CDT 3.0 Release Review

CDT Development Team

August 8, 2005

CDT 3.0 – Introduction

- C/C++ Development Tooling
- Major Milestones
  - Project launched in July 2002
  - CDT 1.0 Dec. 2002
  - CDT 1.1 (May 2003)
  - CDT 1.2 (Oct 2003)
  - CDT 2.0 (June 2004 - sync with Eclipse 3.0)
  - CDT 2.1 (Dec 2004)
  - CDT 3.0 – targeting August 15th
- Continuing to grow adoption
  - CDT being used in variety of domains – embedded, Linux, some enterprise, deeply embedded
  - CDT as basis for other languages (e.g. Fortan)

CDT Ecosystem

- Commercial products
  - QNX Momentics
  - IBM WSDD and Rational Sw Architect
  - Timesys Timestorm
  - Tensilica Xtensa Xplorer
  - Redhat Entreprise Linux
  - Montavista DevRocket
  - PalmOS Developer Suite
  - Intel C++ Compiler
  - TI Code Composer Essentials
  - HP Remote Development
  - Altera NIOS II IDE
  - Xilinx Platform Studio
  - …
- Prototypes
  - Rockwell Collins
  - Symbian

- Participating companies
  - IBM
  - Intel
  - QNX
  - Timesys
  - Tensilica
  - Redhat
  - Montavista
  - WindRiver
  - Altera
  - Rockwell Collins
  - Altera
  - PalmSource
  - Ericksson
  - Nortel
  - Cisco
  - And others…

- Google Stat
  - Google “Eclipse CDT” return 60000+ hits

Requirements and Planning Process

- All enhancements request and bugs were logged in Bugzilla
- Bugzilla was used to solicit feedback from community on enhancements and issues.
- 52 enhancement request were implemented and 776 issues were resolved
- A Plan was posted with all planned feature development and schedule
- Commitment for development was recorded in the plan and features that were not committed to were classified as deferred.
- Project progress was discussed in newsgroup and regular project calls.
- Project Progress was reflected in milestone plan.

CDT 3.0 – Themes and Major Features

- Core- Parser Correctness - Theme: Enterprise Ready - Work focusing on making the CDT parser more accurate and complete.
- Core - AST/DOM - Theme: Design Extensibility: be a better platform - Work focusing on producing an Abstract syntax tree/DOM for C++.
- Managed Build - Extensions- Theme: Enterprise Ready & User Experience - Work on extending the CDT managed build system.

Schedule

- Schedule was set in January 05. A series of pre-release milestones and release candidate milestones were defined in relation to the platform project 3.1.
- The final pre-release milestone was slipped two weeks in order to allow committers time to complete certain features before the first release candidate. (+2 weeks)
- An additional release candidate milestone (RC4) was added to the end of the schedule to allow committers time to address documentation (+2 weeks).

Milestones

- **M5 - Monday Mar. 14, 2005** - stable build reflecting progress,
- **M6 - Monday April 11, 2005** - stable build reflecting progress,
- **M7 - Monday, June 13, 2005 -** stable build, feature complete, APIs frozen
- **RC1 Monday, June 27,2005** - Release Candidate
- **RC2 Monday, July 11, 2005** - Release Candidate
- **RC3 Wednesday July, 27, 2005** – Code Freeze - software complete.
- **RC4 Friday August, 12, 2005** – Final Release Candidate

CDT 3.0 – Documentation and Testing

- This release specifically set aside two weeks of time for committers in order to update their related documentation.
- functional testing was carried out by committers and the CDT user community
- 2600 JUnit test were run against each build
- Sanity testing was performed on each milestone and release candidate build and results were posted to dev group.
- The Bug count targeted for a specific milestone needed to be zero before it was declared to be an official candidate.

Key Features – CDT Core

- **DOM Based Language Backend** The CDT 3.0 is now served by a much more powerful language model in the form of a DOM that helps make everything from the C/C++ Outliner to Searching to Re-factoring faster and more accessible to plugin developers.
- **K&R C Language Support** In addition to the new DOM serving the backend source code model, the CDT native parser is also much more tolerant of K&R C Language dialects.
- **Faster Searches** Significant work has gone into improving the performance and speed of the C/C++ parser and the efficiency of the indexer. As a result, searches (both general and specific as in search for references and declarations) are faster and more accurate than in previous releases.
- **Binary Parser Options** The binary parsers have been made more generic with the capability to redefine what external commands they use for data interpretations.

