



Atlassian 로그 수집 가이드

Project Name:

Department:

Focus Area: ALM Product

Product/Process:

Prepared By:

Document Owner(s)	Project/Organization Role
Kapil Jeon	Atlassian Expert

Project Status Report Version Control

Version	Date	Author	Change Description
v1.0	16/04/25	Kapil Jeon	
[Version #]	[mm/dd/yy]	[Change owner]	• [Change 1]

Confidential

Atlassian Experts

Last printed on 6/7/2016 5:00:00 PM

TABLE OF CONTENTS

1	목적	4
2	JIRA	4
2.1	Catalina.out.....	4
2.2	Atlassian Home Directory Log	4
2.3	Access Log.....	4
2.4	Thread Dump	5
2.5	GC Log.....	5
3	CONFLUENCE	6
3.1	Catalina.out.....	6
3.2	Atlassian Home Directory Log	6
3.3	Debug Log.....	6
3.4	Thread Dump	7
3.5	GC Log.....	7
4	CROWD	8
4.1	Catalina.out.....	8
4.2	Atlassian Home Directory Log	8
4.3	Access Log.....	8
4.4	Thread Dump	9
4.5	GC Log.....	9

5	FISHEYE	10
5.1	Catalina.out	10
5.2	Atlassian Home Directory Log	10
5.3	Access Log.....	10
5.4	Thread Dump	11
5.5	GC Log.....	12
6	기타.....	13
6.1	MySQL SQL Log	13
6.2	Crontab 설정 추가.....	13

1 목적

본 문서는 Atlassian 의 제품군 중 다음의 네 가지 (Jira, Confluence, Crowd, Fisheye)에서 로그를 남기는 방법들에 대해 기술하기 위해 작성되었습니다.

2 Jira

2.1 Catalina.out

위치

<installation directory>/logs/catalina.out

(Ex: /opt/atlassian/jira/logs/catalina.out)

2.2 Atlassian Home Directory Log

위치

<data directory>/log/atlassian_jira.log

(Ex: /var/atlassian/application-data/jira/log/atlassian_jira.log)

2.3 Access Log

위치

<installation directory>/logs/access_log.<날짜>

(Ex: /opt/atlassian/jira/logs/access_log.<날짜>)

2.4 Thread Dump

ps -ef | grep 으로 필요한 프로세스를 잡고 tail -f catalina.out 한 상태에서 kill -3 <pid> 로 10 초 간격으로 5~6 회 정도 시도합니다.

```
#ps -ef | grep java | grep jira | awk '{print $2}'
```

```
#tail -f <installation directory> /logs/catalina.out (쓰레드 덤프가 잘 받아지는지 확인)
```

```
# kill -3 <pid> 로 10 초 간격으로 5~6 회
```

2.5 GC Log

setenv.sh 를 다음 내용으로 수정합니다

```
JVM_EXTRA_ARGS="-XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -verbose:gc -Xloggc:/var/atlassian/application-data/jira/log/jira-gc.log
```

```
(선택사항)-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M -XX:+PrintGCDetails
```

참조문서

<https://jira.atlassian.com/browse/JRA-30188>

3 Confluence

3.1 Catalina.out

위치

<installation directory>/logs/catalina.out

(Ex: /opt/atlassian/confluence/logs/catalina.out)

3.2 Atlassian Home Directory Log

위치

<data directory>/logs/atlassian_confluence.log

(Ex: /var/atlassian/application-data/confluence/logs/atlassian_confluence.log)

3.3 Debug Log

Confluence System Administration 메뉴의 Logging and Profiling 메뉴에서 Enable Profiling 클릭

Logging and Profiling

Performance Profiling
Profiling is currently OFF.
[Enable Profiling](#)

SQL Logging
[Disable SQL Logging](#)

Log4j Logging
loglevel.profileDesc
[Production](#) [Diagnostic](#)

OR:

Customise specific logging settings

Add New Entry

Class/Package Name	New Level
<input type="text"/>	INFO <input type="button" value="Add entry"/>

Existing Levels

Class/Package Name	Current Level	New Level
atlassian.plugin	DEBUG	DEBUG <input type="button" value="Remove"/>
bucket.user	DEBUG	DEBUG <input type="button" value="Remove"/>

Help Tips

Note that any changes you make are not persisted and will be lost when you restart Confluence.

