

---

**TESTA, HURWITZ & THIBEAULT, LLP**

---

ATTORNEYS AT LAW

OFFICE (617) 248-7000  
Direct Dial (617) 248-7269125 HIGH STREET  
BOSTON, MASSACHUSETTS 02110-2704FAX (617) 248-7100  
E-Mail: byer@tth.com

October 31, 2003

Mr. Greg Stein  
Chairman  
The Apache Software Foundation  
1901 Munsey Drive  
Forest Hill, MD 21050-2747

Mr. Dirk-Willem van Gulik  
President  
The Apache Software Foundation  
1901 Munsey Drive  
Forest Hill, MD 21050-2747

Mr. Jim Jagielski  
Vice President, Apache Incubator  
The Apache Software Foundation  
1901 Munsey Drive  
Forest Hill, MD 21050-2747

Dear Gentlemen,

This firm represents JBoss Group, LLC ("JBoss"), which is the provider of a leading J2EE-based application server, namely JBoss (the "JBoss program"). In addition to administering the development of the JBoss software project, JBoss Group offers development, support, and training services in connection with the JBoss program.

We understand that the Apache Software Foundation ("ASF") has for some time been developing a J2EE application server called Geronimo, which will offer functionality similar to that of the JBoss program. This letter follows Larry Rosen's e-mail correspondence to you of July 24, 2003, addressing concerns JBoss had about certain portions of the Geronimo source code. As there are remaining concerns in this regard following Larry Rosen's e-mail, I am writing to provide additional information and to request your further response.

It has come to our attention that portions of the Geronimo program appear to be virtually identical or substantially similar to JBoss program source code. Attached as Exhibits A through C are reproductions of source code demonstrating such similarity. Additionally JBoss has provided us with the following information regarding each Exhibit:

---

TESTA, HURWITZ & THIBEAULT, LLP

---

Apache Software Foundation  
October 31, 2003  
Page 2 of 9

Exhibit A: the two files in this exhibit demonstrate on their face a very high degree of similarity between the two files. Additionally, JBoss believes that the similarity in the name and order of variables employed in these files suggest that the Geronimo file is derived from the JBoss file. Moreover, some of the comments contained in the Geronimo file are in part identical to comments in the JBoss file.

Exhibit B: the two files in this exhibit again show similarity or, in some portions, are identical to the exhibited JBoss file. JBoss believes that certain portions of the similar code, for example the usage of NDC logging, are highly JBoss specific and that their use is therefore evidence of copying JBoss source to develop the exhibited Geronimo file.

Exhibit C: as demonstrated by the file portions displayed in this exhibit, the Geronimo code appears to be nearly identical to the JBoss code.

Further, although not reproduced as an exhibit, we direct your attention to the similarities in the files named `org.apache.geronimo.common.Invocation` (Geronimo) and `org.jboss.invocation.Invocation` (JBoss). Both of these files contain what JBoss believes to be highly JBoss-specific payloads, which are named "AsIs," "Transient," and "Marshalled." JBoss believes that the use of such named payloads in the Geronimo file could only have occurred through copying from the JBoss source code. Further, JBoss believes that the `Invocation` file is central to the architecture of both JBoss and Geronimo. Therefore, any copying evident in this file is of great concern, because it could have impact throughout Geronimo.

We understand that the above-referenced examples do not constitute a complete list of the similarities between the code bases of JBoss and Geronimo. Rather, JBoss believes that they are merely illustrative of similarities that have been discovered to date with a non-exhaustive search. In addition to examples of what appears to be direct copying of JBoss source code in the Geronimo code base, JBoss also has observed a significant number of architectural similarities, similar naming conventions, and other indications that portions of Geronimo may constitute derivative works of JBoss source code.

As you may know, the JBoss program is licensed to users under the Lesser General Public License ("LGPL"). This license allows use of the JBoss source code only to the extent that certain obligations are met, including but not limited to the obligation to license derivative works under the LGPL. Thus, distribution under the Apache Software License of programs which are derived from or incorporate software under the LGPL

---

TESTA, HURWITZ & THIBEAULT, LLP

---


Apache Software Foundation  
October 31, 2003  
Page 3 of 9

would not be permissible and would constitute violation of the LGPL. As such, we are concerned that any Geronimo source code which derives or is copied from JBoss source code would constitute violation of the LGPL if distributed.

