

Re: Column numbers in stack trace – enhancement request

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2007-01/msg02092.html>

- *From:* "Chris Uppal" <chris.uppal@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Fri, 19 Jan 2007 11:55:47 -0000
-

Sasi wrote:

How about including in NullPointerException the compile time type of the subexpression that was null?

In addition, if the trace contains what method call was attempted on the null variable, it might help differentiate many of the cases where the return types of two methods in the same line are identical. JRE should be able to provide the method name, shouldn't it?

The JVM could certainly record what the failed operation had been (it doesn't have to be a method call, it could be a field access or an index into an array for instance). But remember that NPEs can be thrown explicitly too -- that doesn't mean that the information couldn't be recorded when it was generated by the JVM, but that wouldn't always be the case.

I won't be writing any more code with multiple method calls in the same line. At the same time, I have never seen any warning against this practice in the Java books I've read or even Sun's Java coding conventions. Are there any such warnings in some java book or coding standard?

I've never seen a warning against it on those grounds. One partial reason might be that it's only recently (since 1.5 as I recall) that Sun's compilers have routinely included line number information in .class files at all (it has always been available as an option, but was not the default) and without that you don't get line number information in stack traces.

-- chris

Re: Column numbers in stack trace – enhancement request