Eclipse CDT 10.0 and beyond

Alexander Fedorov, ArSysOp
Who am I?

Alexander Fedorov @ ArSysOp

Eclipse Platform Committer
Eclipse CDT Committer
Eclipse Passage Project Lead

Leading IDE effort at Software Task Group
Why Eclipse CDT is the best choice?
Project Maturity

- 18 years of active development
- About 35k of commits
- About 100 contributors
- 2.3M LOC in the main repo

and Functionality

- Configure and Build
- Navigate and Refactor
- Analyse and Fix
- Debug and Run
- Extend and Integrate
High Configurability

Configure the locations where various GNU RISC-V toolchains are installed. The values are stored within Eclipse. Unless redefined more specifically, they are used for all projects in all workspaces.

Default toolchain: GNU MCU RISC-V GCC
Toolchain name: GNU MCU RISC-V GCC
Easy Extensibility

Eclipse Platform
- Workspace hooks
- Natures and builders
- Annotations and markers
- Actions and controls
- View and perspectives
- Debug model
- Launch configurations

Eclipse CDT
- Project templates
- Build definitions
- Error parsers
- Code formatters
- Doc comment tools
- Debug info visualizers
- Indexer policies
SimRel Components

- Eclipse Modeling Framework
- Eclipse Orbit and Eclipse Platform
- Eclipse Parallel Tools Platform (PTP)
- Eclipse EGit and Eclipse Mylyn
- Eclipse TM4E and Eclipse Wild Web Developer
- Eclipse LSP4J and Eclipse LSP4E
- Eclipse Linux Tools
Who is using Eclipse CDT?
Who is actively contributing?

Organization Contribution Activity:
Commits on this project by supporting organization over the last three months.

Active Member Companies:
Member companies supporting this project over the last three months.
What’s changed in Eclipse CDT 10.0?
Eclipse CDT 10.0 Overview

Build
- Autotools
- GCC
- Make
- Meson
- Managed Builder
- MS Windows

CMake
- Core
- ARM
- HPE NonStop
- Intel
- Microsoft
- Nvidia
- UI
- Qt

Codan
- CXX

Core
- Browser
- Doxygen
- Model
- Native
- Parser
- Template Engine
- UI
- Utils

Cross
- Launch Remote
- Launch Serial

Debug
- Standalone
- GDB
- DAP

LSP
- clangd
- cquery

DSF
- GDB
- PDA
- Multicore Visualizer

JTAG
- GDB
- DAP

Launch
- Docker
- Flatpack

Memory
- Floating Point
- Transport

Terminal
- Serial
- Telnet
- Remote
- SSH
CMake CDT Indexer Support

- Origin: https://github.com/15knots/cmake4eclipse
- Better user experience in C/C++ editors:
  - a. code completion
  - b. jump to declaration
  - c. folding of inactive #ifdef's
- Employs compilation database (compile_commands.json)
- Compiler built-in detection
- Best for gcc, gcc based cross-compilers and clang
- Extension point to support commercial compilers
Language Server Support (experimental)

Important!
Use Generic Editor
Eclipse CDT 10.0 Breaking Changes

- Requires Java 11 as a minimum.
- Environment Variables are always case sensitive in CDT.
- Environment variables no longer support \${} to avoid expanding.
- Arduino plug-ins and features removed.
- LRParser, XLC and UPC removed.

Learn more
Eclipse Embedded CDT
Eclipse Embedded CDT

- create/build/manage embedded ARM/AArch64/RISC-V applications
- ready to run templates for some ARM Cortex-M processors
- debugging support via JTAG/SWD
- examine and modify peripheral registers during debug sessions
- supports a wide range of 32 and 64-bit toolchains

Started to supply “Embedded C/C++” package for 2020-12!