Performance profiling:

Logs the speed of pages and will help with diagnosing performance problems. This results in large log files and should not be enabled for long periods.

SQL Logging:

Should only be enabled under direction from Atlassian support.

Log4j Logging:

The *Production* profile allows you to revert to the default setting.

The *Diagnostic* profile logs more information, useful for troubleshooting errors. This profile has a small performance overhead.

The *Custom* settings allows for logging of very specific information, usually under direction from Atlassian support.

Please see the [online help](#) for more information.

3.4 Thread Dump

ps -ef | grep 으로 필요한 프로세스를 잡고 tail -f catalina.out 한 상태에서 kill -3 <pid> 로 10 초 간격으로 5~6 회 정도 시도합니다.

```
#ps -ef | grep java | grep confluence | awk '{print $2}'
```

```
#tail -f <installation directory> /logs/catalina.out (쓰레드 덤프가 잘 받아지는지 확인)
```

```
# kill -3 <pid> 로 10 초 간격으로 5~6 회
```

3.5 GC Log

위치

<installation directory>/logs/gc-날짜

4 Crowd

4.1 Catalina.out

위치

<installation directory>/logs/catalina.out

(Ex: /opt/crowd/apache-tomcat/logs/catalina.out)

4.2 Atlassian Home Directory Log

위치

<data directory>/logs/atlassian_crowd.log

(Ex: /var/atlassian/application-data/crowd/logs/atlassian_crowd.log)

4.3 Access Log

{crowd-install}\apache-tomcat\conf\Catalina\localhost\crowd.xml 에 다음 내용을 추가합니다.

<Valve

 className="org.apache.catalina.valves.AccessLogValve"

 directory="\${catalina.base}/logs"

 prefix="access_log."

 fileDateFormat="yyyy-MM-dd"

 suffix=".log"

 pattern="%a %{Authorization}i %t "%m %U%q %H" %s %b %D
"%{Referer}i" "%{User-Agent}i","

/>

위치

```
tail -f /opt/atlassian/crowd/apache-tomcat/logs/access_log.2016-02-18.log
```

참조문서

<https://answers.atlassian.com/questions/261821/log-an-applications-usage-of-crowd>

4.4 Thread Dump

ps -ef | grep 으로 필요한 프로세스를 잡고 tail -f catalina.out 한 상태에서 kill -3 <pid> 로 10 초 간격으로 5~6 회 정도 시도합니다.

```
#ps -ef | grep java | grep crowd | awk '{print $2}'
```

```
# tail -f <installation directory>/apache-tomcat/logs/catalina.out (쓰레드 덤프가 잘 받아지는지 확인)
```

```
# kill -3 <pid> 로 10 초 간격으로 5~6 회
```

4.5 GC Log

/opt/atlassian/crowd/apache-tomcat/bin 의 setenv.sh 에 다음 내용을 추가합니다

```
-XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -verbose:gc -  
Xloggc:/var/atlassian/application-data/crowd/logs/crowd-gc.log
```

```
(선택사항)-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M -  
XX:+PrintGCDetails
```

5 Fisheye

5.1 Catalina.out

위치

<data directory>/var/log/fisheye.out

(Ex: /data/fishdata/var/log/fisheye.out)

5.2 Atlassian Home Directory Log

위치

<data directory>/var/log/atlassian-fisheye-<날짜>.log

(Ex: /data/fishdata/var/log/atlassian-fisheye-<날짜>.log)

5.3 Access Log

<FishEye install directory>/content/WEB-INF/jetty-web.xml 파일 생성 후 다음 내용을 추가합니다

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE Configure PUBLIC "-//Mort Bay Consulting//DTD Configure//EN"
```

