

Linux Workshop 2004b: NumericalGecko Eclipse Plug-in – Testing and Conclusions



Part 1: Planned Tests

In testing the Numerical Gecko (NG) workshop project, we need to consider more than simply whether the application works and what bugs, if any, remain. We will also consider external aspects, such as issues concerning the distribution of the project in the open source community, and compare the final product to the initial designs. Specifically, we need to test:

- **Program Functionality:** Standard QA tests, looking for bugs and other issues with the application.
- **Platform Independence:** One of the features of NG is the fact that it should be platform-independent, or more specifically – should work properly on any system that can run Eclipse 3.0 and Java. Testing on any possible configuration is impossible, so we'll test on a couple of versions of Windows, the version of Linux that is installed in TAU and a few iterations of Eclipse 3.0 (pre-release and official release versions).
- **Distribution:** Once we upload the finished product to sourceforge and publicize it on a couple of Eclipse plugin websites, we need to make sure the files download and install correctly on a few different machines.
- **Comparison with initial designs:** See what features made it into the finished version, what features didn't and why.

Part 2: Test Results

We ran the aforementioned tests before and after releasing the finished product. Following is an account of our findings and various issues we dealt with:

- **Program Functionality:** Testing proved somewhat problematic, as due to limitations within the Eclipse IDE not all of the features could be tested in the debugger. To test these features, we had to install a production version of the plug-in, run it and check for errors. Generally, there are no major issues left open, though some of the conversion categories have been left unsupported (since they need elements from the original Numerical Chameleon that we had to drop in the development process).
- **Platform Independence:** The plug-in has been tested and found to be working properly on Windows 2000, Windows XP and Red-Hat Linux, with several different versions of Eclipse 3.0.
- **Distribution:** The [project home page](#) has been linked to on two major Eclipse plug-in sites ([eclipse-plugins](#) and [eclipseplugincentral](#)), and has been downloaded hundreds of times in a few days. We tested the downloaded files, and tried to install them on clean Eclipse environments (both on Windows and Linux). The tests went well and the plug-in worked correctly in each of them.

- **Comparison with initial designs:** Generally, we accomplished what we set out to do in the initial design. Features that we had to drop were: Using NG as input to Eclipse's refactoring tools – we had assumed that the refactoring tools came equipped with extension points that would enable us to use their mechanism, and that was not the case. Writing new refactoring tools ourselves would have been a huge undertaking, and could be considered a whole new project. Improved exchange rates update – while we did add support for web services when we added the translation conversion category, the exchange rates mechanism proved to be much more complicated than initially anticipated since it required major changes to the original code. Improved conversion UI – we had initially planned to allow having a list of items as input for NG's conversion engine, but had since discovered that such an option has been added to NumericalChameleon selectively (to some of the categories), and we felt we had nothing more to contribute to this. We did accomplish everything else we anticipated in the initial design, and have also contributed new features to the original NumericalChameleon, such as the Hebrew interface translation. In addition, we have offered NumericalChameleon's author to contribute our new conversion category – Translation, which uses web services, and he may add this feature in a future version.