Lisaac mode for Eclipse IDE

Damien Bouvarel

October 16, 2009

1 Installation

Important Note :
The Lisaac plugin is available in the website repository (git.debian.org/git/lisaac/eclisaac.git).
The plugin sources are maintained in the eclisaac repository (git.debian.org/git/lisaac/eclisaac.git).

- Install lastest Lisaac Compiler.
- Install latest Eclipse distribution ($\geq$ 3.4, *see http://www.eclipse.org/downloads).*
- Copy the org.lisaac.ldt_{version}.jar from eclisaac.git repository to your eclipse plugins/ installation directory.
- Or, Use the Eclipse Update Manager to install the Lisaac plugin. Clone the website.git, add the website/eclipse/update in the manager. The update-site is also hosted on *http://isaacproject.u-strasbg.fr/eclipse/update/*. Select Lisaac feature and click 'Install'.

2 User Guide

The following is assumed:

- You are starting with a new Eclipse installation with default settings.
- You are familiar with the basic Eclipse workbench mechanisms, such as views and perspectives.
- The Lisaac compiler is installed in the environment where the Eclipse workbench is executed.

2.1 Creating your first Lisaac Project

1. Inside Eclipse select the menu item File → *New* → *Project....* to open the New Project wizard.
2. Select Lisaac Project then click Next to start the New Lisaac Project wizard.
3. Enter your project name then click Finish.

Your project is added to your workspace with the following default files:

Resource	Description
make.lip	Default settings for your project
bin/	Output for compiled project
lib/	Contain links to the library files you are using
src/	Default Main prototype

The Lisaac Perspective is designed for working with Lisaac Projects. It consist of an editor area and the following views:

- Navigator
- Outline
- Problems
- History

2.2 Creating a Prototype file

1. In the Navigator view, select the folder where you want to create the prototype. Click on the New Prototype wizard.
2. Enter the prototype name.
3. Write description, choose inheritance, etc...

2.3 Using Content Assist

2.3.1 Auto-Indentation

When the Auto-identation is enabled, three types of events can fire auto-indentation:

- Indent on new line.
- Indent cursor line with 'tab' character.
- Indent whole document with F2 binding.

2.3.2 Code Completion

Code Completion can be activated at any time in the Lisaac editor with the 'Ctrl+space' key binding.
The completion is also auto-activated while typing the '.' character.

The completion proposals can be either: keywords, slots, prototypes, variables or arguments names.

2.3.3 Code Navigation

Lisaac development is made easier with code navigation :

1. Hyperlinks : When the Ctrl key is pressed, Lisaac element become hyperlinks, the mouse click open the editor at the definition point of the hyperlink text.
2. Hovering : Display infopop information about the Lisaac element pointed by the mouse cursor.
3. Outline View : Display the outline information about the current prototype in editor.
   It lists the slots and sections defined in the prototype file.
   The buttons on top of the view allows two types of sorting for the list : appearance order and alphabetical order.
   The selection of an item in the view synchronize the editor around the item definition in Lisaac code.

2.4 Running your program

While typing code or building a project, the Lisaac program is not compiled.
The compilation is explicitely run by the user, therefore semantic errors in projects can only be detected at that time.

1. Select (one of) the main prototype of you project. In the run (respect. debug) menu/button choose *RunAs* → *LisaacApplication* to launch a run (respect. debug) default configuration.
2. Or, edit a run or debug configuration in *Run* → *Run* configuration....
3. Set the Lisaac compiler options (from the project make.lip file).
4. Give the command-line arguments of your program.

When a run configuration is launched :

- if the program is modified or not compiled : The program is compiled with Lisaac Compiler.

- if the program is compiled and not modified : The compiled program is executed.

Note : The debug configuration only add a -no_debug option to the compiler.

2.5 Other Features

2.5.1 Preferences

Syntax highlighting colors and styles are customizable in the Lisaac preference page (menu $Window \rightarrow Preferences \rightarrow Lisaac$).

2.5.2 Slot and Section folding

To shorten the prototype display in the editor, sections and multi-line slots can be folded with the fold/unfold button in the left of editor.

2.5.3 Refactoring (experimental)

The 'refactoring' is code re-formating over a set of source files ('Refactor' menu) :

- Change Project Headers : Re-format the Section Header of all prototypes in the project with the given author, copyright, license or bibliography.
- Rename prototype : Update all references of the old prototype name with the new name. Do not affect comments and strings.

Note : The refactorings should be applied on syntactically correct programs, otherwise some parts of the code might not be updated.

3 Plugin Developer Guide

3.1 Getting Started

- Install latest Eclipse SDK (≥ 3.4, *see http://www.eclipse.org/downloads*).
- Clone the eclisaac.git repository to get the plugin sources.
- Import the 'eclisaac' project (File → Import then select 'Existing Projects into Workspace').
- First Run : open META-INF/MANIFEST.MF and click on the link 'Launch an Eclipse application'.
- Other Run : Green toolbar button Run or $Run \rightarrow RunAs \rightarrow EclipseApplication$.
- Export the plugin project to a suitable jar file. (optional)

3.2 Bug Report and plugin improvement

Mail me at damien.bouvarel@gmail.com, or on the Lisaac devel mailing list.