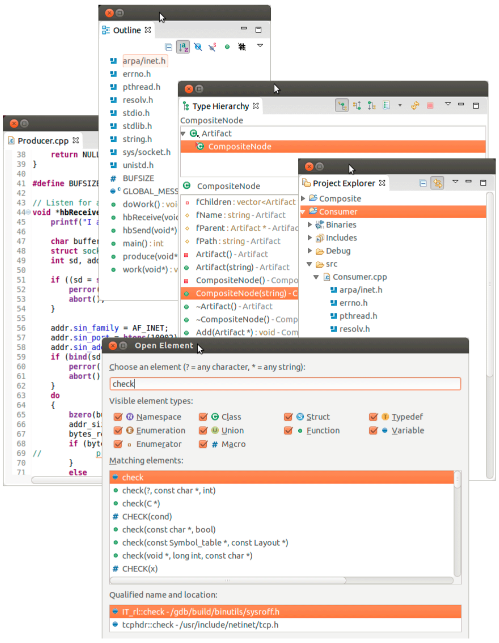
C/C++ Development Tooling

The Leading IDE for C/C++ Development Tooling

The C/C++ Development Tooling (CDT) provides a fully open source commercial-grade C/C++ IDE based on the Eclipse platform. It runs on all major OS's and supports development for many target platforms. The CDT integrates with major compilers and debuggers including GCC, Clang, and GDB. It is the de facto standard used by the large majority of chipmakers and RTOS vendors to support their various tool chains.



```
38      return NULL
39  }
40
41  #define BUFSIZE
42
43  // Listen for a
44  void *hbReceive
45      printf("I a
46
47      char buffer
48      struct sock
49      int sd, add
50
51      if ((sd = s
52          perror(
53          abort()
54      }
55
56      addr.sin_family = AF_INET;
57      addr.sin_port = htons(10002)
58      addr.sin_add
59      if (bind(sd
60          perror(
61          abort()
62      }
63      do
64      {
65          bzero(bu
66          addr_si
67          bytes_r
68          if (byte
69  //              p
70          }
71          else
```

Open Element

Choose an element (? = any character, * = any string):

check

Visible element types:

☑ Namespace ☑ Class ☑ Struct ☑ Typedef
☑ Enumeration ☑ Union ☑ Function ☑ Variable
☑ Enumerator ☑ Macro

Matching elements:

- check
- check(?, const char *, int)
- check(C *)
- CHECK(cond)
- check(const char *, bool)
- check(const Symbol_table *, const Layout *)
- check(void *, long int, const char *)
- CHECK(x)

Qualified name and location:

- IT_rl::check - /gdb/build/binutils/sysroff.h
- tcphdr::check - /usr/include/netinet/tcp.h

CDT offers rich refactorings and an extensible static analyzer with customizable pre-defined rules and checkers. It leads innovation in multi-core debugging as well as integrating with cutting-edge tracing tools.

Modular and Extensible

Being an Eclipse project itself, the CDT is in the best position to leverage the incredible wealth of functionality available through the Eclipse marketplace. Tap into the vast list of plugins available to extend your IDE and support your developers in tackling the wide variety of problems they may face. Or turn to commercial plugins to address specific issues pertinent to your situation. Or even develop plugins in house to solve problems with a proprietary dimension.

Benefits

- Highly-customizable industrial-strength C/C++ IDE
- Consolidates all development tasks into one tool
- Improves developers' productivity
- Allows the creation of better code, faster bug fixing
- De-facto IDE for Embedded Systems companies
- High popularity with developers eases the learning curve, supports quick ramp-up
- GNU Toolchain Integration as well as others
- Embedded Linux, RTOS, bare metal support
- Easily extensible through open source, home-grown, or commercial plugins
- Over 1800 plugins available in Eclipse Marketplace
- Debugging tools for highly complex problems
- Advanced Multicore Debugging tools
- As-you-type Static Code Analysis
- Allows the enforcement of project or company coding standards
- Extremely fast navigation and refactoring
- Highly integrated with source management software such as Git

Refactorings and Source Assists

CDT provides a variety of powerful C/C++ refactorings. They enable the user to quickly and safely proceed with automated code changes. One such refactoring is the ability to rename symbols (variables, constants, method names), that can span multiple files. Another one is taking a block of code and extracting it to a separate function. Similarly it's possible to use the "extract constant" refactoring to extract an expression to a local variable, a global variable or a constant.

CDT also provides some useful tools to assist the user in managing their C/C++ source code, such as the practical "Organize Includes" command, which automatically adds missing #include statements and removes unused ones. Also available are code indentation assist, the automated generation of getter/setter methods and auto-formatting of source code according to predefined, customizable rules.

C/C++ Development Tooling

Static Analysis

CDT provides a customizable and extensible code analysis frame-work that includes out-of-the-box rules and checkers. It allows users to flag coding errors upon request, during compilation, or even as the user types. Results are integrated with compilation errors, highlighted in the editor and complemented with quick-fix support wherever available.

