



## *Developing C/C++ Applications with the CDT*

Doug Schaefer  
QNX Software Systems  
Eclipse CDT Project Lead





## *What is the CDT?*

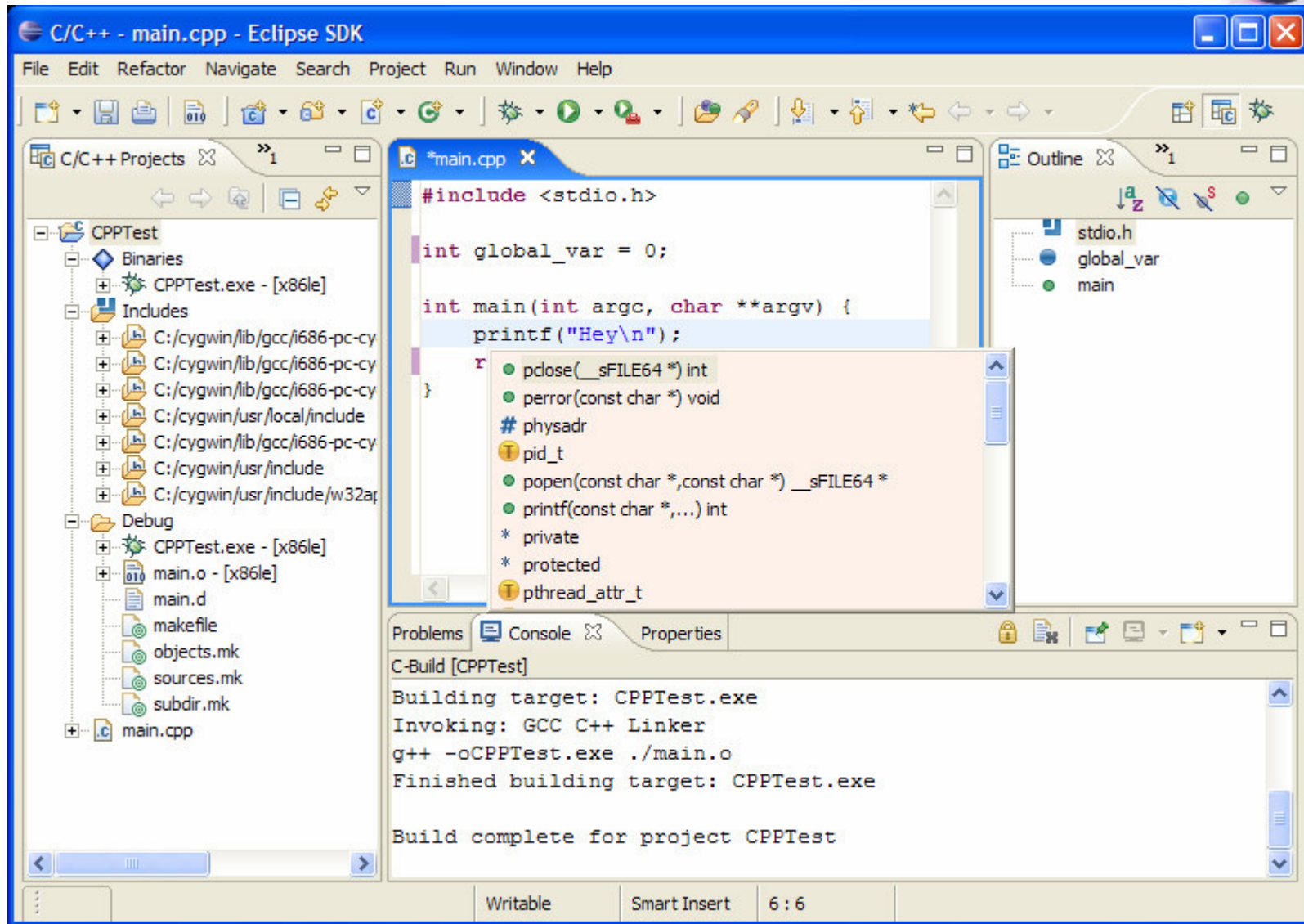
- Edit/Build/Debug cycle for C and C++ projects
  - Source code editor
  - Build integration with third party build tools
  - Visual debug similar with integration to third party debuggers
- Source parsing and indexing
  - Drives source navigation features
  - Content assist, Search, Outline