Key Features – CDT Core (cont'd)

- **PathEntry Variables** -The PathEntry project properties are important for the correct operation of the CDT parser. The new **PathEntry Variables** configuration allows developers to configure a common set of variables that will allow projects to be shared more easily between different developer set-ups.
- **New Discovery Capabilities** Further work has been done to facilitate the discovery of compiler and project settings, including the ability to read in the output from a build output file.
- **Indexer Selection** The C/C++ source indexers are now plugable components, each providing different feature support, allowing users to select from: **No indexing, CTags utility based indexing, CDT parser based indexing**
- **Drag and Drop Support** The **C/C++ Project** and the **C/C++ Outline** views now both have improved support for dragging and dropping of code elements and binaries.

Key Features – CDT Core

- **Editor Platform Alignment** The C/C++ Editor is more aligned with the platform preferences and settings. As well, a few new features have been added to the **C/C++ Editor** syntax highlighting, including the coloring of operators, braces and numbers and a new in-place outline that can be activated via **Ctrl+O**

Key Features – Managed Build

- **Configuration Overhaul** - improved user experience and workflow for developers (and integrators). The new **C/C++ Build** project properties has a number of new enhancements:
  - Full control over the output artifact and extension
  - Pre and Post build command execution
  - Per project and per configuration environment variable settings
- **Build Macro Support** The managed build properties now support a wide set of macros that can be incorporated into your project. The system environment settings are inherited and individual macros can be set globally or local to a particular project that define Strings, Files, Directories or lists of any of these items

Key Features – Managed Build (cont’d)

- **Per File Build Steps** Users now have the ability to define a custom build step for any source file in the project. This build step can generate additional output sources.
- **Configuration Renaming** Build configurations can now be renamed through the selection of the project's build properties.

Key Features – Debug

- **New Modules View** All of the binary components, or modules, are now displayed in the debugger's Modules view. This includes dlls, shared objects and application binaries themselves. Additionally, each module can be expanded allowing the user to put breakpoints directly on binary components without the need of opening the C/C++ Project view
- **Memory View** The debugger now makes use of the standard platform Memory view allowing for a number of format translations and multiple memory location inspection.
- **Register Groups** The **Registers** view now supports the concept of register groups, allowing users to defined collection of registers that are relevant for their particular debugging environment.

Key Features – **CDT 3.0 - Extenders**

The Managed Build System (MBS) has undergone some additional changes in CDT 3.0 to better support tool integrators. Specifically the following features have been added:

- Additions to the tool-chain object model to support more complex tool-chains, particularly regarding the definitions of the inputs and outputs to a tool.
- The ability to define option categories for the entire tool-chain in addition to per tool.
- The ability to define custom "new project" wizard pages.
- Additional callbacks to support dynamic behavior, including:
  - Specifying whether an option is currently visible, enabled, and used in command line generation.
  - Specifying whether the tool set used by a tool-chain is installed on the system.
  - Defining environment variables and build macros a tool-chain.
- A mechanism for supporting multiple versions of a tool-chain.
- A mechanism for converting configurations from using one tool-chain, or a version of a tool-chain, to another.
- Extraction of the GNU tool-chain into its own plug-in in order to allow the Managed Build System to be packaged without GNU tool-chain support when

Non-code aspects

- Updated User and ISV docs
- Addition of ISV docs
  - DOM
  - Managed Build extensibility doc
- Creation of a "what's new" doc
- Articles for journals (proposed or in progress)
  - Managed Build article for Dr. Dobbs
  - SDTimes article
- Active CDT community
  - Mailing lists (~4000 emails in last year)
  - Newsgroup (~3000 posts in last year)

Code metrics and APIs

CDT Version	# Plugins	# Classes	LOC
2.1	19	2524	253308
3.0	26	3170	332638

- Reasonable code growth from 2.1 to 3.0
- Major CDT "infrastructure" changes (DOM/AST, MBS)
  - New Core APIs which will require a few releases to mature
  - DOM, CModel and Indexer API to solidify and become "platform" with next major CDT release
  - This includes rounding out of API documentation and test coverage to conform to Eclipse Quality document
- Debug interfaces (CDI) quite mature
  - Several third-party implementations exist
  - However, DSDP/Platform/CDT debug discussions may result in CDI changes