```
"http://www.eclipse.org/jetty/configure.dtd" >
```

```
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
```

```
  <Set name="handler">
```

```
    <New class="com.cenqua.fisheye.web.jetty.FishEyeRequestLogHandler">
```

```
      <Set name="requestLog">
```

```
        <New id="RequestLogImpl" class="org.eclipse.jetty.server.NCSARequestLog">
```

```
<Arg><SystemProperty name="jetty.logs" default="./var/log/" />fisheye-access-  
yyyy_mm_dd.log</Arg>  
  
<Set name="retainDays">90</Set>  
  
<Set name="append">true</Set>  
  
<Set name="extended">false</Set>  
  
<Set name="LogTimeZone">GMT</Set>  
  
</New>  
  
</Set>  
  
</New>  
  
</Set>  
  
</Configure>
```

위치

```
<data directory>/var/log
```

(Ex: /data/fishdata/var/log/fisheye-access-2016_02_18.log)

5.4 Thread Dump

ps -ef | grep 으로 필요한 프로세스를 잡고 tail -f catalina.out 한 상태에서 kill -3 <pid> 로 10 초 간격으로 5~6 회 정도 시도합니다.

```
#ps -ef | grep java | grep crowd | awk '{print $2}'
```

```
# tail -f <data directory>/var/logs/fisheye.out (쓰레드 덤프가 잘 받아지는지 확인)
```

```
# kill -3 <pid> 로 10 초 간격으로 5~6 회
```

5.5 GC Log

/etc/profile 에 다음내용을 추가합니다.

```
export FISHEYE_OPTS="-Xms128m -Xmx1024m -XX:MaxPermSize=256m -Xss512k -  
XX:+PrintGCTimeStamps -XX:+PrintGCDateStamps -verbose:gc -  
Xloggc:/data/fishdata/var/log/fisheye-gc.log"
```

```
(선택사항)-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=2M -  
XX:+PrintGCDetails
```

참조문서

<https://confluence.atlassian.com/fishkb/fix-out-of-memory-errors-165609660.html#FixOutOfMemoryerrors-java.lang.OutOfMemoryError:requested32756bytesforChunkPool::allocate.OutOfswapspace?>

6 기타

6.1 MySQL SQL Log

/etc/my.conf 설정에 Binary Log 가 생성되는 위치 및 파일 명

```
#log_bin=/var/lib/mysql/mysql-bin
```

위와 같이 설정하면 /var/lib/mysql 디렉토리에 아래와 같은 파일이 생성됩니다.

...

```
-rw-rw----. 1 mysql mysql 106 2013-02-27 06:05 mysql-bin.000017
```

```
-rw-rw-r--. 1 mysql mysql 544 2013-02-27 06:05 mysql-bin.index
```

-생성되는 binary log 최대 크기. (크기를 넘어가면 새로운 번호의 파일이 생성됩니다) #max_binlog_size=100M

-Binary Log 포맷은 반드시 ROW 이어야 합니다.

```
#binlog_format=ROW
```

반드시 위 설정을 my.conf 파일의 [mysqld] 섹션에 넣어야 합니다.

6.2 Crontab 설정 추가

가장 최근의 7 일치만 보관하기 위해 crontab -e 로 다음설정을 추가합니다 (시간은 본 문서에서는 임의로 2 분간으로 하였으며, 상황에 맞게 조정합니다)

```
*/2 * * * * /usr/bin/find /opt/atlassian/confluence/logs/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /opt/atlassian/jira/logs/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /opt/atlassian/crowd/apache-tomcat/logs/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /var/atlassian/application-data/fisheye/var/log/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /var/atlassian/application-data/jira/log/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /var/atlassian/application-data/confluence/logs/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

```
*/2 * * * * /usr/bin/find /var/atlassian/application-data/crowd/logs/ -mtime +7 -name "*.log" -print -exec /bin/rm {} \;
```

7 긴급 상황에서 로그 수집

- 기본적으로 활성화 된 로그: Catalina.out, Atalssian Home Directory Log, Access Log, GC Log, MySQL Log

- 긴급 상황에서 받아야 하는 로그: Thread Dump, Support ZIP(가능하다면)

- 기타 권고 로그

```
netstat -an | grep <Listen 포트>
```

```
pmap <pid>
```

