니다.
스크립트 파일 경로와 실행시간을 확인할 수 있습니다. NOTE: IIS 재시작 후 적용됩니다.

whatap.profile_compile_file_basetime
Default : 200
Type : Millisecond
실행시간의 기준을 설정합니다. 설정한 시간을 초과하여 실행되는 경우만 정보를 수집합니다.

whatap.profile_compile_file_basetime을 너무 작게 설정할 경우 너무 많이 수집하게 되어 서버에 부하를 줄 수 있습니다.

IIS 재시작 후 적용됩니다.

4.8. HTTP 트랜잭션 추적
whatap.profile_http_header_enabled
Default : false
Type : Boolean
프로파일 내역에 http 헤더 정보를 기록하고자 할 때 사용합니다.
whatap.profile_http_parameter_enabled
Default: false
Type: Boolean
프로파일 내역에 http 파라미터 정보를 기록하고자 할 때 사용합니다. 파라미터는 별도 보안키를 입력해야 조회 할 수 있습니다.

whatap.profile_http_parameter_url_prefix
Default: /
Type: String
프로파일 내역에 http 파라미터 정보를 기록할 대상 URL의 prefix를 정의 할 때 사용합니다.

whatap.profile_http_host_enabled
Default: false
Type: Boolean
트랜잭션의 Host 정보를 출력합니다. False일 경우 트랜잭션의 URL에 URI만 표기하고 True일 경우 /xxx.aaa.com/URL 형식으로 출력됩니다.

whatap.biz_exceptions
Default: Empty
Type: String
특정 Error 또는 Exception을 Business Exception으로 지정합니다. 허트맵에서는에러로 표기되지 않고 통계정보에만 수집됩니다. 수집되는 Error Class명과 동일하게 적용됩니다. 콤마를 구분자로 하여 여러개를 등록할 수 있습니다.

whatap.ignore_exceptions
Default: NONE
Type: String
무시할 Exception을 등록합니다. 여기에 등록되면 에러 자체가 무시됩니다.

whatap.ignore_exceptions
Exception의 경우 Custom Exception Class가 정의되어 있지 않으면, 기본으로 "Exception"으로 수집됩니다. 특정 Exception을 구분하기 위해서는 Custom Exception Class를 정의하고 해당 이름을 지정하면 제외 시킬 수 있습니다.

4.9. DB, SQL

whatap.trace_sql_normalize_enabled
Default: true
Type: Boolean
SQL문에서 리터럴 부분을 추출하여 SQL문을 정규화 하는 기능을 활성화 합니다. 리터럴 부분은 "#"으로 표기됩니다.

whatap.profile_sql_resource_enabled
Default: false
Type: Boolean
프로파일에서 SQL이 수집될 때 해당 스텝에서 사용한 CPU와 메모리 사용량을 추적합니다.

whatap.profile_sql_param_enabled
Default: false
Type: Boolean
프로파일 내역에 SQL 파라미터 정보를 기록하고자 할 때 사용합니다. 파라미터는 별도 보안키를 입력해야 조회 할 수 있습니다.

보안 키는 WAS서버 %PROGRAMFILES%\WhaTap .NET\paramkey.txt 파일내에 6자리로 지정합니다.
paramkey.txt 파일이 존재하지 않는 경우 편집 값으로 자동 생성됩니다.
4.10. HTTPC, API Call

whatap.profile_httpc_resource_enabled
Default : false
Type : Boolean
프로필에서 HTTP Call 스테ップ이 수집될 때 해당 스테ップ에서 사용한 CPU 와 메모리 사용량을 추적합니다.

whatap.trace_httpc_normalize_enabled
Default : true
Type : Boolean
트랜잭션 내 HTTPC URL을 파싱하여 정규화하는 기능을 활성화합니다.

whatap.trace_httpc_normalize_urls
Default : NONE
Type : String
정규화 할 HTTPC URL 패턴을 정의 합니다. 호출 URL 패턴을 파싱하여 패스파라미터를 제거합니다.

  ex) /a/{v}/b 라고 선언하면 a/123/b ⇒ a/{v}/b 로 치환한다
여러 개를 등록할 때는 괄마(,)를 사용합니다. 치환패턴 정리 후 보완필요

4.11. 사용자 수

whatap.trace_user_enabled
Default : true
Type : Boolean
실시간 사용자 집계 여부를 지정합니다.

  IIS 재시작 후 적용됩니다.

whatap.trace_user_using_ip
Default : false
Type : Boolean
실시간 사용자의 구분을 IP로 하고자 하는 경우 설정합니다. 이 데이터는 Real Time User에서 확인 가능합니다.

