

Wind River Compiler

Companies developing embedded devices are looking to increase productivity, meet challenging schedules, and reduce project risks. Software developers must contend with fitting more complex applications into the same memory space and adding new capabilities while maintaining real-time performance. The choice of embedded tools has a significant impact on these goals, and an embedded cross-compiler is a key tool, affecting every piece of C or C++ code written for a project. The choice of a compiler can determine the overall memory footprint of the device software and also have a significant impact on the performance of an embedded system.

The Wind River Compiler tool suite (formerly known as the Diab Compiler) includes software development tools, such as a C/C++ compiler, an assembler, a linker, ANSI C and ANSI C++ standard libraries, and an instruction set simulator. It is based on the industry-hardened Diab Compiler technology and has a proven track record for generating robust, compact, and fast-executing code for thousands of designs and millions of devices. Wind River Compiler also provides the control and flexibility required to meet the demands of device software development, allowing

developers to incorporate Wind River Compiler into any build environment. The compiler offers numerous options for controlling code generation and assisting with porting code developed with other tools.

Product Highlights

Wind River Compiler provides the following capabilities for embedded software developers:

- Generate faster-executing code and smaller software footprints
 - Optimize to allow applications to run faster or with minimal use of memory
 - Reduce bill of materials (BOM) cost by minimizing hardware requirements
- Provide reliable, stable, and mature code generation technology
 - Enhance product quality and deploy mission-critical applications
 - Reduce risk of catastrophic failures and cost of downtime
- Ensure compliance with the latest ANSI and ISO standards
 - Produce code interoperable with other embedded application binary interface (EABI)-compliant compilers
 - Use with almost any debugger due to ELF and DWARF compatibility
- Control and have flexibility with options and build characteristics
 - Customize the build options for unique constraints

- Deploy into any build environment
- Improve your productivity, speed time-to-market, and reduce schedule risks
- Take advantage of flexible business models
 - License the compiler for one target architecture on a perpetual basis or license for all target architectures with an annual subscription
- Reduce project risk with responsive, knowledgeable, award-winning support and special services
 - Take advantage of long-term support model for customized compiler versions
- Get support for a wide variety of target architectures
 - Ensure your investment is maintained when you decide to migrate to a different processor architecture

Benefits

High Performance

Wind River Compiler uses sophisticated optimization technology to generate exceptionally fast, compact, high-quality object code. This reduces costs and improves competitive position by doing the following:

- Reducing hardware costs and allowing you to use lower-frequency devices and less memory than your competition and improving the profitability of your product
- Helping you develop applications that run faster and include more functionality and features than your competition, making your product more desirable to the market
- Improving time-to-market and reducing risk by eliminating the need to hand-optimize code at the last minute to meet aggressive performance goals, ensuring your product will be first to market

Table of Contents

Product Highlights.....	1
Benefits.....	1
Technical Highlights.....	3
Toolchain Components.....	4

Technical Specifications	4
Supported Targets.....	4
Professional Services	5
Support Services	5
How to Purchase Wind River Solutions.....	6