## *CDT Edit Perspective*



- Standard JFace Text Editor
  - Keyword highlighting, key bindings
- Outline View
  - Using internal parser
- Content Assist
  - Using internal parser and index
- C/C++ Projects View
  - Showing CDT specific things: includes, binaries
- Build Console Output

# C/C++ Perspective

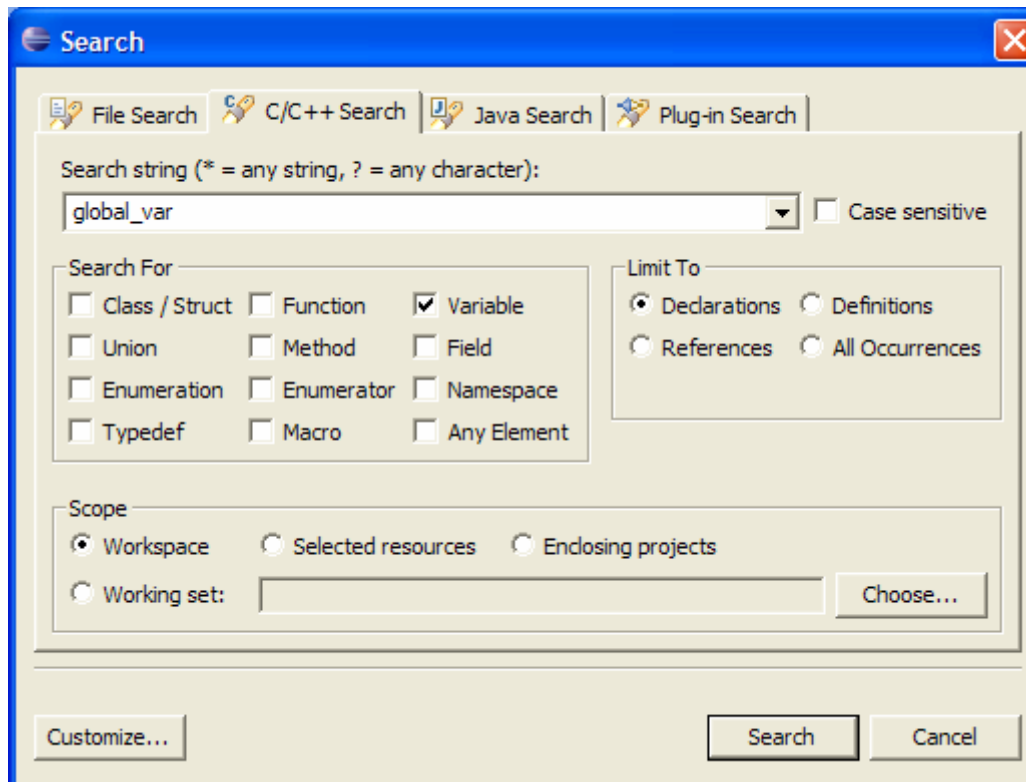


# CDT Search



- Index created on resource change
  - Uses internal parser
- Searches for declarations, definitions & references
- Searches for types, variables, functions, macros
- Search can be invoked from:
  - Search Dialog
  - Context menu on Outline View and C/C++ Project View
  - Action on text selection in editor

