

Jwebap User Guide

leadyu (yu-lead@163.com)

1 Introduction

Jwebap is an open-source profiler tool for j2ee application. It is pure java-based and working on any computer with jdk 1.4 and up. It is also very easy to use because it only launches two files. Following are some of its features:

High Performance: Jwebap uses ASM (A Java bytecode engineering library) to inject class to achieve runtime information. The function is very low-overhead. And now it is deployed in some China Telecom product systems with ten Millions users.

Pure Java: Most profilers use the JVMPi (Java Virtual Machine Profiling Interface) and some localized interface to monitor runtime details inside JVM. However, Jwebap is written entirely in java. It has no dependency on JVM or OS. So you can easily deploy it in any platform with very low-overhead.

Plugin Architecture: Jwebap is based on a plug-in architecture. Every trace analysis tool is deployed in jwebap as a plugin. You can configure what plugin you need use. You can do timing filter on method execution, database statement, http request, etc. You can also view the leak connections, view the heap stack in execution, and count average execution time, over-time rate, etc. A plugin can

be extended easily under the plug-in architecture, so Jwebap can implement more analysis components and reports in the future.

Easy To Use: When you use Jwebap, you will find it is easy to use. It has simple configure file, web console, and helpful analysis report.

2 How to use?

2.1 Step I: deploy jarfiles

Include jwebap.jar and it's dependency libraries to your application classes path. It is generally 'WEB-INF/lib/' if your application is just a web module, or the server lib path if your application is an EJB application with EAR.

2.2 Step II: configure

And then change your web.xml:

Add context-param to locate jwebap.xml

```
<context-param>
    <param-name>jwebap-config</param-name>
    <param-value>/WEB-INF/jwebap.xml</param-value>
</context-param>
```

Add Jwebap start listener at the first of all listeners

```
<listener>
    <listener-class>org.jwebap.startup.JwebapListener</listener-class>
</listener>
```

Add http filter for HttpComponent

```

<filter>
    <filter-name>PageDetectFilter</filter-name>
    <filter-class>org.jwebap.plugin.tracer.http.DetectFilter</filter-class>
    <init-param>
        <param-name>excludeUrls</param-name>
        <param-value>/detect;/detect/*.*.js;*.jpg;*.htm;*.html;*.gif;*.png;*.css;*.swf</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>PageDetectFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

Add Servlet for Jwebap console

