**SUMMARY**

A concurrency bug in Eclipse 3.4 Platform Debug.

**DETAILS**

Some details can also be found at: <https://bugs.eclipse.org/bugs/show_bug.cgi?id=210023>

This bug is due to a data race.

disposeAllProxies() method called when the viewer input changes. In this case the DefaultWatchExpressionModelProxy.dispose() is called before DefaultWatchExpressionModelProxy.install() which probably breaks an implied API contract.

|  |  |
| --- | --- |
| Thread1 () | Thread2 () |
| public void installed(final Viewer viewer) {  super.installed(viewer);  UIJob job = new UIJob("install watch expression model proxy") { //$NON-NLS-1$  public IStatus runInUIThread(IProgressMonitor monitor) {  if (!isDisposed()) {  IWorkbenchWindow[] workbenchWindows = PlatformUI.getWorkbench().getWorkbenchWindows();  for (int i = 0; i < workbenchWindows.length; i++) {  IWorkbenchWindow window = workbenchWindows[i];  if (viewer.getControl().getShell().equals(window.getShell())) {  fWindow = window;  break;  }  }  if (fWindow == null) {  fWindow = DebugUIPlugin.getActiveWorkbenchWindow();  }  IDebugContextService contextService = DebugUITools.getDebugContextManager().getContextService(fWindow);  contextService.addDebugContextListener(DefaultWatchExpressionModelProxy.this);  ISelection activeContext = contextService.getActiveContext();  if (activeContext != null) {  contextActivated(activeContext);  }  }  return Status.OK\_STATUS;  }    };  job.setSystem(true);  job.schedule();  } | public synchronized void dispose() {  super.dispose();  DebugUITools.getDebugContextManager().getContextService(fWindow).removeDebugContextListener(this);  fWindow = null;  } |
| 0-lock | 1-lock |