We understand that any JBoss source code copied into the Geronimo project may have occurred inadvertently and without the ASF's knowledge. If so, we request that measures be taken immediately to remedy any such examples of copying or in the alternative that other measures be taken to bring your use of JBoss code into compliance with the LGPL. We further ask that you take action to prevent any improper copying from occurring in the future.

I look forward to your reply in this matter. If you have any questions, please do not hesitate to contact me.

Very truly yours,



David J. Byer

cc: Marc Fleury

TESTA, HURWITZ & THIBEAULT, LLP

Apache Software Foundation  
 October 31, 2003  
 Page 4 of 9

**Exhibit A: XLevel**

<pre> JBoss (org.jboss.logging.XLevel): /*****  *  * JBoss: The OpenSource J2EE WebOS  *  *  * Distributable under LGPL license.  *  * See terms of license at gnu.org.  *  *****/  package org.jboss.logging;  import org.apache.log4j.Level;  /**  * Provides custom extension levels for use with the Log4j  * logging framework.  *  * &lt;p&gt;  * Adds a trace level that is below the standard Log4j  * &lt;tt&gt;DEBUG&lt;/tt&gt; level.  *  * &lt;p&gt;  * This is a custom level that is 100 below the {@link  * Level#DEBUG_INT}  * and represents a lower level useful for logging events  * that should only  * be displayed when deep debugging is required.  *  * @see org.apache.log4j.Level  *  * @author &lt;a href="mailto:Scott.Stark@jboss.org"&gt;Scott  * Stark&lt;/a&gt;  * @version \$Revision: 1.3.2.1 \$  */ public class XLevel     extends Level {     /** The integer representation of the level, ({@link     Level#DEBUG_INT} - 100) */     public static final int TRACE_INT = Level.DEBUG_INT -     100;      /** The string name of the trace level. */     public static final String TRACE_STR = "TRACE";      /** The TRACE level object singleton */     public static final XLevel TRACE = new     XLevel(TRACE_INT, TRACE_STR, 7);      /**      * Construct a &lt;tt&gt;XLevel&lt;/tt&gt;.      */     protected XLevel(final int level, final String strLevel, final int     syslogEquiv)     {         super(level, strLevel, syslogEquiv);     } </pre>	<pre> GERONIMO (org.apache.geronimo.core.log.XLevel): -----  *  * The Apache Software License, Version 1.1  *  * =====  *  */ package org.apache.geronimo.kernel.log;  import org.apache.log4j.Level;  /**  * Extension levels for Log4j  *  * @version \$Revision: 1.1 \$ \$Date: 2003/09/08 04:38:34 \$  */  public final class XLevel extends Level {     public static final int TRACE_INT = Level.DEBUG_INT - 1;     private static String TRACE_NAME = "TRACE";      /**      * The Log4j Level Object to use for trace level      * messages      */     public static final XLevel TRACE = new XLevel(TRACE_INT,     TRACE_NAME, 7);      protected XLevel(int level, String name, int syslogEquiv) {         super(level, name, syslogEquiv);     } </pre>
--	---

---

 TESTA, HURWITZ & THIBEAULT, LLP
 

---

Apache Software Foundation  
 October 31, 2003  
 Page 5 of 9

```

////////////////////////////////////
//          Factory Methods          //
////////////////////////////////////

/**
 * Convert an integer passed as argument to a level. If
 * the conversion
 * fails, then this method returns the specified default.
 *
 * @return the Level object for name if one exists,
 *         defaultLevel otherwise.
 */
public static Level toLevel(final String name, final Level
defaultLevel)
{
    if (name == null)
        return defaultLevel;

    String upper = name.toUpperCase();
    if (upper.equals(TRACE_STR)) {
        return TRACE;
    }

    return Level.toLevel(name, defaultLevel);
}

/**
 * Convert an integer passed as argument to a level.
 *
 * @return the Level object for name if one exists
 */
public static Level toLevel(final String name)
{
    return toLevel(name, TRACE);
}

/**
 * Convert an integer passed as argument to a priority.
 * If the conversion
 * fails, then this method returns the specified default.
 *
 * @return the Level object for l if one exists,
 *         defaultLevel otherwise.
 */
public static Level toLevel(int l)
{
    return toLevel(l, TRACE);
}

/**
 * Convert an integer passed as argument to a level. If
 * the conversion
 * fails, then this method returns the specified default.
 *
 * @return the Level object for l if one exists,
 *         defaultLevel otherwise.
 */
public static Level toLevel(final int l, final Level
defaultLevel)
{
    Level p;
    if (l == TRACE_INT)
        p = TRACE;