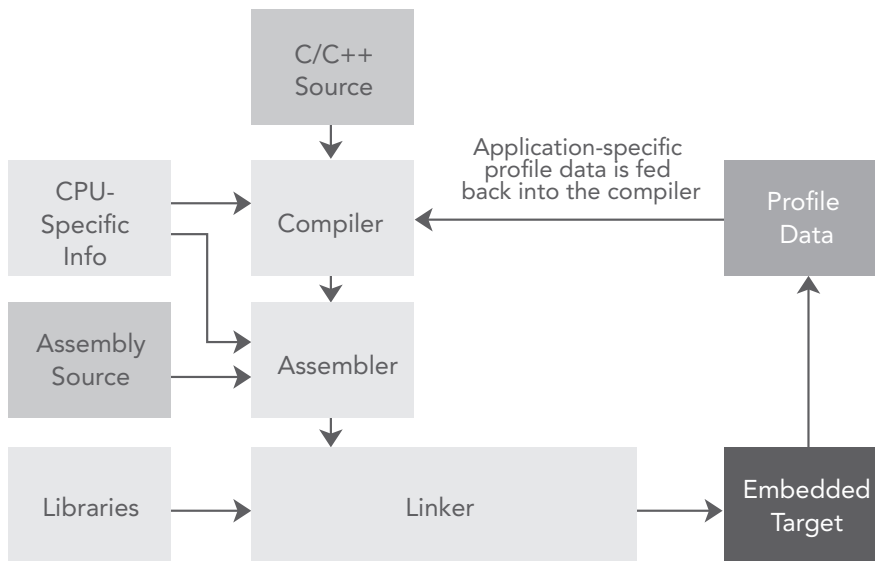


Figure 1: Profile-driven, application-specific optimizations

Wind River Compiler uses a wide range of highly refined global, local, processor-specific, and application-specific (profile-driven) optimization techniques to generate code that runs faster with a smaller footprint. Whole program optimization permits the compiler to inline functions across multiple modules and source files, significantly boosting performance. Profile-driven optimizations employ the compiler’s capability to instrument the code and collect profile information specific for the application being developed. This information is then fed back into the compiler, enabling it to make better decisions when performing function inlining, register allocation, branch prediction, and other optimizations, further improving the application performance and footprint (see Figure 1).

Flexibility

Embedded devices differ in the processors and microcontrollers in use, the amount and types of memory available, the interfaces and peripherals, the operating systems, and the development environments. This means all embedded developers face challenges that are unique to their projects.

Wind River Compiler lets you control the Compiler’s output to match an application’s requirements. For example, you can do the following:

- Select various optimizations to balance execution speed with code size
- Use multiple debugging options, depending on how much memory is available on the target device
- Generate position-independent code (PIC) or position-independent data (PID)
- Generate packed and byte-swapped data structures
- Generate PROMable code
- Create a large number of data sections for customized memory layouts
- Use absolute addressing or the extensive memory mapping capabilities included in the compiler to target devices with unique memory layouts and constraints

Reliability

Mission-critical applications, such as those used in automotive and aerospace and defense (A&D) industries, require absolute reliability from the code running on the system and the software tools that generate this code—the compiler in particular. To minimize defects, code must be rigorously tested and must conform to standards. Wind River Compiler has been used in many mission-critical applications in aerospace, avionics, defense, automotive under-the-hood, medical, and industrial industries.

With more than 20 years of technology investment, Wind River Compiler has been thoroughly tested in the field and in Wind River’s own test lab. Wind River Compiler has achieved certified conformance to the POSIX PSE52 standard. In addition, it is used to compile code for Wind River VxWorks DO-178B Platform and Wind River VxWorks 61508 Platform. This experience and stability significantly reduces the risk of selecting a compiler for your mission-critical application.

Standards Compliance

Wind River Compiler uses the Edison Design Group’s compiler front end to ensure full compliance with the most recent ANSI/ISO C and C++ compiler standards, including the following:

- ISO/IEC 9899:1990 (“C89”) standard
- ISO/IEC 9899:1999 (“C99”) standard
- ISO/IEC 14882:2003(E) C++ standard

This ensures that your application code can conform to these coding standards and will remain portable to other environments. In addition, Wind River Compiler supports older standards such as Kernighan and Ritchie and System V.3 UNIX in order to ease porting of legacy applications. Wind River Compiler adheres strictly to the EABI as well as ELF and DWARF binary and symbolic formats for maximum tool interoperability.

Flexible Business Models

Wind River Compiler complements its technical advantages with a choice of proven business models. Under our Enterprise License Model, the platform is offered as an annual, per-developer subscription, applicable across the enterprise and any target architecture supported by Wind River Compiler. This flexibility allows pervasive use of Wind River Compiler across many development projects. The subscription includes Wind River support and all product updates. Our Perpetual License Model is for companies desiring project-based licensing for a single supported architecture.

Support Services

Wind River Customer Support, an award-winning and Service Capability and Performance (SCP)-certified organization, provides support for Wind River Compiler. Your subscription to Wind River Compiler includes full maintenance and support, delivered through Wind River's Online Support (OLS) website and our worldwide technical support team.

If an embedded device needs to be supported for many years or even decades, Wind River Long-Term Support Service lengthens the support window for Wind River Compiler beyond the standard product life cycle. This service also provides customized updates and quality assurance to minimize the impact of the compiler changes to your binary code. Contact Wind River Customer Support (www.windriver.com/support) or your Wind River account manager for details.

Technical Highlights

Selectable Speed/Size Optimizations

Certain compiler optimizations involve trade-offs between execution speed and code density. With Wind River Compiler's numerous compiler switches, users can choose whether to optimize for speed or code size.

Flexible Mixing of C/C++ and Assembly

Wind River Compiler provides several methods for mixing C/C++ and assembly code. `asm` macros can be used to inline sections of assembly code that can be invoked as a function. `asm` strings provide a simple way to embed assembly instructions. Wind River Compiler also offers a number of compiler macros that correspond to assembly instructions that improve compiler optimization.

Whole Program Optimization

Wind River Compiler's whole program optimization capability allows the compiler to optimize calls between functions in different source files. This can improve execution efficiency by allowing function inlining across different modules.

