



## GF Eclipse Plugin: an IDE for grammar development in GF

**John J. Camilleri, Krasimir Angelov**  
**john.j.camilleri, krasimir@chalmers.se**  
**University of Gothenburg, Sweden**  
**<http://www.grammaticalframework.org/eclipse/>**

### Description

*The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. FP7-ICT-247914.*

This work introduces an Integrated Development Environment (IDE) for developing grammars using the Grammatical Framework (GF). GF is a functional programming language for writing grammars targeting multiple parallel languages simultaneously. Typical application areas of the framework are machine translation in limited domains and multilingual natural language generation from some formal representation.

The GF IDE is built on top of the Eclipse Platform and aims to provide a modern set of development tools which replace the traditional text editor and console-based approach. In doing so, we hope to make the task of grammar-writing more efficient, reduce the barrier to entry to GF and encourage new users and uses of the framework.

Apart from the standard features made available by the Eclipse platform—including multiple repositionable editor panels, syntax highlighting, code folding, automatic formatting and version control system integration—the GF Eclipse plugin also provides:

1. Immediate notification of **syntax errors** and helpful **semantic warnings** before ever reaching compilation.
2. **Module outlining**, which provides a summary view of every top-level judgement in a GF module, annotated with type information, and allowing quick navigation within source files.
3. **Identifier resolution** for both local and inter-module cross-references, allowing for instant notification of linking errors and jumping to the definition of a referenced identifier.
4. **Inline contextual documentation** for linked identifiers, including type information and listing of other overloaded forms or alternatives.
5. **Wizards** and **code generation** tools for creating new template modules when writing application grammars.
6. **Launch configurations**, allowing developers to set up multiple GF compilation configurations and run-time scripts, and run them directly from the IDE with a single click.
7. **Treebank test management**, providing a graphical interface for testing the grammar's linearisation and parsing performance against a predefined gold standard.