

Re: Detecting multiple class loaders

Source: <http://coding.derkeiler.com/Archive/Java/comp.lang.java.programmer/2008-03/msg01699.html>

- *From:* Silvio Bierman <sbierman@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 20 Mar 2008 09:12:48 +0100
-

Chris wrote:

Is there any way to detect when a class has been loaded by two different class loaders?

Our app has a class that reads and writes a file on disk. The file gets corrupt if more than one process gets at it at once. We can detect multiple processes by locking the file on disk and throwing an error; no problem. We can control access by multiple threads using synchronization. So far, so good.

The problem is Websphere. When this class gets used in a webapp, Websphere, like most other app servers, creates a classloader just for that webapp. But it does more than that; it creates multiple classloaders and they all load the class independently. And we get corruption.

Other than fix Websphere (and we're working on it), is there another way, in general, we can use to ensure that only one classloader within a JVM loads a given class?

Are you by any chance using the broken singleton pattern based on a static instance member? If so your only option is to use an external resource to lock on as a workaround. As others have pointed out sockets are among the obvious candidates.

Silvio Bierman

.