```

<servlet>
    <servlet-name>detect</servlet-name>
    <servlet-class>org.jwebap.ui.controler.JwebapServlet</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>detect</servlet-name>
    <url-pattern>/detect/*</url-pattern>
</servlet-mapping>

```

2.3 Step III: lanunch jwebap!

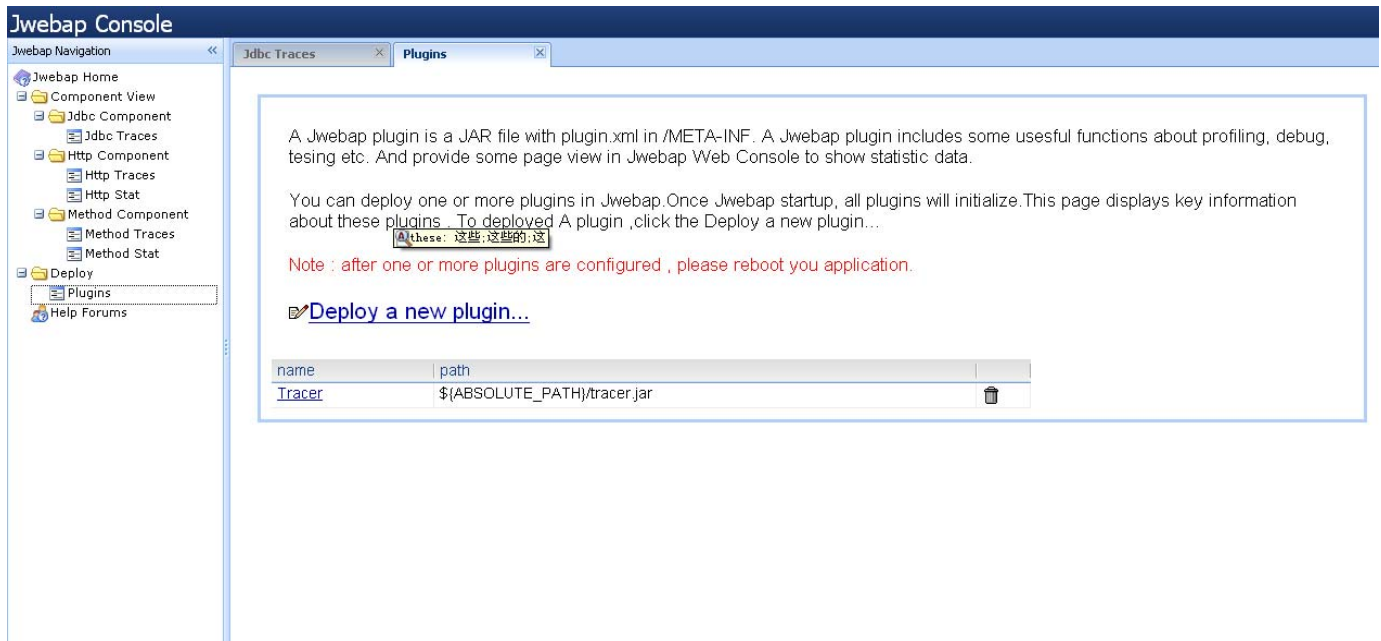
Lanunch tracer.jar in your application classes path(the default plugin in jwebap defined in jwebap.xml plugin-ref tag)

Now ,startup your application ! if you config correctly ,you can access JwebapServlet to see the Jwebap Console .

2.4 Step IV: config tracer-plugin

When you see the jwebap console ,that's not enough . We have not any plugin in it . Next, we should deploy plugins in jwebap

console.



Jwebap implement a default plugin 'tracer'. It has been deployed default when jwebap startup. So, we can see it in jwebap console → deploy → plugins. And then, we should modify some parameter with this plugin.

Click 'Tracer', and choice component which you will config, here has three component :

1) HttpComponent

trace-filter-active-time: (ms) timings filter's over time.

trace-max-size: max over-time trace size

2) MethodComponent

trace-filter-active-time: (ms) timings filter's over time.

trace-max-size: max over-time trace size

detect-clazzs: package name and class name that detected by

MethodComponent, like 'test.*;test.Test' , divided by ';'.

3) JdbcComponent

trace-filter-active-time: (ms) timings filter's over time.

trace-max-size: max over-time trace size

connection-listener: connection listener, divided by ';'.

driver-clazzs:

Local datasource: set your <ConnectionManagerClass>,or the connection pool 's datasource. If you have more than one class ,divided by ';'.

c3p0:com.mchange.v2.c3p0.ComboPooledDataSource;

dbcp: org.apache.commons.dbcp.BasicDataSource

also,other class. Jwebap will inject **driver-clazzs**,and detect any connection and datasource object it's method return.

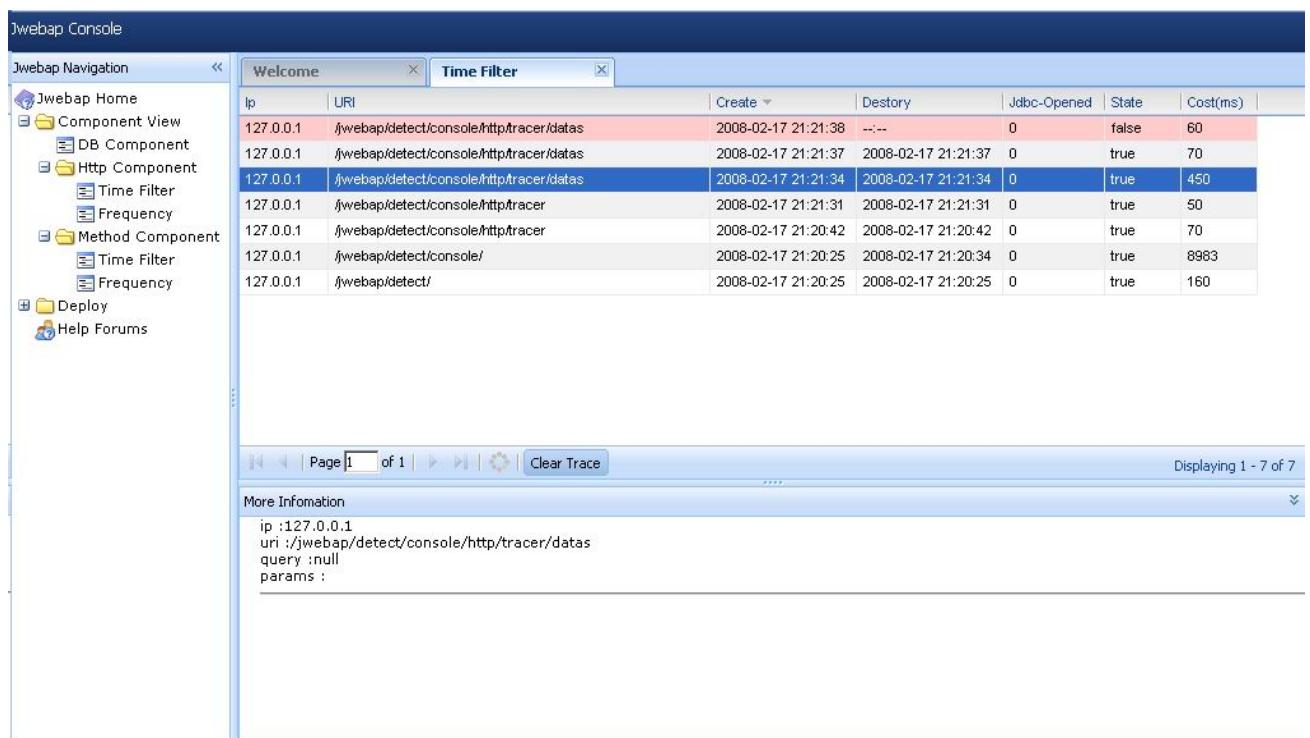
Note: 'driver-clazzs =<jdbc driver>' is deprecated. Beacause of connection pool, set 'driver-clazzs =<jdbc driver>', jwebap will find out all connection is leaked.

JNDI datasource: If your application uses jndi datasource, you can set the class which manages connections in your application as driver, e.g.: 'com.china.telecom.ConnectionManager'. Else if you use spring to get jndi datasource ,you also can set **driver-clazzs=org.springframework.jndi.JndiObjectFactoryBe**

an.JdbcComponent will inject this class to proxy all connection
the class's method return.

3 ScreenShot

3.1 Http Traces



The screenshot shows the Jwebap Console interface. On the left is a navigation pane with a tree view containing: Jwebap Home, Component View, DB Component, Http Component, Time Filter, Frequency, Method Component, Time Filter, Frequency, Deploy, and Help Forums. The main area has two tabs: 'Welcome' and 'Time Filter'. The 'Time Filter' tab is active and displays a table of HTTP traces. The table has columns: Ip, URI, Create, Destory, Jdbc-Opened, State, and Cost(ms). Below the table is a pagination bar showing 'Page 1 of 1' and a 'Clear Trace' button. At the bottom, there is a 'More Information' section with details for the selected trace.

Ip	URI	Create	Destory	Jdbc-Opened	State	Cost(ms)
127.0.0.1	/jwebap/detect/console/http/tracer/datas	2008-02-17 21:21:38	--	0	false	60
127.0.0.1	/jwebap/detect/console/http/tracer/datas	2008-02-17 21:21:37	2008-02-17 21:21:37	0	true	70
127.0.0.1	/jwebap/detect/console/http/tracer/datas	2008-02-17 21:21:34	2008-02-17 21:21:34	0	true	450
127.0.0.1	/jwebap/detect/console/http/tracer	2008-02-17 21:21:31	2008-02-17 21:21:31	0	true	50
127.0.0.1	/jwebap/detect/console/http/tracer	2008-02-17 21:20:42	2008-02-17 21:20:42	0	true	70
127.0.0.1	/jwebap/detect/console/	2008-02-17 21:20:25	2008-02-17 21:20:34	0	true	8983
127.0.0.1	/jwebap/detect/	2008-02-17 21:20:25	2008-02-17 21:20:25	0	true	160

Page 1 of 1 Clear Trace Displaying 1 - 7 of 7

More Information

ip :127.0.0.1
uri :/jwebap/detect/console/http/tracer/datas
query :null
params :

3.2 Method Traces

Jwebap Console

Jwebap Navigation

- Jwebap Home
- Component View
 - Jdbc Component
 - Jdbc Traces
 - Http Component
 - Http Traces
 - Http Stat
 - Method Component
 - Method Traces
 - Method Stat
- Deploy
- Help Forums

Method Traces

Thread	Method	Create	Destory	Jdbc-Opened	State	Cost(ms)
http-9090-Processor22#30960243	public static java.lang.String[] test.Test.sm(java.l...	2008-02-23 23:45:08	2008-02-23 23:45:08	0	closed	0
http-9090-Processor20#5415651	public static java.lang.String[] test.Test.sm(java.l...	2008-02-23 23:45:10	2008-02-23 23:45:10	0	closed	0
http-9090-Processor23#21228916	public static java.lang.String[] test.Test.sm(java.l...	2008-02-23 23:45:11	2008-02-23 23:45:11	0	closed	0

Page 1 of 1 Clear Trace

More Information

