**SUMMARY**

A concurrency bug in Eclipse 3.6.1. Platform Resources.

**DETAILS**

Some details can also be found at: <https://bugs.eclipse.org/bugs/show_bug.cgi?id=307587>

This bug is due to a data race.

The class NatureManager is NOT threadsafe at all, causing incorrect values to be return for isNatureEnabled().

Scenerio: 2 (or more) job are trying are simultaneously asking if a project has a nature (isNatureEnabled()) soon after the project was created for the first time. This leads to a race condition in getEnabledNatures().

The real fix would make it so that all (not just isNatureEnabled) methods in that class should be thread-safe.

|  |  |
| --- | --- |
| Thread1 () | Thread2 () |
| public boolean isNatureEnabled(Project project, String id) {  String[] enabled = getEnabledNatures(project);  for (int i = 0; i < enabled.length; i++) {  if (enabled[i].equals(id))  return true;  }  return false;  } | public boolean isNatureEnabled(Project project, String id) {  String[] enabled = getEnabledNatures(project);  for (int i = 0; i < enabled.length; i++) {  if (enabled[i].equals(id))  return true;  }  return false;  } |
| protected String[] getEnabledNatures(Project project) {  String[] enabled;  if (natureEnablements != null) {  enabled = (String[]) natureEnablements.get(project);  if (enabled != null)  return enabled;  }  enabled = computeNatureEnablements(project);  setEnabledNatures(project, enabled);  return enabled;  } | protected void setEnabledNatures(IProject project, String[] enablements) {  if (natureEnablements == null)  natureEnablements = Collections.synchronizedMap(new HashMap(20));  natureEnablements.put(project, enablements);  } |
| 0-lock | 0-lock |