

Re: strange behavior of the JVM or compiler...

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-07/msg00696.html>

- *From:* Twisted <twisted0n3@xxxxxxxxxx>
 - *Date:* Sun, 08 Jul 2007 19:34:14 -0000
-

On Jul 8, 12:33 pm, Roedy Green <see_webs...@xxxxxxxxxxxxxxxxxxxxxxxx> wrote:

Second, static finals get turned into literals (constants) even when they are referred outside the class. If you modify them, you must recompile the universe to make sure the new value gets propagated.

Efficient though this is, it seems to mean that for "constants" that you expect might change in a future version of a library you should provide accessor methods and not a public static final field. Otherwise, drop-in replacement of the older version with the newer version in a deployed app's classpath might fail in bizarre ways. To make sure newer versions of a library are binary-compatible with apps that use the library, and not just source-compatible, requires not changing the values of public static final fields.

So you'd want to use

```
public static final int MAX_FACTORIAL = 12; // 12! is less than
// Integer.MAX_VALUE; 13! is greater; and this will never change.
```

```
public static int getMaxTreeDepth () {
return 5;
// In later versions this might be increased to 6 or even 7.
}
```