Profile-Driven, Application-Specific Optimizations

These optimizations require two-staged compilation. In the first stage, the compiler instruments the code, which is then executed on the target or in a simulation environment on a typical dataset. The instrumented code collects profile information, which is then fed back into the compiler for the second stage of the compilation.

During the second stage, the compiler uses the profile information to further improve optimizations such as loop-unrolling, inlining, basic block reorganization, register allocation, and branch prediction. Because a typical dataset is supplied during the first phase of the compilation, these optimizations will be highly tuned for the given application.

Easy Interrupt Handling

Wind River Compiler makes it easy to handle interrupt processing for embedded systems by providing interrupt keywords and interrupt pragmas. For instance, the compiler provides the ability to design an interrupt vector table at C level without using any assembly language.

Multiple Debugging Options

Wind River Compiler provides flexible controls for generating debuggable code. Users can control the trade-offs between the amount of debug information vs. the speed of debugging, and performance optimizations vs. ease of debugging.

Position-Independent Code (PIC) and Data (PID)

Wind River Compiler can generate code and data that can be loaded at any address. This is useful in devices that dynamically load/unload modules.

Volatile Keyword, or All Memory Is Volatile

Users can mark areas of code as volatile, which prevents the compiler from optimizing away data accesses. This feature is useful for accessing memory-mapped device I/O.

Control of Structure Formats

Wind River Compiler can reduce footprint by packing structures and ensuring that all padding is removed. The compiler can also create byte-swapped structures in which it swaps the byte order for data structures as they are stored in memory, allowing the communication of information in a byte order different than the device's native byte order. This capability can help optimize performance when sharing data between big and little endian processors.

Extensive Link Command Language for Memory Mapping

Every embedded device has a unique memory layout, with various types of memory available, such as fast RAM, flash, and shared memory. The link command language provides users fine-grain control to lay out code and data in memory in the optimal way.

Absolute Addressing from C and Assembler

Variables and functions can be assigned to specific memory addresses. This feature is particularly useful for accessing memory-mapped device I/O and for setting up interrupt vectors.

Ability to Generate PROMable Code

Wind River Compiler can generate code that can be burned into ROM and can access memory that is moved into RAM at startup time.

Support for Multiple Object Module Formats

The compiler supports ELF, COFF, IEEE-695, and S-Records and can generate object modules in multiple formats.

Link-Time Lint Checker

The lint facility is a powerful tool for finding common C programming mistakes at compile time and at link time, including unused variables and functions, missing return statements, constants out of range, and function call mismatches. Link-time checking finds inconsistencies across modules, which is impossible to do at compile time.

Run-Time Error Checker

Wind River Compiler supports a run-time error checker that can quickly identify the root cause of software errors during program development. The run-timer error checker can detect memory leaks, stack overflows, improper use of pointers, and memory allocation errors. This powerful tool can save time and enhance code quality prior to integration and QA activities.

Instruction Set Simulator

The Wind River instruction set simulator (ISS) is a simulator for executables and a disassembler for object files and executables. The ISS accurately simulates the instructions of the target architecture allowing developers to write and debug applications without the need for target hardware.

Toolchain Components

Wind River Compiler includes the following programs and utilities:

- **Driver:** Intelligent wrapper program invokes the compiler, assembler, and linker
- **Assembler:** Macro assembler invoked automatically by the driver program or as a complete standalone assembler generating object modules; includes the following key features:
 - Conditional macro assembler with more than 30 directives
 - Unlimited number of symbols
 - Debug information for source-level debugging of assembly programs
- **Linker:** Precisely controls allocation, placement, and alignment of code and data; includes the following key features:
 - Links object modules into absolute or relocatable modules
 - Reads/writes/mixes ELF and COFF object files
 - Generates fully EABI-compliant ELF/DWARF output for tool interoperability
- **Libraries**
 - Complete reentrant C libraries compliant with ANSI/ISO, POSIX, and SVID standards
 - Complete C++ library (STL)

- Full complement of math libraries, including IEEE-754 appendix functions
- Fast, efficient floating-point libraries
- Library source code
- **Archiver/librarian:** Creates and maintains libraries
- **Instruction set simulator:** Simulates the core instructions of the target processor and allows it to run C and C++ programs with the simulated environment
- **Profiler:** Instruments application code that when executed saves profile information for viewing by the profiler; profile data can also be automatically fed back to the compiler for additional code optimization based on execution paths
- **Object file converter:** Provides conversion routines for generating S-Record or IEEE-695 output formats and also provides the following:
 - Symbol table management
 - Detailed code size reports
 - C++ symbol name demangler
 - Munch routine
- **Documentation:** Includes extensive documentation specific to the chosen architecture, with all manuals available in PDF formats; a detailed "Getting Started" manual enables users to get up to speed quickly and enhances the out-of-the-box experience