## C/C++ Search Dialog

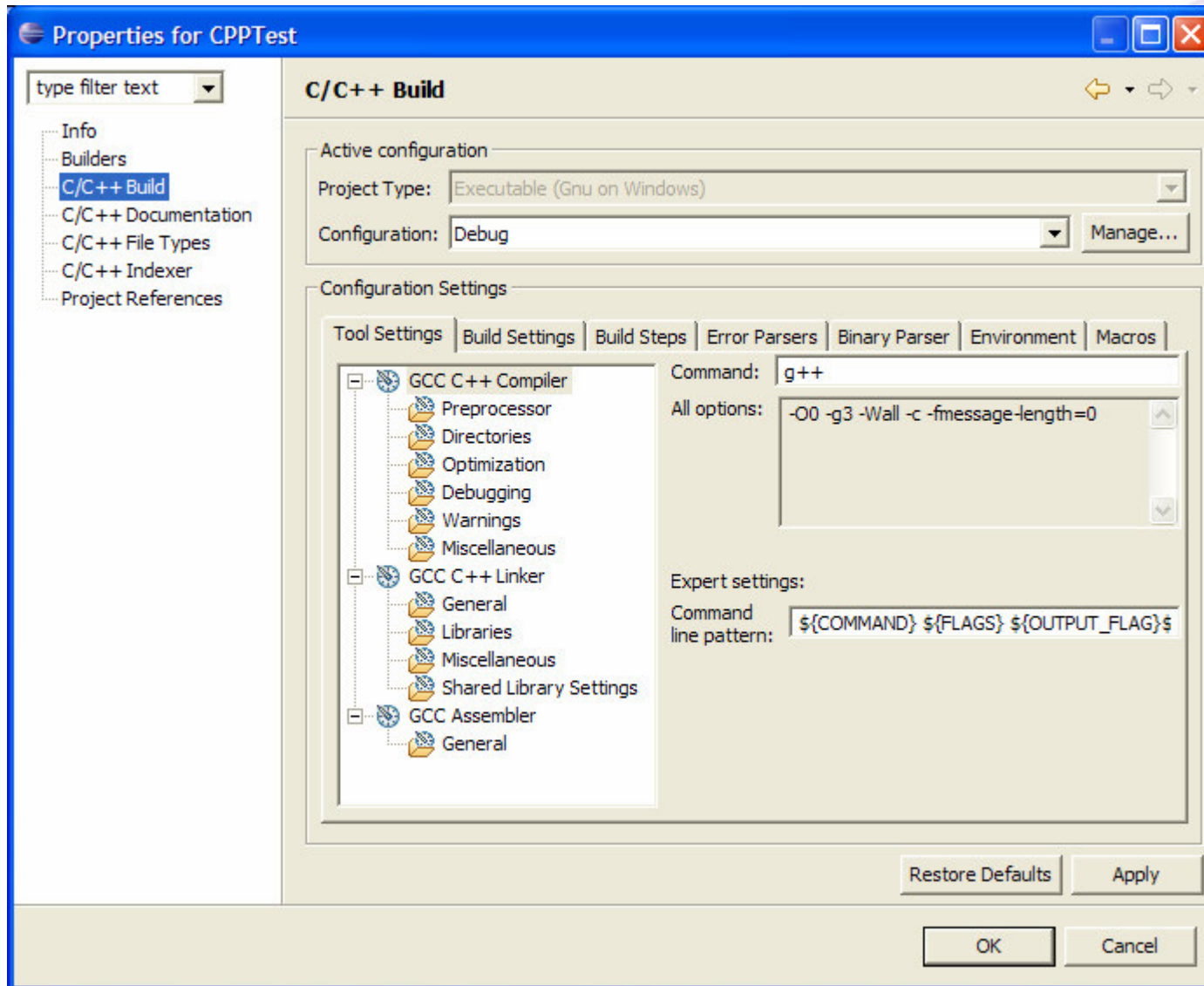


# *Build*



- Standard Build
  - Calls any external tool to do builds
  - Makefile editor for editing Makefiles
  - Error parsing to mapping errors to Problem markers
- Managed Build
  - Extension points for defining tools and options
  - Generated GUI for editing build settings
  - Generate build file
  - Hook up appropriate binary parsers, error parsers