```
org.jwebap.core.StackTail.(StackTail.java:11)
org.jwebap.core.Trace.(Trace.java:21)
org.jwebap.core.StatistableTrace.(StatistableTrace.java:62)
org.jwebap.plugin.method.MethodTrace.(MethodTrace.java:29)
org.jwebap.plugin.method.TraceMethodHandle.invoke(TraceMethodHandle.java:42)
test.Test.sm(Test.java)
org.apache.jsp.test.leaktest_jsp._jspService(leaktest_jsp.java:54)
org.apache.jasper.runtime.HttpJspBase.service(HttpJspBase.java:94)
javax.servlet.http.HttpServlet.service(HttpServlet.java:802)
org.apache.jasper.servlet.JspServletWrapper.service(JspServletWrapper.java:324)
org.apache.jasper.servlet.JspServlet.serviceJspFile(JspServlet.java:292)
org.apache.jasper.servlet.JspServlet.service(JspServlet.java:286)
```

Thread Stack Arguments

3.3 Jdbc Traces

Jwebap Console

Jwebap Navigation

- Jwebap Home
- Component View
 - Jdbc Component
 - Jdbc Traces
 - Http Component
 - Http Traces
 - Http Stat
 - Method Component
 - Method Traces
 - Method Stat
- Deploy
- Help Forums

Jdbc Traces

Thread	Create	Destory	State	Cost(ms)
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#15436652	2008-02-26 14:18:11	---	opened	12344
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#682921	2008-02-26 14:18:20	2008-02-26 14:18:20	closed	0
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	47
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	15
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	16
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#6467249	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	0
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#6467249	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	0
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#6467249	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	125

Page 1 of 2 Clear Trace

More Information