Learn more [https://projects.eclipse.org/projects/iot.embed-cdt](https://projects.eclipse.org/projects/iot.embed-cdt)
Eclipse Embedded CDT on Raspberry Pi 4 (based on Eclipse Platform for Aarch64)
Eclipse Embedded CDT on Raspberry Pi 4 (not a toy!)
Eclipse CDT and Open Hardware Group
CORE-V IDE

Eclipse Modeling Framework

Eclipse Platform

Eclipse CDT

Eclipse Embedded CDT (GNU MCU/ARM Eclipse Plug-ins)
CORE-V IDE: planned items

- Integrate GCC-based toolchain from Embecosm
- Integrate LLVM-based toolchain from Thales
- Add “Hello World” sample project
- Provide project templates
- Publish binaries to be a foundation for downstream solutions

https://github.com/openhwgroup/core-v-ide-cdt
Eclipse CDT and Platform IO
Embedded development

PlatformIO is a professional collaborative platform for embedded development that support multiple IDE including Eclipse

- **800+ target boards** (development kits)

- **20+ software frameworks**
  
  (Arduino, ARM mbed, CMSIS, ESP-IDF, FreeRTOS, STM32Cube, Zephyr RTOS, and others)

- **30+ semiconductor architectures and development platforms**
  
  (ARM, AVR, Espressif 8266/32, MCS-51, MSP430, PIC32, STM8, RISC-V, and others)

- Over **10,000 libraries**

- **All famous operating systems**
  
  (Windows, macOS, Linux, FreeBSD, Linux ARMv6+, card-sized PCs)
PlatformIO

- Multi-board and Multi-architecture programming experience
- Debugging, Unit Testing, Static Analysis, Firmware Inspection, and Remote Development out-of-the-box
- Developers can work simultaneously on the same embedded project using different development environments and the favourite operating system
- Code for any supported framework can be compiled and uploaded to a target platform in minutes
- Developers no longer have to manually find and assemble an environment of toolchains

Eclipse integration

ECLIPSE 2020 CON
Creating Safe Products with Eclipse CDT
ISO 26262-2: 6.4.5-13

Safety Plan (Process)

- System Safety Concept
- Hardware Safety Concept
- Software Safety Concept
- Tools: ISO 26262-2: 6.4.6.5.i

Create Safe Product

Safely Case

- System SC
- Hardware SC
- Software SC
- Tools:

Safe Tool Usage Report

TCR

TSM

TQP

TQR
Qualification Kit → IDE Integration → Project Metadata

Safe Tool Usage Report

The used version '4.9.3' of tool 'gcc' was qualified successfully
Eclipse CDT needs your contribution
Step 1: Eclipse Account

1. Create account at https://www.eclipse.org/

2. Sign Eclipse Contributor Agreement (electronically)

3. Specify your GitHub id in your Eclipse profile
Step 2: Bugzilla

1. Login to Bugzilla  
   https://bugs.eclipse.org/bugs

2. Configure notifications  
   https://bugs.eclipse.org/bugs/userprefs.cgi?tab=email

3. Actualize existing and create new records
Step 3: Gerrit

1. Login to Gerrit  https://git.eclipse.org/r/

2. Upload SSH keys  
   https://git.eclipse.org/r/settings/#SSHKeys

3. Configure notifications  
   https://git.eclipse.org/r/settings/#Notifications
Step 4: Oomph

1. Download Eclipse Installer
2. Switch to “Advanced” mode
3. Select “Eclipse IDE for Eclipse Committers” package
4. Select “CDT” project
Step 4: Oomph (2)

Gerrit

User ID

ECLIPSE 2020 CON
Step 5: Patch

1. Select a bug
2. Prepare a patch
3. Compose commit message.
4. Commit and Push to Gerrit
5. Follow the review comments

Contact us: cdt-dev@eclipse.org
Thank you!

Join the conversation:

@EclipseCon  |  #EclipseCon
Evaluate the Sessions

Sign in and vote at Eclipsecon.org:

-1 0 +1