# Managed Build Settings



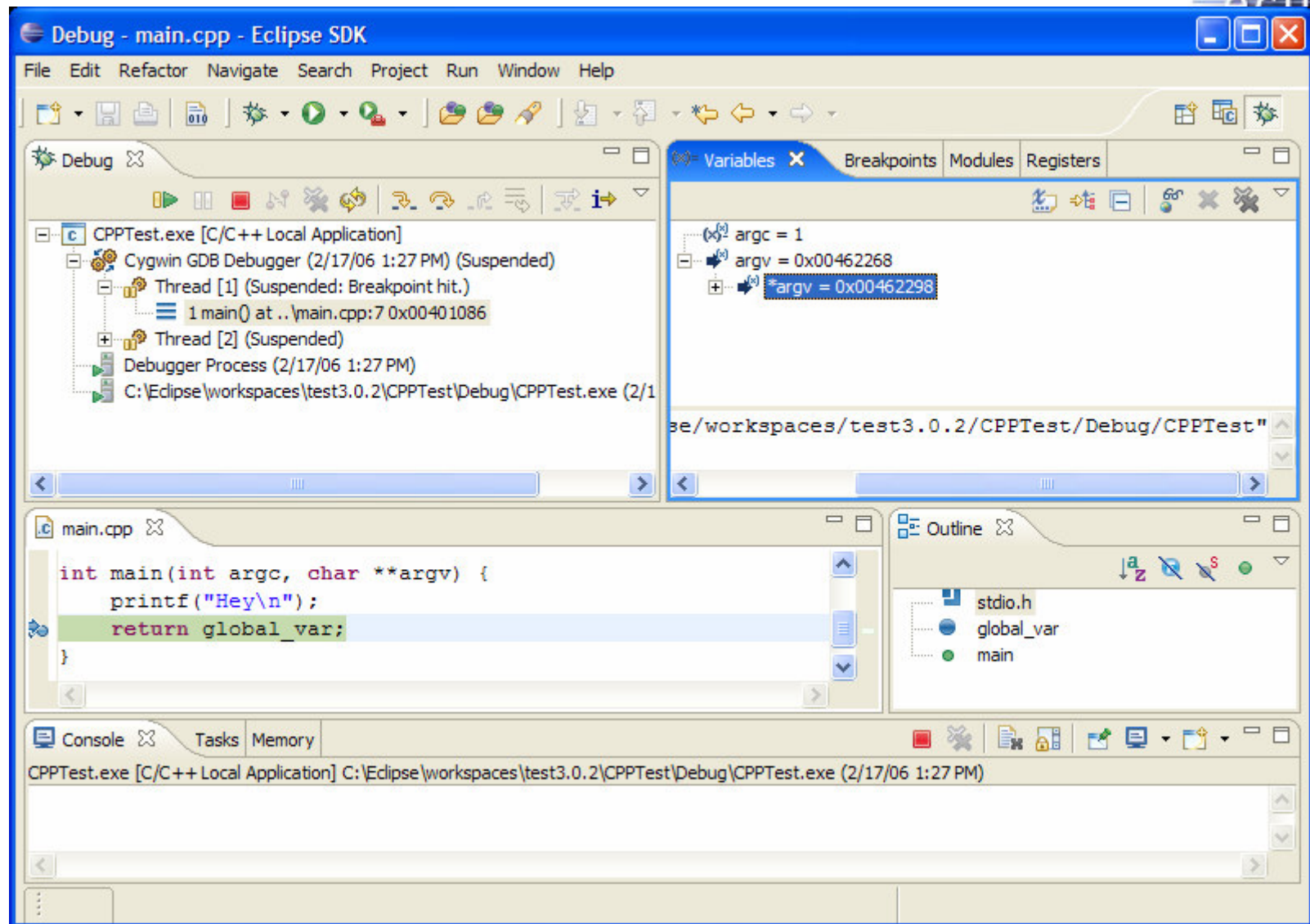


# *Debug Perspective*

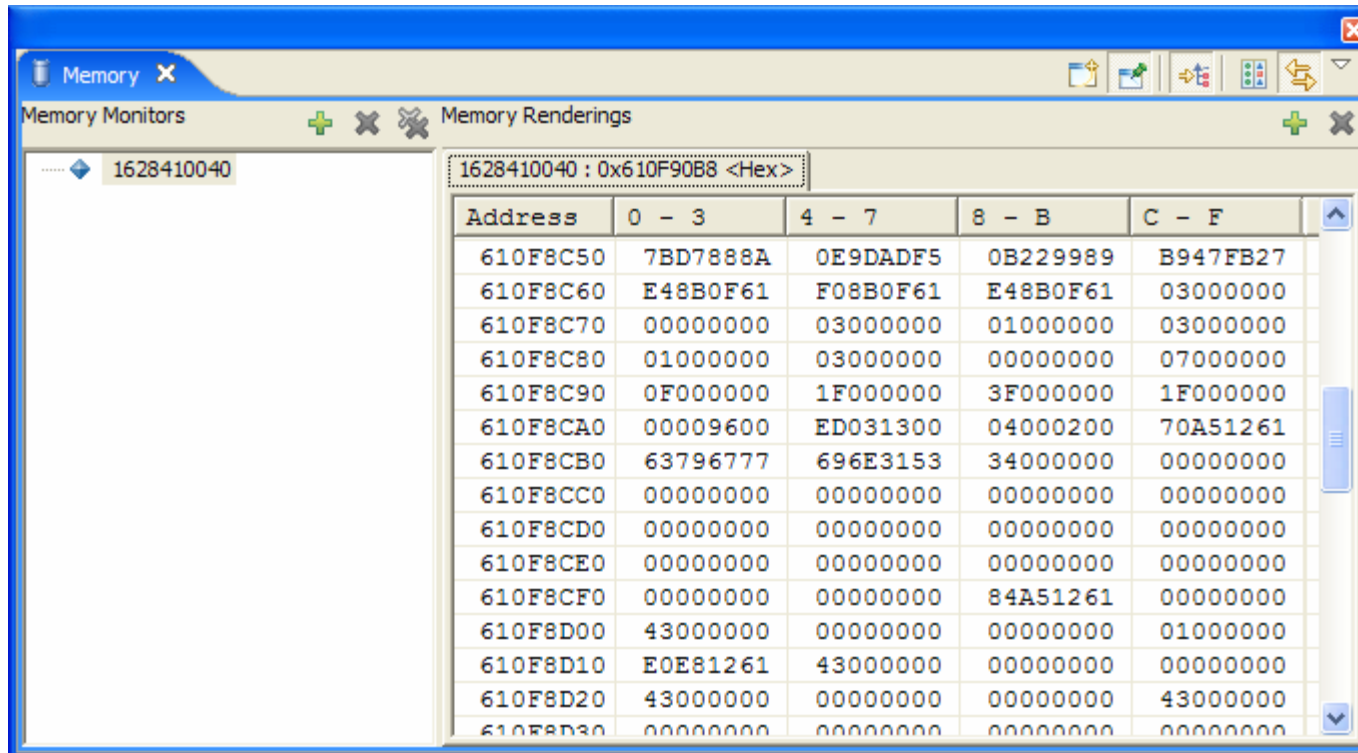


- Standard Eclipse Debug Extension
  - Breakpoints
  - Execution Control
- Provides a layer to adapt Eclipse API to C-ish Debugger Engines
- Views
  - Register
  - Modules
  - Memory
  - Disassembly

