

# An IDE for the Grammatical Framework

Third International Workshop on Free/Open-Source Rule-Based  
Machine Translation (FreeRBMT 2012)  
Gothenburg, Sweden

John J. Camilleri

Chalmers | University of Gothenburg  
*john.j.camilleri@chalmers.se*

14 June 2012

*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.*

What is GF?

- A grammar formalism
- A functional language for grammars
- A software system (compiler/runtime)
- A software library for syntax and morphology

How do we write GF grammars?

- Text editor (editor modes for Emacs, Gedit, Geany)
- GF interpreter in separate console

How do we browse libraries?

- Online documentation
- Web-based or grammar-based tools
- `grep` is always fun

How do we test grammars?

- Custom scripts, Makefiles
- We all love a good `diff`

### Why use an IDE?

- Unified development experience
- Collect tools together
- Avoid context switching
- Not all developers love command lines
- GF's uptake in commercial settings

# The GF Eclipse Plugin (GFEP)

The screenshot displays the Eclipse IDE with the GFEP plugin installed. The interface is divided into several panes:

- Project Explorer:** Shows a project named 'Hello' containing files like 'AbsCat.gf', 'HelloAbs.gf', and 'ResEng.gf'.
- Code Editors:** Three editors are open, showing code in the GF (Gallium) language:
  - AbsCat.gf:** Defines an abstract type `HelloAbs` as `AbsCat` with parameters `Greeting` and `Recipient`. It includes flags for `startcat`, a `cat` function for farewells, and a `fun` function for greetings.
  - ResEng.gf:** Defines a `resource` `ResEng` with a `param` for `Gender` (Masc | Fem) and a `Case2` (Nom | Acc2). It includes an `oper` function for `superate`.
  - HelloEng.gf:** Defines a `concrete` `HelloEng` of `HelloAbs` using `ResEng`. It includes flags for `coding`, a `lincat` function for greetings and farewells, a `lin` function for greetings and farewells, and a `World` function.
- Outline:** Shows the structure of the `HelloEng` type, including `Extends` (ResEng), `flags` (coding), `lincat` (Greeting, Farewell, Recipient), and `lin` (Hello, Goodbye, World).
- Console:** Displays the GFEP Log with build messages for `/Hello/ResEng.gf`, `/Hello/HelloEng.gf`, `/Hello/HelloAbs.gf`, and `/Hello/AbsCat.gf`.

- Leverage existing IDE features
- Syntax errors & semantic warnings
- Wizards, code snippets
- Code formatting
- In-line documentation
- Context-sensitive suggestions
- Cross-reference resolution
- Navigate external libraries
- Run code and test suites directly from IDE

## Approach

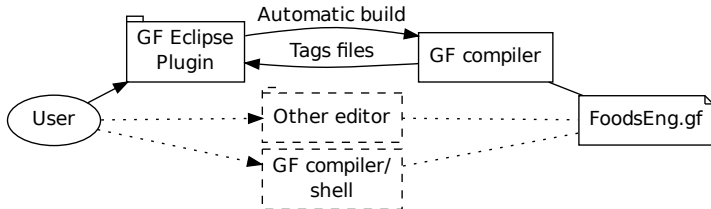
- Layered *on top* of GF, not replacing it
- Requires GF already installed on system
- Eclipse **project** concept
  - Regular GF files + some metadata files
- Eclipse **builder** concept
  - Compiles your code as you write it

## Target audience

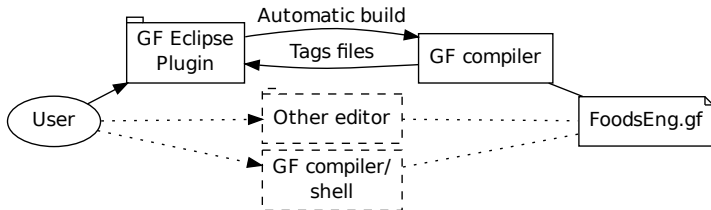
- All levels of GF development
- People who like IDEs

# GF Eclipse Plugin

Delegation to GF







GFEP natively does

- syntax checking
- semantic warnings

GFEP relies on GF for

- scoping information
- type checking

Tags files enable communication of scope information from GF back to Eclipse (though not exclusively).

Tags generation

- Dump the entire scope for any given module
- Requires only binaries (.gfo files)

Run command

```
gf --tags FoodsEng.gf
```

## Command output

```
- compiling /home/john/Foods/Foods.gf...
write file Foods.gf-tags
write file Common.gf-tags
write file Cat.gf-tags
write file Structural.gf-tags
write file Noun.gf-tags
write file Verb.gf-tags
write file Adjective.gf-tags
write file Adverb.gf-tags
write file Numeral.gf-tags
write file Sentence.gf-tags
write file Question.gf-tags
...
```

## Tags file contents

```
mkN3 oper-def .../ParadigmsEng.gf:406
mkN3 oper-type .../ParadigmsEng.gf:118 {s:Number => Case => Str;
mkPN overload-def .../ParadigmsEng.gf:390-393
mkPN overload-type .../ParadigmsEng.gf:390-393 Str -> {s:Case =>
mkPN overload-type .../ParadigmsEng.gf:390-393 {s:Number => Case
mkNoun indir ResEng R ResEng.gf-tags
...
```

GFEP uses this for:

- resolving cross-references
- in-line documentation
- library browser tool

## Time for a demo!



<http://xkcd.com/376/>

### Example

- 1 Syntax errors, other warnings
- 2 Outline view
- 3 Auto-complete
- 4 Unresolved names
- 5 Automatic building
- 6 Jump to definition
- 7 External libraries

## Treebank testing

- ① Generate list of trees
- ② Compile grammar and linearize trees
- ③ Manually correct and save as *Gold Standard*
- ④ For each change, repeat (2) and compare against (3)

We can now do this directly in the plugin!

- Also works in opposite direction (parsing)

## Example

Let's add a new language

- 1 Clone from English to Swedish
- 2 Change some strings
- 3 Create gold standard
- 4 Test against it, iterate

### GFEP **does**:

- leverage useful IDE features
- give you errors and warnings as you type
- help you navigate local and external cross-references
- ease the development-test cycle



### GFEP **does**:

- leverage useful IDE features
- give you errors and warnings as you type
- help you navigate local and external cross-references
- ease the development-test cycle

### GFEP **doesn't**:

- type-check its suggestions
- write your grammars for you
- expose any models, bindings or APIs

## How to evaluate?

- Adoption from two private companies (MOLTO consortium members)
- Eclipse experts, relatively new to GF
- Bug reports & user feedback
- No objective measure of productivity impact

## Great!

Please try it out and give us your feedback.

*Unfortunately I have to miss the tutorial day.*

## Links

Web [www.grammaticalframework.org/eclipse](http://www.grammaticalframework.org/eclipse)

Source [github.com/GrammaticalFramework/gf-eclipse-plugin](https://github.com/GrammaticalFramework/gf-eclipse-plugin)

Bug tracker [github.com/GrammaticalFramework/gf-eclipse-plugin/  
issues](https://github.com/GrammaticalFramework/gf-eclipse-plugin/issues)