Numerical Gecko Eclipse Plug-in: Guidelines for Future Development

General

The Numerical Gecko Eclipse Plug-in (NG) is a fully working implementation of a pre-existing open source Java application, called NumericalChameleon (NC), designed as an Eclipse IDE view and fully integrated into the Eclipse development environment. While staying true to the source application, NG omits some of the features found in NC, changes some of the others, and expands on some of the rest. As this is an open source project, built to function in a constantly updating and dynamic environment, it, too, may be modified in the future and taken to new and exciting directions. This document will highlight some of the possible expansions and/or changes that can be made to this project.

Before attempting to modify NG, the developer should familiarize himself with the Eclipse IDE and Plug-in Development Environment (PDE), with NG and its specification documents (found at the project homepage) and, preferably, with the parent application, NC.

Additions and Changes to the Data Model

As detailed in the specification document, the data model includes the core conversion engine, the original conversion objects (called clusters) and new conversion modules, and is essentially NC's core, without the GUI elements, and wrapped in a new API. Changes to this model may include re-enabling some of the dropped features from NC or adding new conversion modules.

- **Re-enabling dropped features:** When developing NG, we had made an effort not to change any of the original NC code or resources. While we were not able to use NC's core binaries, due to technical difficulties (open issues with Eclipse's PDE and external jars), we were able to limit our changes to the original sources to a very small degree. This, combined with the fact that we were not able to use NC's main class, lead to some conversion categories (like international time zones) not being supported in NG, and to some of the new feature names in NG (such as the translation conversion category) not being localized along with the rest of the interface. Any developer looking to add some of these features, should study the jonelo.NumericalChameleon.Main class, its use of the ISO-8859-1 resource bundle, and the use of other resource bundles under the "data" folder (which can be found inside the plug-in's main jar file – NCPlugin.jar). The best solution, of course, will be the modification of these classes by their original author.
- **Adding new conversion modules:** NC used objects called clusters to handle various forms of conversions. All of these are inherited from a common parent class – jonelo.NumericalChameleon.Clusters.ClusterObject.

When we came around to adding a new conversion module, Translation, we followed the same guidelines and created a new cluster object – org.numerical_gecko.clusters.ClusterTranslation (using a new package, so as not to disrupt NC's original directory structure). Developers wishing to add new conversion modules should study this class, as well as its use in org.numerical_gecko.Controller and the cluster generation API in org.numerical_gecko.ClusterGenerator. The translation conversion module is also a fine example of using web services within NG and could be used for further study in that area.

Additions and Changes to the Data View

As detailed in the specification document, the data view includes NG's user interface (written in SWT to comply with Eclipse's requirements) and the IDE integration. Both of these components may be extended or modified, using existing Eclipse extension points, adding new user preferences, changing the look and feel of the plug-in and more.

- **Modifying the GUI:** The entire GUI creation for NG is contained within org.numerical_gecko.views.SampleView, including some variations on the basic view (such as the "Thin" skin, which radically alters the appearance of the plugin). Developers wishing to modify the appearance of the plug-in or to add new features (including appropriate GUI elements), should study this class and modify it as needed.
- **New or changed user preferences:** NG adds a new preference page to the Eclipse workspace, utilizing an extension point found in org.eclipse.ui.preferencePages. The user preferences are stored in the plug-in's built-in preference store, and are initialized and accessed (mainly) through the plug-in's main class, org.numerical_gecko.NumericalGecko. The preferences are then used, as needed, throughout the plug-in's classes (org.numerical_gecko.Controller and org.numerical_gecko.views.SampleView, for example). The preference page itself is created in org.numerical_gecko.preferences.GeckoPreferencePage. Any developer who wants to alter or add to the use of preferences may want to study these classes and add to them.
- **Adding to the IDE integration:** NG adds a new command to the context menu of Eclipse's text editors, which sends the highlighted text to the NG's source field. The command was added through the extension point in org.eclipse.ui.popupMenus and is contained in the class org.numerical_gecko.actions.SendToConvertor. Another command that sends converted values back to the active editor can be found in org.numerical_gecko.views.SampleView. These are basic examples of IDE integration. Future developers may wish to add similar commands or go in different directions, such as writing new refactoring tools that use NG ('search-convert-replace' for example).

Other changes

Changes to NG are limited only by the developer's imagination and skills, and are encouraged. However, before adding new features, the developers should consider whether these apply merely to the NG plug-in or whether they could be applied to the parent application, NC. If so, it is strongly recommended to contribute to NC

first, and then use the updated NC as the basis to any modified version of NG. For example, when we first set out to convert NC to NG, we contributed a Hebrew translation to the NC's interface, and then used the newly updated version of NC, including that new interface, as our starting point. That way, both NC's core user base and our new user base could benefit from the new feature, and enjoy the magic of open source.