# Debug Perspective



# Memory View



Address	0 - 3	4 - 7	8 - B	C - F
610F8C50	7BD7888A	0E9DADF5	0B229989	B947FB27
610F8C60	E48B0F61	F08B0F61	E48B0F61	03000000
610F8C70	00000000	03000000	01000000	03000000
610F8C80	01000000	03000000	00000000	07000000
610F8C90	0F000000	1F000000	3F000000	1F000000
610F8CA0	00009600	ED031300	04000200	70A51261
610F8CB0	63796777	696E3153	34000000	00000000
610F8CC0	00000000	00000000	00000000	00000000
610F8CD0	00000000	00000000	00000000	00000000
610F8CE0	00000000	00000000	00000000	00000000
610F8CF0	00000000	00000000	84A51261	00000000
610F8D00	43000000	00000000	00000000	01000000
610F8D10	E0E81261	43000000	00000000	00000000
610F8D20	43000000	00000000	00000000	43000000
610F8D30	00000000	00000000	00000000	00000000

# Disassembly View

A screenshot of the Eclipse IDE's Disassembly View window. The window has a blue title bar with the text "Disassembly" and a close button. The main area displays assembly code for a function named "main". The code is color-coded: keywords like "int", "push", "mov", "sub", "and", "call", "printf", "return", and "ret" are in blue; registers like "%ebp", "%esp", "%eax" are in green; and memory addresses and constants are in black. The instruction at address 0x00401075, "call 0x401120 <\_\_main>", is highlighted with a light blue background. A mouse cursor is visible over this instruction. The code ends with a "ret" instruction at address 0x0040108c.

```
int main(int argc, char **argv) {
0x00401050 <main>:    push    %ebp
0x00401051 <main+1>:    mov     %esp,%ebp
0x00401053 <main+3>:    sub     $0x8,%esp
0x00401056 <main+6>:    and     $0xffffffff0,%esp
0x00401059 <main+9>:    mov     $0x0,%eax
0x0040105e <main+14>:   add     $0xf,%eax
0x00401061 <main+17>:   add     $0xf,%eax
0x00401064 <main+20>:   shr     $0x4,%eax
0x00401067 <main+23>:   shl     $0x4,%eax
0x0040106a <main+26>:   mov     %eax,0xffffffffc(%ebp)
0x0040106d <main+29>:   mov     0xffffffffc(%ebp),%eax
0x00401070 <main+32>:   call    0x401090 <_alloca>
➔ 0x00401075 <main+37>: call    0x401120 <__main>
    printf("Hey\n");
0x0040107a <main+42>: movl    $0x402000, (%esp)
0x00401081 <main+49>: call    0x401130 <printf>
    return global_var;
0x00401086 <main+54>: mov     0x403010,%eax
}
0x0040108b <main+59>: leave
0x0040108c <main+60>: ret
```



## *Who uses the CDT?*

- Embedded developers
  - Flexible build and debug integrations enable many platforms
- Linux developers
  - Powerful cross-platform tools
  - Built-in support for GNU tools
- High performance computing
  - Basis for Fortran (Photran) and Parallel Tools Platform projects

## *CDT Popularity - Marketing View*



- “We also asked a question to determine which Eclipse projects were being used or planned to be use. No surprise that JDT came out on top with 57.5% but CDT came in second with 47.3%. I think this is amazing and a big congratulations to Doug and the CDT team. Could there be a day when more people are actually using CDT than JDT?”
- “Well, IDC has just made available a new research report that estimates there to be 2.27 million Eclipse users worldwide.”
- Does that mean 2.27 million x 47.3% CDT users?

## CDT 3.1.1 Stats



- CDT 3.1.1 Released Sept 29, 2006

	callisto	cdt update	cdt zips	cdt sdk	Total	%
win32	7767	8355	11606	2713	30441	68.8%
linux-x86	2227	3126	3573	1154	10080	22.8%
macosx	811	686	441	195	2133	4.8%
linux-x86_64	281	373	341	155	1150	2.6%
solaris	72	24	113	46	255	0.6%
aix	9	0	56	20	85	0.2%
linux-ppc	15	5	34	15	69	0.2%
linux-ia64	9	0	0	19	28	0.1%
Total	11191	12569	16164	4317	<b>44241</b>	