  IIS 재시작 후 적용됩니다.

whatap.trace_user_header_ticket
Default : false
Type : Boolean
사용자의 구분을 HTTP 헤더의 특정 값으로 구분하고 싶을 때 사용 할 HTTP 키를 지정합니다. 단, trace_user_using_ip 배타적으로 설정을 동시에 적용할 수 없다. HTTP키를 찾지 못하는 경우 Cookie를 기준으로 사용자를 구분합니다.

  IIS 재시작 후 적용됩니다.

whatap.trace_user_set_cookie
Default : false
Type : Boolean
사용자의 구분을 하기 위해 쿠키에 “WHATAP” 키 이름으로 사용자 구분 값을 설정한다. trace_user_using_ip 설정 되는 경우 반영되지 않습니다.
trace_user_header_ticket이 설정되는 경우 header key를 찾지 못하는 경우 설정됩니다. HTTP키를 찾지 못하는 경우 Cookie를 기준으로 사용자를 구분합니다.
whatap.trace_http_client_ip_header_key
Default : ""
Type : String
Client IP(Remote IP)의 정보를 특정 HTTP 헤더의 값으로 변경하여 설정하는 기능을 추가하였습니다. Proxy(프락시) 환경에서 X-Forwarded-For 헤더 값을 Client IP 로 설정할 수 있습니다.

whatap.trace_active_transaction_slow_time
Default : 3000
Type : Millisecond
수집 정보를 확인하는 대시보드의 액티브트랜잭션 아크이콜레이져 그래프에서 Slow 구간으로 표기될 수 있는 트랜잭션 응답 시간의 기준을 지정합니다. 트랜잭션의 응답시간이 지정 시간을 초과할 경우 Slow 액티브트랜잭션의 개수에 포함됩니다.

whatap.trace_active_transaction_very_slow_time
Default : 8000
Type : Millisecond

whatap.trace_active_transaction_lost_time
Default : 30000
Type : Millisecond
트랜잭션의 종료를 기다리는 제한 시간. 5분안에 트랜잭션이 종료되지 않는 경우 트랜잭션 정보는 수집하지 않습니다. 트랜잭션의 프로파일 정보에서 “Lost Connection”을 확인할 수 있습니다.

whatap.trace_normalize_enabled
Default : true
Type : Boolean
트랜잭션 URL을 일반화하는 기능을 활성화합니다.

whatap.trace_normalize_urls
Default : NONE
Type : String
트랜잭션 URL을 일반화하는 대상 URL을 지정합니다. 호출 URL패턴을 변환할 때 필요한 파라미터를 제거합니다. 예를 들어 “/a/{v}/b” 라고 설정하면 해당 형식으로 호출되는 트랜잭션 URL은 “/a/v/b” 형식으로 변환됩니다. 여러 개를 등록 할 때는 콤마(,)를 사용합니다.

whatap.stat_domain_enabled
Default : false
Type : Boolean
도메인별 트렌잭션 통계를 수집합니다.

whatap.stat_domain_max_count
Default : 7000
Type : Int
5분동안 수집할 도메인별 트랜잭션 통계의 최대 레코드 수입니다.

**whatap.stat_mtrace_enabled**
Default : false  
Type : Boolean  
멀티 서버 트랜잭션에서 Callser&Callee간에 버전별 의존성 통계를 수집합니다.

**whatap.mtrace_spec**
Default: v1  
Type : String  
현 인스턴스의 애플리케이션 버전을 지정합니다. 임의의 문자열을 지정할 수 있습니다. 이 데이터는 호출통계를 위해 사용됩니다.

```
  IIS 재시작 후 적용됩니다.
```

**whatap.stat_mtrace_max_count**
Default : 7000  
Type : Int  
멀티 서버 트랜잭션에서 Callser&Callee간에 버전별 의존성 통계의 최대 레코드 수입니다.

**whatap.stat_referer_enabled**
Default : false  
Type : Boolean  
Referer별 트랜잭션 통계를 수집합니다.

**whatap.stat_referer_max_count**
Default : 7000  
Type : Int  
5분동안 한 JVM에서 수집할 Referer별 트랜잭션 통계의 최대 레코드 수입니다.

**whatap.tx_caller_meter_enabled**
Default : false  
Type : Boolean  
트랜잭션 정보로 토폴로지 통계를 생성합니다.

**whatap.sql_dbc_meter_enabled**
Default : false  
Type : Boolean  
Database 연결 정보로 토폴로지 통계를 생성합니다.

**whatap.httpc_host_meter_enabled**
Default : false  
Type : Boolean  
Http 외부 호출 정보로 토폴로지 통계를 생성합니다.

**whatap.okind**
Default : ""  
Type : String  
그룹 토폴로지를 위해 임의로 부여합니다. 여러 에이전트를 같이 지정하여 해당 그룹명으로 토폴로지를 확인 할 수 있습니다.