# CDT 6.0 - New and Noteworthy

Note: "New and Noteworthy" for [previous versions](#3dy6vkm) is at the bottom of this file.

See [What's new in CDT 6.0](http://wiki.eclipse.org/CDT/User/NewIn60) on the CDT wiki; may contain other information.

| **Editor** | |
| --- | --- |
| Rename in File | To make rename refactoring work interactively in the editor, position the cursor on an identifier in the editor and hit Ctrl-1 (the hotfix key). Choose "Rename in file." Changing an identifier simultaneously changes all other references to that identifier. |
| Refactoring | "Extract local variable" is added via editor context menu. Select an expression in the editor and select Refactor > Extract local variable from the context menu.      Give the new local variable a name    A local variable is created from the expression and its use is inserted in the original expression. |
| Add Include | Add Include significantly improved.   * CDT can infer the header file of a function or class from its use and automatically insert the appropriate #include directive. For example create a file with a main method and a call to printf. Right click on the call to printf and select Source > Add Include. CDT will infer that printf is defined in the cstdio header and automatically insert the #include directive for that header.   + This feature works much better in CDT 6.0. |
| Outline View | * Inactive code is shown in Outline View. The CDT parser can now detect top-level declarations within inactive code blocks and display them in the outline view. Inactive declarations appear slightly greyed out and the icon has a slash through it. * Grouping of method definitions in the Outline view. The outline view is now capable of displaying class members defined outside the class in the same way it displays members defined inside the class. This feature can be turned on/off in the preferences. |
| Macro Exploration | Improved Macro Exploration control.   * The Macro Expansion hover now has a toolbar at the bottom of the control with back, forward and open declaration buttons. * The title bar of the Macro Expansion hover is now draggable. |
| Block selection mode | Block selection mode now works in the CDT editor. |
| C style block commenting | C style block commenting (Ctrl-Shift-/) improved |
| **Indexing** | |
| Implicit references and overloaded operators | Index support for implicit references and overloaded operators:   * The indexer now picks up references for uses of overloaded operators. * The editor now has full support for overloaded operators including semantic highlighting, mark occurrences and open declaration. * A category has been added to the syntax highlighting options for overloaded operators. * It is now possible to search for references to overloaded operators. * Uses of overloaded operators now show up in the call hierarchy view. * The indexer is picking up more information to support this feature, therefore indexing time may be longer and the size of the index file will be larger. For this reason the indexer options page provides the option to turn off the collection of implicit references. |
| System Includes | * Heuristics to pick up otherwise unresolved include files from the project In older versions of CDT the user was forced to manually set up all include paths that were not discovered by the build system. Now CDT is capable of automatically finding any header file that is located within the project without manual setup. (See [Bug 213562](https://bugs.eclipse.org/bugs/show_bug.cgi?id=213562)) |
| Indexer Accuracy | * Rework of template instantiation There have been significant improvements to the CDT editor for supporting C++ templates. Content assist now works much better within code that makes heavy use of templates. * There have been numerous improvements to the performance and accuracy of the CDT parser and indexer. |
| **Navigation and Search** | |
| Open Declaration | Open Declaration (F3) improved, particularly for unresolved symbols.  Open Declaration is capable of detecting potential matches.    It is now possible to navigate from a delete statement to the destructor method that is implicitly called. |
| **Project and Build** | |
| Converters | * Added a converter to convert a general project to a Managed make project * Fixes to Makefile converter |
| Working Sets | * Complete overhaul of the user interface for Working Set Configurations:   + Updated Manage Working Set Configurations dialog for simpler workflows   + Quick access to configurations via a property page for Working Sets   + Context menu actions on Working Sets in the Project Explorer for activating and building configurations   + For more details, see the [working sets design document](http://wiki.eclipse.org/CDT/designs/workingSets) |
| Make Targets View | * Icons, menu and dialogs have been enhanced. Build action uses hammer icon for consistency with project build icon in editor toolbar. * Drag and Drop of Make Targets in Make Targets View. It is possible to copy/paste, drag files from Project Explorer to the view, or Make Targets to external editor, various options available. |
| Error and Problem management | ErrorParserManager has been significantly simplified and improved. See [bug 264715](https://bugs.eclipse.org/bugs/show_bug.cgi?id=264715) and its subtasks.   * Uses a much more accurate algorithm for associating errors/warnings from build output with files in the workspace. * Other C/C++ projects from the workspace are considered when looking for files. * RSE EFS projects and EFS resources are supported. * Improved performance.     In the **Problems View**, external file locations (outside of the workspace) have been moved to the Location column and associated with "Open External Location" menu.     * Remote Project Support   + New **ICommandLauncher interface** supports overriding how build commands are launched.   + **EFS resources** now supported with Managed Build. * XL C/C++ Compiler Support   + The **XL C/C++ Error Parser** has been modernized. It can handle compiler errors/warnings better and now also xlC linker warnings.   + **XL C/C++ v10.1 option support** in Managed Build   + **XL UPC compiler** support added. * **Headless Build** You can now Import and Build CDT Projects without starting the IDE. |
| **Debug and Launch** | |
| DSF Integration | DSF (Eclipse Debugger Services Framework) integrated. See [DSF features](http://www.eclipse.org/dsdp/dd/development/relnotes/dd_news-1.1.html) for more information. |
| Launch Group | New "Launch Group" launch configuration. Allows to launch several processes at once. |
| **Bugs fixed** | |
| Bugs fixed | Bugs fixed in this release:  [CDT 6.0 bug fixes](http://bugs.eclipse.org/bugs/buglist.cgi?query_format=advanced&short_desc_type=allwordssubstr&short_desc&classification=Tools&product=CDT&target_milestone=6.0&long_desc_type=allwordssubstr&long_desc&bug_file_loc_type=allwordssubstr&bug_file_loc&status_whiteboard_type=allwordssubstr&status_whiteboard&keywords_type=allwords&keywords&bug_status=RESOLVED&bug_status=VERIFIED&bug_status=CLOSED&resolution=FIXED&emailtype1=exact&email1&emailtype2=substring&email2&bugidtype=include&bug_id&votes&chfieldfrom&chfieldto=Now&chfieldvalue&cmdtype=doit&order=Reuse+same+sort+as+last+time&field0-0-0=noop&type0-0-0=noop&value0-0-0) |