## CDT 3.1.0 Stats



- CDT 3.1.0 Released June 30, 2006

	callisto	cdt update	cdt zips	cdt sdk	Total	%
win32	61482	8994	39101	17726	127303	71.7%
linux-x86	16995	4370	11663	4247	37275	21.0%
macosx	5200	836	1091	416	7543	4.2%
linux-x86_64	1987	297	1119	441	3844	2.2%
solaris	281	89	440	188	998	0.6%
aix	29	20	189	53	291	0.2%
linux-ppc	70	32	115	46	263	0.1%
linux-ia64	24	9	77	33	143	0.1%
Total	86068	14647	53795	23150	<b>177660</b>	



## CDT 3.0.2



- CDT 3.0.2 Released Feb 9, 2006

	update	zip	sdk	Total	%
win32	140945	69414	24697	235056	67.6%
linux-x86	60528	24097	6773	91398	26.3%
macosx	8310	1571	496	10377	3.0%
linux-x86_64	4902	1893	561	7356	2.1%
solaris	890	871	351	2112	0.6%
aix	173	341	82	596	0.2%
linux-ppc	213	208	75	496	0.1%
linux-ia64	46	139	58	243	0.1%
Total	216007	98534	33093	<b>347634</b>	

## *CDT Commercial Adoption*



- Included in many commercial products (that I know about)
  - QNX Momentics
  - Altera NIOS II IDE
  - IBM Rational Software Architect, Rational Systems Developer
  - Intel C++ Compiler for Linux
  - Texas Instruments Code Composer Essentials
  - Nokia Carbide.c++
  - Timesys LinuxLink Developer Suite
  - MontaVista DevRocket
  - Red Hat, SUSE, Debian Linux

## *CDT Commercial Adoption*



- Mercury Computer Systems MultiCore Plus
- HI-TECH IDE for HI-TIDE 3
- Xilinx Platform Studio SDK
- Tensilica Xtensa Xplorer
- STMicroelectronics
- CoWare ConvergenSC
- KPIT Cummins KPIT Corona
- Etnus TotalView
- HP Remote Development

## *CDT Commercial Adoption*



- Corelis CodeSymphony IDE
  - Amontec sdk4arm
  - eCosPro Developer's Kit
  - Ultimate Solutions LinuxScope-JTD
  - Ronetix Starter Kit for ARM9
  - Lattice Semiconductor LatticeMico32 Development Tools
  - Xilinx Embedded Development Kit
- 
- And they keep on coming...



## *Word is spreading...*

- Articles

- <http://www-128.ibm.com/developerworks/java/library/os-ecc/index.html?ca=drs->
- [http://www.codeproject.com/useritems/CPP\\_Dev\\_eclipse\\_CDT.asp](http://www.codeproject.com/useritems/CPP_Dev_eclipse_CDT.asp)
- [http://www.computer.org/portal/site/cise/menuitem.92a12adebee18778161489108bcd45f3/index.jsp?&pName=cise\\_level1\\_article&TheCat=1001&path=cise/2006/v8n4&file=sci.xml&](http://www.computer.org/portal/site/cise/menuitem.92a12adebee18778161489108bcd45f3/index.jsp?&pName=cise_level1_article&TheCat=1001&path=cise/2006/v8n4&file=sci.xml&)
- <http://www.kineteksystems.com/white-papers/mixedjavaandc.html>
- <http://www.webpronews.com/expertarticles/expertarticles/wpn-62-20060914RevisitingCConEclipse.html>

- 11,000 occurrences of cdtproject in Google.

- <http://www.google.ca/search?hl=en&q=cdtproject&meta=>

*Demo*



## CDT 4.0



- Part of Europa Release, June 2007
- Managed APIs
- Offline Indexing
  - Pre-generated and sharable indexes for SDKs, etc.
- New views of index information
  - E.g. Call Hierarchy View
- New flexible debug architecture
- Internal Builder for Managed Build

*Thank you*

