

JAVA AGENT Plugin V0.0.4

Java bytecode instrumentation 옵션 중 plugin 활용과 관련한 내용에 대하여 기술한다.

1. Java bytecode instrumentation 대상 설정

* 활성화를 위해서는 피모니터링 서버의 재기동을 필요로 한다.

hook_trace_helper_patterns

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

* plugin을 활용한 커스터마이징 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입된다.

`$WHATAP_HOME/plugin/TraceHelperStart.x`

`$WHATAP_HOME/plugin/TraceHelperEnd.x`

hook_trace_helper_end_patterns

default:

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

* plugin을 활용한 커스터마이징 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입된다.

`$WHATAP_HOME/plugin/TraceHelperEnd.x`

hook_trace_helper_start_patterns

default:

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

* plugin을 활용한 커스터마이징 된 profile 정보 수집을 위한 용도로 하기 plugin 코드가 주입된다.

`$WHATAP_HOME/plugin/TraceHelperStart.x`

설정 예시

`whatap.bytecode.instrument.PluginTestA.testA,whatap.bytecode.instrument.PluginTestB.testB`

`*PluginTestA.testA,whatap.bytecode.instrument.PluginTestB.*`

`*.testA,*PluginTestB.testB`

`*.*`

2. Plugin 유형

WhaTap Agent 설치 경로 하위에 plugin 디렉토리를 생성한다.

디렉토리 하위에 .x 파일을 생성 시 적용 가능한 파일 유형, 용도, 활용 가능 옵션은 다음과 같다.

| 파일명 | 클래스명.메소드명 | 파라메타 | 설명 및 사용가능키 |
|---|-------------------------|-------------|--|
| 공통 | 공통 | *1) \$ctx | setAttribute(key, value) getAttribute(key, value) profile(step명) profile(step명, step내용) profile(step명, step내용, step 수행 시간, 추가 값) HTTP Request로 부터의 추출 값 service() 서비스명? serviceHash() 서비스 해쉬값? remotelp() remote IP isError() 에러 발생 여부 login() ? httpMethod() httpQuery() httpContentType() userAgent() status() |
| | | *2) \$point | class1 클래스명 method 메소드명 desc this1 Hooking 대상 클래스/메소드명 args 유입 파라메타(Object[]) |
| | | *3) \$point | class1 클래스명 method 메소드명 desc this1 Hooking 대상 클래스/메소드명 return1 반환 결과 |
| HTTP 프로토콜 이외의 서비스(TCP 또는 배치)를 Hooking 한 경우, 추가적 정보를 필요로 할 경우 활용 | | | |
| AppServiceStart.x | AppServiceStart.process | \$ctx | *1) \$ctx 참조 |
| | | \$point | *2) \$point 참조 |
| AppServiceEnd.x | AppServiceEnd.process | \$ctx | *1) \$ctx 참조 |
| | | | |
| Extension.x | Extension.process | \$pack | values - MapValue getPackType() read() - Pack size() |
| HTTP Outbound Call 정보를 추가 활용 | | | |

| | | | |
|--------------------------------|--------------------------|--------|---|
| HttpCallStart.x | HttpCallStart.process | \$ctx | *1) \$ctx 참조 |
| | | \$req | url() host() port() isOk() error() |
| HttpCallEnd.x | HttpCallEnd.process | \$ctx | *1) \$ctx 참조 |
| | | \$resp | header(key) pcode() oid() status() statusReason() exception() |
| HTTP Service의 정보를 추가 활용 | | | |
| HttpServiceStart.x | HttpServiceStart.process | \$ctx | *1) \$ctx 참조 |
| | | \$req | getCookie(key) getRequestURI() getRemoteAddr() getMethod() getQueryString() getParameter(key) getAttribute(key) getHeader(key) getParameterNames() getHeaderNames() getSession() - Session Wrapper 객체 getSessionNames() getSessionAttribute(key) isOk() error() |
| | | \$res | getContentType() getCharacterEncoding() isOk() error() |
| HttpServiceEnd.x | HttpServiceEnd.process | \$ctx | *1) \$ctx 참조 |
| | | \$req | getCookie(key) getRequestURI() getRemoteAddr() getMethod() getQueryString() getParameter(key) getAttribute(key) |

| | | | |
|--|--|-------|--|
| | | | getHeader(key) getParameterNames() getHeaderNames() getSession() - Session Wrapper 객체 getSessionNames() getSessionAttribute(key) isOk() error() |
| | | \$res | getContentType() getCharacterEncoding() isOk() error() |

Hooking 지점을 지정하여 Plugin 을 통해 해당 지점의 정보를 활용

| | | | |
|--------------------|--------------------------|---------|----------------|
| TraceHelperStart.x | TraceHelperStart.process | \$ctx | *1) \$ctx 참조 |
| | | \$point | *2) \$point 참조 |
| TraceHelperEnd.x | TraceHelperEnd.process | \$ctx | *1) \$ctx 참조 |
| | | \$point | *3) \$point 참조 |

* 프로파일 기록을 위한 기능을 제외하고, 참조 기능만 명시함.

3. Plugin 코드 작성

plugin 디렉토리 하위의 코드 작성 예시입니다. 코드 작성 시에는 명시적인 type casting을 수행하도록 코드를 작성한다.

\$WHATAP_HOME/plugin/TraceHelperStart.x

```
String prefix = $point.class1 + "." + $point.method;
$ctx.setAttribute(prefix + "st", new Long(System.currentTimeMillis()));
$ctx.profile(prefix + " Start", new java.util.Date().toString());
```

\$WHATAP_HOME/plugin/TraceHelperEnd.x

```
String prefix = $point.class1 + "." + $point.method;
long st = ((Long) $ctx.getAttribute(prefix + "st")).longValue();
long gap = System.currentTimeMillis() - st;

StringBuilder sb = new StringBuilder();
sb.append(new java.util.Date().toString() + " (Gap:" + gap + " milliseconds)");
$ctx.profile(prefix + " End", sb.toString());
```

\$WHATAP_HOME/whatap.conf

```
hook_trace_helper_patterns=triosyi.socket.SocketTestBootApplication.testSocket
```