Technical Specifications

Supported Host Operating Systems

- Windows XP Professional, Service Pack 2
- Windows Vista (Business and Enterprise)
- Red Hat Enterprise Linux Workstation 4, Update 5 or 6, x86 (32-bit)
- Red Hat Enterprise Linux Desktop with Workstation option 5, Update 2*
- Red Hat Fedora Core 9, x86-64
- SUSE Linux/openSUSE** 10.2*
- SUSE Linux/openSUSE** 11, x86-64
- Ubuntu Desktop 8.04, Update 4, x86-64
- Solaris 9 (with GTK), Update 09/05, SPARC 32-bit
- Solaris 10 (with GTK), Update 09/05, SPARC 32-bit

**Both 32-bit and 64-bit versions are supported. For 64-bit version, only x86-64 is supported. IA-64 (original Intel Itanium Architecture) is not supported.*

***SUSE Linux has been renamed to openSUSE.*

Supported Targets

Wind River Compiler supports a wide range of embedded architectures and provides processor-specific optimizations for each of these architectures. This range of choice offered by the compiler permits continuity should you decide to migrate from one architecture to another.

ARM

- Thumb Instruction Set
- Thumb2 Instruction Set
- All ARM7, ARM9 cores
- ARM9 processors
 - Atmel AT9x
 - Broadcom BCM21xx
 - Freescale i.MX1,L,S,2x
 - Marvell MV88x
 - TI DM644x
 - TI OMAP 7xx, 8xx, 16xx, 17xx
 - TI OMAPV 103x
- ARM11, ARM11 MP processors
 - Freescale i.MX31
 - TI OMAP 24xx, 591x
 - TI OMAPV 2230
- ARM Cortex cores
 - Cortex-A8
 - Cortex-R4
 - Cortex-M1
 - Cortex-M3

ColdFire

- MCF51xx
- MCF52xx
- MCF53xx
- MCF54xx

MIPS

- MIPS4, MIPS16, MIPS32, MIPS32R2, MIPS64
- MIPS 4Kx, 4KEx, 5Kx
- MIPS32 M4K
- MIPS 20Kx, 24Kx, 25Kx
- MIPS 34K
- R3xxx, R4xxx, R5xxx
- AMD AU12xx
- Broadcom BCM11xx, 12xx, 14xx
- Broadcom BCM33xx, 35xx, 47xx
- Broadcom BCM53xx, 5621x, 58xx
- Broadcom BCM63xx
- Broadcom 70xx, 71xx, 73xx, 74xx
- Cavium CN3xxx, CN5xxx
- IDT RC3xxx, RC4xxx, RC5xxx
- NEC VR41xx, 54xx, 55xx, 77xx

- Philips PR19xx, 39xx, 44xx
- Philips PNX30xx, 73xx, 83xx, 85xx
- PMC-Sierra RM79xx, 9xxx
- RMI XLR, XLS
- Toshiba TX49xx

Power

- P.A. Semi PA6T-1682M

PowerPC

- PowerPC VLE instruction set
- PowerPC SPE instruction set
- AltiVec instruction set
- PowerPC 4xx
 - AMCC PPC 405xx
 - AMCC PPC 440xx
 - Xilinx Virtex-II Pro and Virtex 4
- 603e, including
 - Freescale MPC603e, MPC52xx, MPC824x, MPC827x, MPC828x
- G2 core, including
 - Freescale MPC825x, MPC826x
- G4 core
 - Freescale MPC7xx, MPC74xx
- e200 z0z1/z3/z4/z6/z7 cores
 - Freescale MPC55xx
 - Freescale MPC56xx
- e300 core
 - Freescale MPC83xx
- e500/e500v2 cores
 - Freescale MPC85xx
- e500mc core
 - QorIQ P20x0, P40x0
- e600
 - Freescale MPC86xx
 - Freescale MPC5xx
 - Freescale MPC8xx
 - IBM/Freescale PPC6xx
 - IBM/Freescale PPC7xx

XScale

- Intel IXP4xx
- Intel IXP2xxx
- Intel IOP3xx
- Marvell PXA2xx/3xx

Freescale MCORE

- MCORE2XX, MCORE3XX
- MCORE200x, MCORE 21xx

Renesas SuperH

- SH-1 core
- SH-2, SH-2A cores
- SH-3 core
- SH-4, SH-4A cores

Freescale MC68K

- MC680XX, MC683XX
- MC68000, MC68008, MC68010, MC68020, MC68040, MC68060
- MC6830X, MC68322, MC68349, MC68356

SPARC

- SPARC, SPARCLite

Infinion TriCore

- TriCore TC12, TC13, TC20
- TC11xx, TC17xx, TC19xx

Intel x86

- Intel Pentium, Pentium Pro, Pentium II, Pentium III, Pentium 4
- Intel Core, Intel Core 2
- AMD Opteron, Athlon

For current target availability, contact your Wind River sales representative.