[Back to Top](#1t3h5sf)

# CDT 5.0 - New and Noteworthy

CDT 5.0 includes new features in Editor, Call Hierarchy, Refactoring, Indexer, Projects & Build, and Debug.

See [What's New in CDT 5.0](http://wiki.eclipse.org/CDT/User/NewIn50) on the CDT Wiki for more information including screen shots.

Highlights include:

### Editor

* Code and File Templates - user-definable templates can be used in New Class and New Source/Header File wizards.
* New preference page for Code Templates
* Outline view for assembly files
* Code formatter improvements including new whitespace and line wrapping options, Improved GNU coding style compliance
* Content assist improvements
* Doxygen editor support - auto-generation of tags and a pluggable framework for other documentation tools is now available.
* Mark Occurrences - highlights where the selected identifier occurs elsewhere in the editor
* Folding of compound statements
* Macro Expansion hover and exploration tool
* Spell checking available and enabled by default
* Scalability mode for working with very large files.
* Visual Studio key bindings

### Navigation and Search

* Open Declaration for operators, empty macros, element in outline view, etc
* Open Element support for static functions/variables, other improvements
* Search - for Static functions/variables, macros, references of local variables, Search view usability improvements

### Call Hierarchy

* Read/write decorators indicate read/write status for variables
* Polymorphic method calls (virtual methods) support

### Refactoring

* Refactoring infrastructure much improved, models transformation on the AST
* Generate Getters and Setters
* Hide Method
* Implement Method
* Extract Constant
* Extract Function

### Indexer

* Improved accuracy and performance
* Entirely new preprocessor
* Improved infrastructure
* Visual indication of indexer status

### Projects and Build

* Project Properties improved usability
* Make target locations - make targets at the project level are built in the project build directory

### Debug

* Executables view
* Event breakpoints - added gdb catchpoints support

[Back to Top](#1t3h5sf)