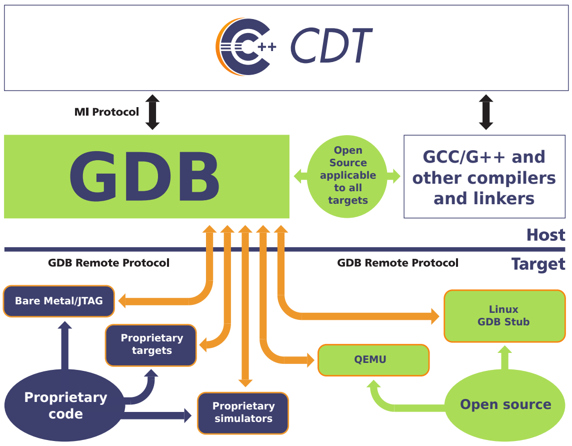
Unit Testing

With the CDT, developers can perform their unit test activities directly in the IDE. The C/C++ unit test integrations provide the ability to develop, run and examine the results of tests. Full code navigation is provided, as well as time measurements, statistics, selective test execution, execution history, and more. The currently supported frameworks are Boost.Test, Qt Test, and Google Testing Framework, and more can be added.

Debugger

The integration of GDB into CDT provides to the users some leading edge technologies, including multicore support, such as:

- All-stop and Non-stop execution modes
- Multi-process debug within a single session, multi-target with different sessions
- Pretty-printing of STL structures
- Enhanced-expressions using pattern matching and grouping
- Reverse debugging
- Project-less debugging
- Linux Kernel resource display
- Multicore Visualizer view to monitor and control target graphically
- Dynamic-printf technology allows adding printouts to processes with no recompilation or redeployment
- Stand-alone Debugger provides CDT's debugging features in a lightweight package, usable without any previous configuration
- Full GDB console: now get all the features of GDB, including command completion and history, without leaving the IDE



CDT

MI Protocol

GDB

Open Source applicable to all targets

GCC/G++ and other compilers and linkers

Host

GDB Remote Protocol

GDB Remote Protocol

Target

Bare Metal/JTAG

Proprietary targets

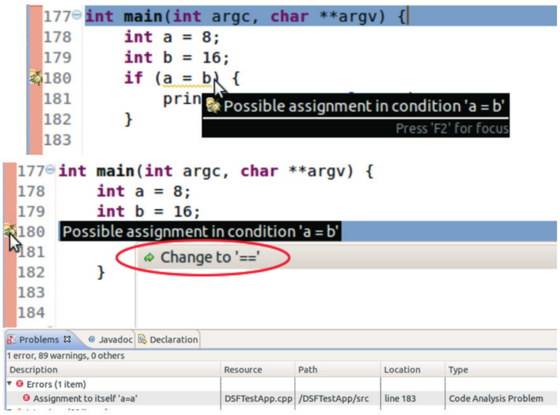
Proprietary code

Proprietary simulators

QEMU

Linux GDB Stub

Open source



```
int main(int argc, char **argv) {
    int a = 8;
    int b = 16;
    if (a = b) {
        prin
    }

```

Possible assignment in condition 'a = b'

Press 'F2' for focus

```
int main(int argc, char **argv) {
    int a = 8;
    int b = 16;

    }


```

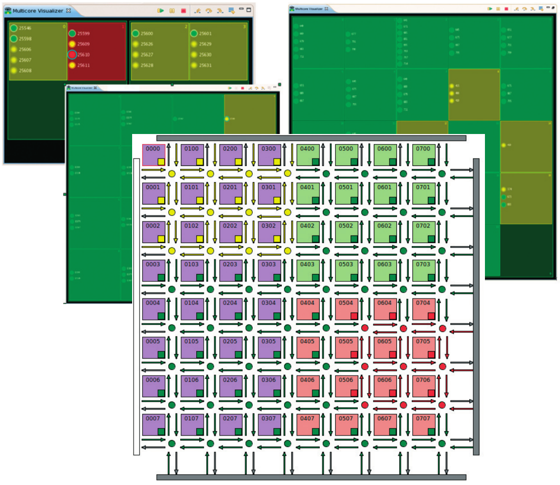
Possible assignment in condition 'a = b'

Change to '=='

1 error, 89 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (1 item)				
Assignment to itself 'a=a'	DSFTestApp.cpp	/DSFTestApp/src	line 183	Code Analysis Problem

Multicore Visualizer



What is Eclipse?

Eclipse is a community for individuals and organizations who wish to collaborate on commercially-friendly open source software. Its projects are focused on building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle. The Eclipse Foundation is a not-for-profit, member-supported corporation that hosts open source projects and helps cultivate both an open source community and an ecosystem of complementary products and services.

Resources

CDT Eclipse project:
https://www.eclipse.org/cdt/

Multicore Debug WG:
https://wiki.eclipse.org/CDT/MultiCoreDebugWorkingGroup

CDT Developers Summits:
https://wiki.eclipse.org/CDT/summits

Eclipse Foundation