

# Running Bioclipse Plugin Unit tests: solving the XPCOM error

Egon Willighagen 

Published August 5, 2009

## Citation

Willighagen, E. (2009). Running Bioclipse Plugin Unit tests: solving the XPCOM error. In *chem-bla-ics*. chem-bla-ics. <https://doi.org/10.59350/b3t7r-0ge09>

## Keywords

Bioclipse, Ubuntu, Eclipse

## Abstract

Sometimes you can feel so stupid. For example, when the answer is right on front of you, but only after many hours you realize the right question belonging to that answer.

## Copyright

Copyright © Egon Willighagen 2009. Distributed under the terms of the [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## chem-bla-ics

Sometimes you can feel so stupid. For example, when the answer is right on front of you, but only after many hours you realize the right question belonging to that answer. For example, take [this answer](#):

```
add the line: -Dorg.eclipse.swt.browser.XULRunnerPath=/usr/lib/xulrunner
```

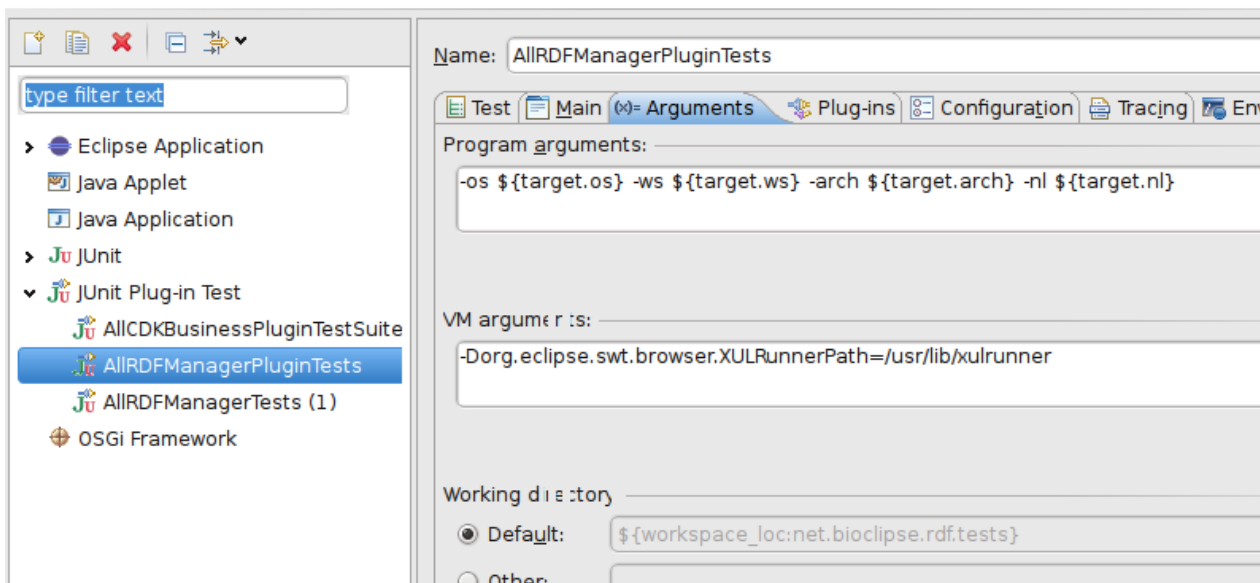
This is the problem I was trying to solve: I'm running 64bit Ubuntu Jaunty with Eclipse 3.4.2 for [Bioclipse](#) development. The answer above is the correct answer. So, I added the line. To the `$HOME/eclipse.ini` and to the eclipse command line to start the program. But I still good not run Bioclipse plugin unit tests; I kept getting that stupid error:

```
org.eclipse.swt.SWTError: XPCOM error -2147467262
at org.eclipse.swt.browser.Mozilla.error(Mozilla.java :1638)
at org.eclipse.swt.browser.Mozilla.setText(Mozilla.java :1861)
```

In retrospect, I was sort of asking the wrong question. I should have asked myself not why I got that XPCOM error even though I was using the solution, but why running the unit tests was not affected by that solution. Realizing that, it became so obvious: the plugin unit testing was using a clean environment, not based on the Eclipse environment I was working in; therefore, adding that line to my Eclipse environment did not help. Instead, I only had to that line to the Run Configuration of my plugin unit tests too:

### Create, manage, and run configurations

Create a configuration that will launch a JUnit plug-in test.



Surely, there are aspects to this which helped me overlook this solution. For example, I had installed Eclipse freshly yesterday, and then the it worked fine. Only after installing some EMF and GEF features, it stopped working again. Bitten by the correlation/causation pattern :(