

Confluence user macros using class.forName() break in Confluence 7.1+

Confluence [user macros](#) put power in the hands of administrators to create custom macros utilizing the full Confluence API. They are one of Confluence Server's best features, analogous to Jira ScriptRunner 'Script Fields'. I've written [a number of them](#) over time.

However there is a strong trend in tech for Big Tech to take power away from end users in the name of security. Thus, if you upgrade to Confluence 7.9+, you may find your more interesting User Macros have broken:



Error rendering macro 'param'
Error occurred rendering template content

A common trick in writing User Macros is to get a class via reflection with a construct like:

```
#set($df = $action.getDateFormatter())
#set($now = $action.getClass().forName("java.util.Date").newInstance())
```

In Confluence 7.x this is no longer permitted, in the name of security. See forum post:

<https://community.developer.atlassian.com/t/problem-to-get-about-me-information-from-user-details-after-confluence-7-update/35258/2>

The fix as described works, namely to remove the restrictions on `java.lang.reflect`, `java.lang.Class` and `java.lang.ClassLoader`:

```
diff --git a/confluence/WEB-INF/classes/velocity.properties b/confluence/WEB-INF/classes/velocity.properties
--- a/confluence/WEB-INF/classes/velocity.properties
+++ b/confluence/WEB-INF/classes/velocity.properties
@@ -163,7 +163,7 @@ velocimacro.max.depth=-1
# https://extranet.atlassian.com/display/CSPF/Restrict+packages+and+classes+usage+from+velocity+files
# -----

-introspector.restrict.packages = java.lang.reflect,\
+introspector.restrict.packages = \
com.atlassian.cache,\
com.atlassian.confluence.util.http,\
com.atlassian.failurecache,\
@@ -230,8 +230,7 @@ org.springframework.util.concurrent,\
org.quartz,\
oshi

-introspector.restrict.classes = java.lang.Class,\
-java.lang.ClassLoader,\
+introspector.restrict.classes = \
java.lang.Compiler,\
java.lang.InheritableThreadLocal,\
java.lang.Package,\
```