



Linticator

Linticator gives you immediate feedback on programming style and common programmer mistakes by integrating Gimpel Software's popular PC-lint and FlexeLint static analysis tools into Eclipse CDT.

PC-lint and FlexeLint are powerful tools, but they are not very well integrated into a modern developer's workflow. **Linticator** brings the power of Lint to the Eclipse C/C++ Development Tools by fully integrating them into the IDE. With Linticator, you get continuous feedback on the code you are working on, allowing you to write better code.

• Automatic Lint Configuration

Lint's configuration, like include paths and symbols, is automatically updated from your Eclipse CDT settings, freeing you from keeping them in sync manually.

• Suppress Messages Easily

False positives or unwanted Lint messages can be suppressed directly from Eclipse, without having to learn Lint's inhibition syntax—either locally, per file or per symbol.

• Interactive “Linting” and Information Display

Lint is run after each build or on demand, and its findings are integrated into the editor by annotating the source view with interactive markers, by populating Eclipse's problem view with Lint's issues and by linking these issues with our *Lint Documentation View*.

• Quick-Fix Coding Problems

Linticator provides automatic fixes for a growing number of Lint messages. For example, making a reference-parameter const can be done with two keystrokes or a mouse-click.

Register at <http://linticator.com> if you want to try it, or if you want to be notified about Linticator releases

see more on <http://linticator.com>
to be released 2011

Individual licensing for Linticator TBA.

Enquiries for corporate or site licenses are welcome at ifs@hsr.ch.

Linticator requires a corresponding PC-Lint (Windows) or FlexeLint license per user.

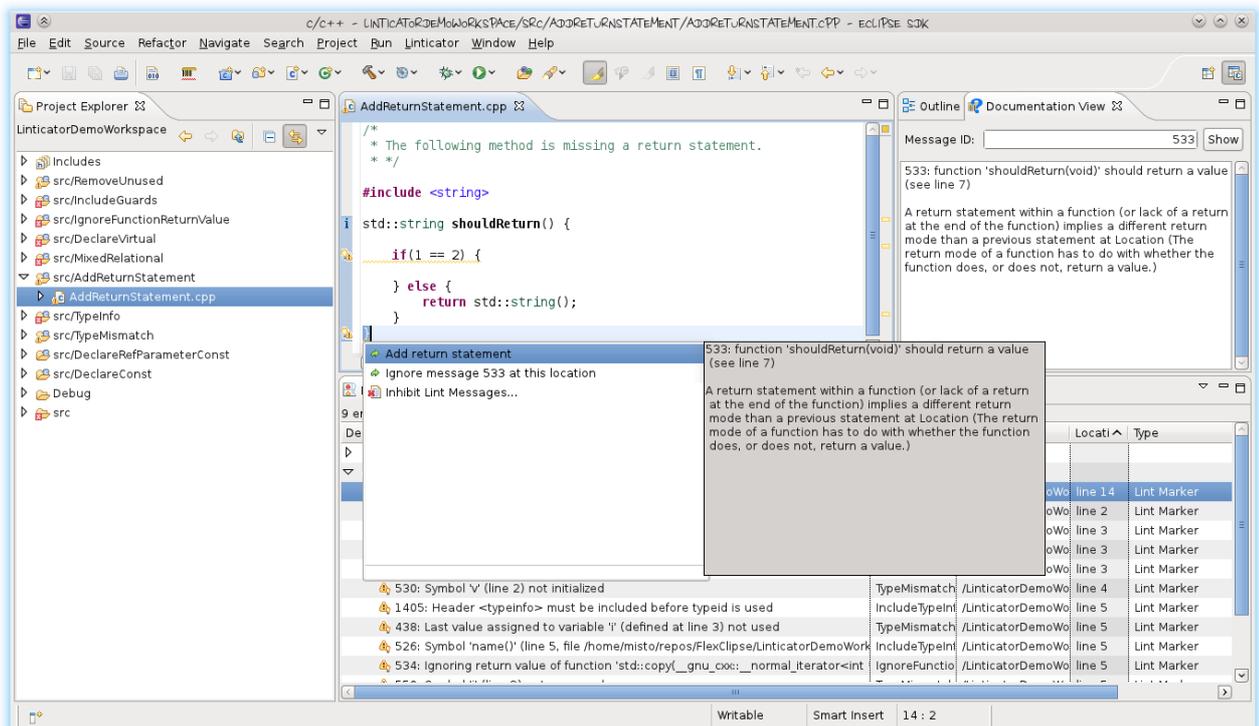
IFS Institute for Software

IFS is an institute of HSR Rapperswil, member of FHO University of Applied Sciences Eastern Switzerland.

In January 2007 IFS became an associate member of the Eclipse Foundation.

The institute manages research and technology transfer projects of four professors and hosts about a dozen assistants and employees.

<http://www.ifsoftware.ch/>





Includator

#include Structure Analysis and Optimization for C++ for Eclipse CDT

The **Includator** plug-in analyzes the dependencies of C++ source file structures generated by #include directives, suggests how the #include structure of a C++ project can be optimized and performs this optimization on behalf of the developer. The aim of these optimizations is to improve code readability and quality, to reduce the duration of builds and the development time of C++ software.

Includator Features

• Find unused includes

Scans a single source file or a whole project for superfluous #include directives and proposes them to be removed. This also covers the removal of #include directives providing declarations that are (transitively) reachable through others.

• Directly include referenced files

Ignores transitively included declarations and proposes to #include used declarations directly, if they are not already included. This provides an "include-what-you-use" code structure.

• Organize includes

Similar to Eclipse Java Development Tool's *Organize imports* feature for Java. Adds missing #include directives and removes superfluous ones.

• Replace includes with forward declarations

Locates #include directives for definitions that can be omitted, when replacing them with corresponding forward declarations instead. This one is useful for minimizing #includes required in header files.

• Static code coverage

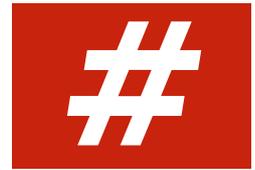
Marks C++ source code as either *used*, *implicitly used* or *unused* by transitively following C++ elements' definitions and usages. This helps to trim declarations and definitions not used from your source code. In contrast to dynamic code coverage, such as provided by our CUTE plug-in (<http://cute-test.com>) it allows to determine required and useless C++ declarations and definitions instead of executed oder not-executed statements.

• Find unused files

Locates single or even entangled header files that are never included in the project's source files.

User Feedback and Participation

Includator is still in beta testing. Register at <http://includator.com> if you are interested in becoming a beta tester, if you just want to try it or if you want to be notified about Includator releases.



see more on
<http://includator.com>
to be released 2011

Individual licensing TBA.

Enquiries for corporate or site licenses are welcome
at ifs@hsr.ch.

