# CDT 4.0 - New and Noteworthy

CDT 4.0 includes new features in the editor, new code navigation tools, debugger enhancements, and other improvements.

| **Projects** | |
| --- | --- |
| **New Project Creation** | Now it's easier to create and configure a new CDT project: |
| **Project Templates** | Projects can be created from one of the available templates. Project templates include wizard pages that let you configure basic settings for new project code. |
| **Editor** | |
| **Semantic Highlighting** | Identifiers in the source code can now be colored according to their role, e.g. typedefs, functions, variables, fields, etc. |
| **Inactive Code Highlighting** | Code being excluded by conditional compilation is marked with grey background. |
| **Code Folding Improvements** | Now you can collapse and expand blocks of comments and preprocessor branches in C and C++ files in addition to functions, structures, etc. |
| **Code Formatter** | A configurable code formatter with 4 predefined styles has been added: |
| **Smart Typing** | Typing assistance has been greatly improved. Options include automatic closing of brackets, smart paste and more: |
| **Code Navigation Views** | |
| **Include Browser** | The Include Browser now lets you see the hierarchy of included headers in a source file. |
| **Call Hierarchy** | The Call Hierarchy view shows you who calls a function. |
| **Type Hierarchy** | The Type Hierarchy view shows you how C/C++ types are related. |
| **Indexer Improvements** | The indexer is greatly improved with more features and accuracy: |
| **Debugger** | |
| **GDB Hardware Debugging** | The GDB Hardware Debugging feature provides a specialized debug launch configuration that allows you to start up GDB using CDT's GDB integration in a specialized manner that allows for sessions that connect to hardware debugging devices, such as JTAG and BDM, that support an integration with GDB. In particular it allows you to specify commands that are executed when GDB connects to the device, optionally specify an image to load onto the target, and specify commands that set up the target for execution. |
| **Breakpoint Actions** | You can now attach actions to a breakpoint. These are additional activities performed when a breakpoint is hit. CDT 4.0 comes with four standard actions that: play a sound, log text to an expression to the console, resume execution, or run an external tool. |
| **Contextual Launch Support** | CDT 4.0 now takes advantage of the contextual launch support in Eclipse 3.3 to make the run and debug commands just "do the right thing" for C/C++ projects.    Just click the Debug button to start a debug session instead of manually creating a launch configuration. |
| **Locating Source Files** | When the debugger can't find a source file using the path in the symbol table it now lets you locate it directly instead of having to configure a source lookup path. |
| **Other Improvements** | |
| **Project Build Button** | The toolbar now includes a button that builds the active configuration of the selected project. It's menu lets you quickly build another configuration. |
| **MinGW Toolchain Integration** | Support for the GNU toolchain on Windows now specifies two separate toolchains Cygwin and MinGW. The MinGW toolchain integration automatically detects the location of your MinGW install and sets the paths appropriately.  MinGW tool chain for new projects:    MinGW debugger launch configuration: |
| **Task Tags** | The Task View now includes entries based on tags in your C/C++ source files.    Task Tag options: |
| **IBM XL C/C++ Toolchain Integration** | If you install the optional CDT feature "XL C/C++ Compiler Support," xlc and xlC toolchain integration is available. Then when you create a new project, the XLC project types are available for selection.    xlC compiler options are available in the project properties: |
| **UPC Support** | If you install the optional CDT feature "Unified Parallel C Support," the Unified Parallel C programming language is supported in the editor. After installing the feature, you can create a source file with extension ".upc" and the UPC language constructs are recognized. |