**SUMMARY**

A concurrency bug in Eclipse 3.2 Platform Debug.

**DETAILS**

Some details can also be found at: <https://bugs.eclipse.org/bugs/show_bug.cgi?id=144560>

This bug is due to a data race.

Simple race condition. The input read job runs in a worker thread. When disposeStreams is called, the input read job is still running and may be reading data. I think it is safe to add a simple null check.

|  |  |
| --- | --- |
| Thread1 ProcessConsole.java () | Thread2 (ProcessConsole.java) |
| protected IStatus run(IProgressMonitor monitor) {  try {  byte[] b = new byte[1024];  int read = 0;  while (read >= 0) {  read = fInput.read(b); | private synchronized void disposeStreams() {  for (Iterator i = fStreamListeners.iterator(); i.hasNext();) {  StreamListener listener = (StreamListener) i.next();  listener.dispose();  }  fFileOutputStream = null;  fInput = null;  } |
| 0-lock | 1-lock |