Professional Services

Wind River Professional Services helps companies reduce risk and improve competitiveness. Our team delivers device software expertise within structured engagements that directly address key development challenges and contribute to the success of our clients. Our track record of timely delivery and in-depth understanding of market and technology dynamics makes Wind River a valuable implementation partner for clients worldwide. Based on our commercial-grade project methodology, service offerings include device design, board support package (BSP) and driver optimization, software system and middleware integration, and legacy application and infrastructure migration.

Services for Wind River Compiler

Whether you select Wind River Compiler as a standalone product or as part of our platform solutions, Wind River Professional Services knows how to jump-start your development efforts. Types of services provided may include the following:

- Extending Wind River Compiler processor support
- Adopting Wind River Compiler for your target OS
- Applying Wind River Compiler optimizations for maximum performance

Installation and Orientation

Proper installation and orientation of Wind River Compiler means you won't waste time solving easily avoidable problems before you can begin your next development project. Wind River offers an Installation and Orientation Service to ensure that your project starts on time and without hassle by delivering the following:

- **Onsite installation:** Guided install on your hardware and host platform, along with a sample build process, demonstrations, and examples of customizations
- **Hands-on orientation:** Architecture, development file system, adding open source packages, porting drivers, addressing design issues
- **Advice:** Introduction to Wind River support channels and processes, additional services, project review, and consultation

The Wind River Installation and Orientation Service will expedite your path to productivity, allow you to rest assured that we have eliminated a common source of user error, and help you realize all of the platform's potential.

Support Services

Wind River provides full technical support for our development solutions, including Wind River Compiler. Our products are backed by the most comprehensive customer support network in the Device Software Optimization (DSO) industry.

Wind River's global support organization is staffed with experienced engineers who have extensive knowledge of Wind River products and device software development. With 10 major support centers and 15 additional support hubs worldwide, our local experts can help diagnose problems, provide guidance, and answer basic "How do I...?" questions.

In addition to standard support, Wind River offers Long-Term Support Services for Wind River Compiler customers. Long-Term Support lengthens the support window beyond the standard product life cycle to meet any needs that require special support and maintenance

services. In addition, the service provides custom bug-fixing and quality assurance to minimize the impact of compiler changes to your code. Contact Wind River Customer Support (www.windriver.com/support) or your account manager for more details.

Support is available 24/7 at our Online Support website or by email at support@windriver.com. The website provides patches, manuals, the latest errata, and other announcements. Online Support also offers tech tips and application notes and answers to FAQs. Visit Online Support at www.windriver.com or consult our Customer Support User's Guide at www.windriver.com/support/resources/csug.pdf.

Wind River experts are also available for telephone support during standard business hours. If you cannot find the information you need through Online Support, contact our global support team:

**North America, South America,
Asia/Pacific**

support@windriver.com
Toll-free tel.: 800-872-4977 (800-USA-4WRS)
Tel.: 510-748-4100
Fax: 510-749-2164
Hours: 6:00 a.m.–5:00 p.m. (Pacific time)

Japan

support-jp@windriver.com
Tel.: +81 3 5778 6001
Fax: +81 3 5778 6003
Hours: 9:00 a.m.–5:30 p.m. (local time)

Europe, Middle East, Africa

support-ec@windriver.com
Toll-free tel.: +800 4977 4977
France tel.: +33 1 64 86 66 66
France fax: +33 1 64 86 66 10
Germany tel.: +49 899 624 45 444
Germany fax: +49 899 624 45 999
Italy tel.: +39 011 2448 411
Italy fax: +39 011 2448 499
Middle East region tel.: +972 9741 9561
Middle East region fax: +972 9746 0867
Nordic tel.: +46 8 594 611 20
Nordic fax: +46 8 594 611 49
UK tel.: +44 1793 831 393
UK fax: +44 1793 831 808
Hours: 9:00 a.m.–6:00 p.m. (local time)

How to Purchase Wind River Solutions

Visit www.windriver.com/company/contact-us/index.html to find your local Wind River sales contact. To have a sales representative contact you, call 800-545-9463 or write to inquiries@windriver.com.