CDT APIs

Name	Project	Doc status	Test cases	# of clients	Compatibility	Maturity
C Debug Interface (CDI)	Debug	Javadoc	Some JUnit, manual sanity	3	Platform*	3 years
GDB MI	Debug	_	_	Multiple	MI version 1 and above	3 years
C Model	Core	Javadoc	Yes	3	Provisional	2 years
DOM	Core	Javadoc Partial “howto”	Yes	4	Provisional	6 months
Binary Parser	Core	Javadoc	Yes	5	Platform	2 years
Indexer	Core	Some Javadoc	Yes	2	Provisional	6 months
Code Formatter	Core	Example “indent” integration	No	1	None	6 moths
Toolchain definition (xml)	MBS	Sample schema ISV doc	Some	2	Provisional	1 year

* Based on debug framework discussion with DSDP/Platform/CDT, the CDI may break compatibility in the future. All current implementors of CDI are participating in these discussions.

Architectural Issues

- Major rework of CDT core infrastructure in CDT 3.0
  - DOM/AST and language backend with improved parsing
  - Migration of CDT services to use the DOM
    - Code assist
    - Indexer
    - Search
  - Due to tight timeframes, some services still use old “DOM”
    - Outline view and core model
    - Migration to new DOM will be complete in next release
- Some decoupling of builder plugins from CDT core
  - First step to enables stand-alone or re-use with other languages
- C++ class browser and type hierarchy disabled due to stability issues
  - Needs re-architecting to efficiently use the indexer and core model
  - Will be addressed in next release

Defect Statistics

- Between July 1st, 2004 and July 25th, 2005
  - New defects: 2440
  - Fixed and resolved (inc. feature requests): 2184
- Current state (3.0 and future)
  - Release backlog: 898 (inc. enhancements)
  - Blocking: 6
  - Critical: 3
  - P1: 0
- CDT 3.0 Project Defects
  - Current defects: 3
  - Zero defect count criteria for shipping

Process: Committers and Contributions

- Committers
  - Total of 23 committers to CDT (IBM, QNX, Redhat, Intel, TI)
    - Addition of 3 committers during 3.0 cycle: Intel (2) and TI (1)
  - Committers voted in following top-level project process
- Contributors
  - WindRiver, Symbian, Timesys, Montavista, Palmsource, Altera, HP
  - Additional contributors from QNX/IBM/TI
  - Community contributors
- Miscellaneous
  - Re-alignment of components and committers along 3 major subsystems: core, debug and build
  - Coordination with other projects (platform, Fortran, DSDP, etc.)

Standards

- C/C++ Parser
  - ISO C++ support
  - C99
  - K&R C
- Toolchain integration
  - GDB and GCC
  - GDB MI version 1 and 2

Intellectual Property

- In-bound Contributions
  - All contributions made under EPL
  - All prior contributions (CPL) re-licensed under EPL
  - EMO approval for CPL->EPL transition on March 1st, 2005
- Out-bound licensing
  - CDT provided under EPL
  - Redhat contributed plugins (oprofile, RPM) contain some dual-licensed code (EPL and GPL)
    - Approved by EMO in Jan. 2005
    - Shipped separately from CDT
- Due Diligence
  - External patches and contributions reviewed by committers
  - List of contributors (from patches and changelogs) reviewed by project lead prior to release
  - Spreadsheet of contributors sent to Eclipse legal and EMO

Forward Plan

- A draft plan for the next release has been checked in to the CDT CVS repository. This plan is under development and is targeted for Fall 05.
  - See http://dev.eclipse.org/viewcvs/index.cgi/*checkout*/cdt-home/plans/CDT_3.1_plan.html?rev=HEAD&cvsroot=Tools_Project
- A CDT conference is planned this fall. One aspect of it is future release planning. It will be focused in the next significant release which will be July 06.
  - Please see the CDT mailing list or contact dkeefe@qnx.com for more information
- Patch releases will be done as needed.