```
org.jwebap.core.StackTail.(StackTail.java:11)
org.jwebap.core.Trace.(Trace.java:21)
org.jwebap.core.StatistableTrace.(StatistableTrace.java:62)
org.jwebap.plugin.jdbc.TraceDetectConnection.(TraceDetectConnection.java:38)
org.jwebap.plugin.jdbc.JdbcDriverInjectHandle.invoke(JdbcDriverInjectHandle.java:40)
com.ztesoft.common.dao.DAOUtils.getDBConnection(DAOUtils.java)
com.ztesoft.crm.ps.sql.dao.CommonDAOImpl.getVerUpdateInfo(CommonDAOImpl.java:26)
com.ztesoft.crm.ps.sql.service.SqSearchService.getVerUpdateInfo(SqSearchService.java:1152)
sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:29)
```

Stack Traces Thread Stack SQLs

leak connection

Jwebap Console

Jwebap Navigation << Jdbc Traces Method Traces Http Traces Method Stat

- Jwebap Home
 - Component View
 - Jdbc Component
 - Jdbc Traces
 - Http Component
 - Http Traces
 - Http Stat
 - Method Component
 - Method Traces
 - Method Stat
 - Deploy
 - Help Forums

Thread	Create	Destory	State	Cost(ms)
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#15436652'	2008-02-26 14:18:11	---	opened	12344
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#682921'	2008-02-26 14:18:20	2008-02-26 14:18:20	closed	0
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	47
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	15
ExecuteThread: '12' for queue: 'weblogic.kernel.Default#11944035'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	16
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#8467249'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	0
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#8467249'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	0
ExecuteThread: '14' for queue: 'weblogic.kernel.Default#8467249'	2008-02-26 14:18:17	2008-02-26 14:18:17	closed	125

Page 1 of 2 Clear Trace Displaying 1 - 20 of 35

More Information

Statement: 2008-02-26 14:18:20/2008-02-26 14:18:20
 sql= SELECT first 1 ver_release_id, ver_release_title, ver_release_person, ver_release_content, ver_release_time from crmdb@test_dbs:informix.ver_update_info order by ver_release_time desc

Thread Stack SQLs

3.4 Plugin Deploy

Jwebap Console

Jwebap Navigation << Jdbc Traces Plugins

- Jwebap Home
 - Component View
 - Jdbc Component
 - Jdbc Traces
 - Http Component
 - Http Traces
 - Http Stat
 - Method Component
 - Method Traces
 - Method Stat
 - Deploy
 - Plugins
 - Help Forums

A Jwebap plugin is a JAR file with plugin.xml in /META-INF. A Jwebap plugin includes some useful functions about profiling, debug, testing etc. And provide some page view in Jwebap Web Console to show statistic data.

You can deploy one or more plugins in Jwebap. Once Jwebap startup, all plugins will initialize. This page displays key information about these plugins. To deployed A plugin, click the Deploy a new plugin...

Note: after one or more plugins are configured, please reboot you application.

[Deploy a new plugin...](#)

name	path
Tracer	\$(ABSOLUTE_PATH)\tracer.jar

4 Dependency

- 1) JDK1.4 and up
- 2) commons-digester-1.7.jar

3) commons-betwixt-0.8.jar

4) commons-beanutils.jar

5) commontemplate-0.8.1.jar

6) commons-logging.jar

configuration file:

1) jwebap.xml

Middleware Support:

Tomcat4+, WebLogic8+, WAS6+, JBoss