```

```

/**
 * Convert the String argument to a level. If the
 * conversion
 * fails then this method returns {@link #TRACE}.
 */
public static Level toLevel(String name) {
    return toLevel(name, XLevel.TRACE);
}

/**
 * Convert the String argument to a level. If the
 * conversion
 * fails, return the level specified by the second
 * argument,
 * i.e. defaultValue.
 */
public static Level toLevel(String name, Level defaultValue) {
    if (name == null) {
        return defaultValue;
    }
    if (name.toUpperCase().equals(TRACE_NAME)) {
        return XLevel.TRACE;
    }
    return Level.toLevel(name, defaultValue);
}

/**
 * Convert an integer passed as argument to a level. If
 * the
 * conversion fails, then this method returns {@link
 * #DEBUG}.
 */
public static Level toLevel(int level) throws
IllegalArgumentException {
    if (level == TRACE_INT) {

```

TESTA, HURWITZ & THIBEAULT, LLP

Apache Software Foundation  
October 31, 2003  
Page 6 of 9

<pre>else   p = Level.toLevel(l); return p; }</pre>	<pre>return XLevel.TRACE; } else {   return Level.toLevel(level); }</pre>
---	---

TESTA, HURWITZ & THIBEAULT, LLP

Apache Software Foundation
October 31, 2003
Page 7 of 9

Exhibit B: PatternParser

Table with 2 columns: JBOSS (org.jboss.logging.layout.PatternParserEx): and GERONIMO (org.apache.geronimo.core.log.PatternParser):. Each column contains Java code snippets with comments and class definitions.

---

TESTA, HURWITZ & THIBEAULT, LLP

Apache Software Foundation  
October 31, 2003  
Page 8 of 9

<pre>protected void finalizeConverter(char c) {     PatternConverter pc = null;     switch©     {         case 'z':             pc = new ThreadNDCCConverter(formattingInfo);             currentLiteral.setLength(0);             break;         case 'Z':             String key = extractOption();             pc = new ThreadMDCCConverter(formattingInfo, key);             currentLiteral.setLength(0);             break;         default:             super.finalizeConverter©;             return;     }     addConverter(pc); } }</pre>	<pre>protected void finalizeConverter(char c) {     PatternConverter pc = null;     switch © {         case 'a':             String key = extractOption();             pc = new NamedNDCCConverter(formattingInfo, key);             currentLiteral.setLength(0);             break;         default:             super.finalizeConverter©;             return;     }     addConverter(pc); } }</pre>
---	---



TESTA, HURWITZ & THIBEAULT, LLP

Apache Software Foundation  
October 31, 2003  
Page 9 of 9

**Exhibit C: InvocationType**

<p>JBOSS (org.jboss.invocation.InvocationType):</p> <p style="text-align: center;">[***]</p> <pre>public static final InvocationType REMOTE =   new InvocationType("REMOTE", 0); public static final InvocationType LOCAL =   new InvocationType("LOCAL", 1); public static final InvocationType HOME =   new InvocationType("HOME", 2); public static final InvocationType LOCALHOME =   new InvocationType("LOCALHOME", 3);</pre> <p style="text-align: center;">[***]</p>	<p>GERONIMO (org.apache.geronimo.common.InvocationType):</p> <p style="text-align: center;">[***]</p> <pre>public static final InvocationType REMOTE = new InvocationType("REMOTE", 0, false, false); public static final InvocationType HOME = new InvocationType("HOME", 1, false, true); public static final InvocationType LOCAL = new InvocationType("LOCAL", 2, true, false); public static final InvocationType LOCALHOME = new InvocationType("LOCALHOME", 3, false, false);</pre> <p style="text-align: center;">[***]</p>
--	---