

WIND RIVER

Wind River[®] Compiler

RELEASE NOTES

5.8

Copyright © 12/8/09 Wind River Systems, Inc.

All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means without the prior written permission of Wind River Systems, Inc.

Wind River, Tornado, and VxWorks are registered trademarks of Wind River Systems, Inc. The Wind River logo is a trademark of Wind River Systems, Inc. Any third-party trademarks referenced are the property of their respective owners. For further information regarding Wind River trademarks, please see:

www.windriver.com/company/terms/trademark.html

This product may include software licensed to Wind River by third parties. Relevant notices (if any) are provided in your product installation at the following location:
installDir/product_name/3rd_party_licensor_notice.pdf.

Wind River may refer to third-party documentation by listing publications or providing links to third-party Web sites for informational purposes. Wind River accepts no responsibility for the information provided in such third-party documentation.

Corporate Headquarters

Wind River
500 Wind River Way
Alameda, CA 94501-1153
U.S.A.

Toll free (U.S.A.): 800-545-WIND

Telephone: 510-748-4100

Facsimile: 510-749-2010

For additional contact information, see the Wind River Web site:

www.windriver.com

For information on how to contact Customer Support, see:

www.windriver.com/support

*Wind River Compiler
Release Notes
5.8*

8 Dec 09

Contents

1	Overview	1
1.1	Features	1
1.2	Installation and Licensing	2
	Special Note: Installing on Linux Hosts	2
1.3	Configuration	3
	Selecting a Target	3
1.4	Migration and Backward Compatibility	3
	Binary Compatibility with Previous Releases	3
1.5	Latest Release Information	4
2	Changes in This Release	4
2.1	Enhancements	4
	Architecture-Independent Compiler Enhancements	4
	Enhancements, Multiple Architectures	6
	Enhancements, PowerPC Architecture	6
	Enhancements, ARM Architecture	6
	Enhancements, x86 Architecture	6
2.2	Fixed Problems	7
2.3	Deprecated Features	7
2.4	Unsupported Features	7
3	System Requirements	7
3.1	Host System Requirements	7
	Windows Host	7
	Solaris Host	8
	Linux Host	8
3.2	Installation Requirements and Issues	10
	Fedora 9 64-Bit Host	10
	Fedora 11 64-bit Host	10
	Ubuntu 8.04 64-Bit Host	10
	Ubuntu 9.04 64-Bit Host	10
3.3	Target System Requirements	10

4	Usage Caveats	11
	C and C++ Compilers	11
	Linker	12
5	Known Problems	12
6	Documentation Errata	13
	Customer Services	15

Wind River Compiler

RELEASE NOTES

5.8

1. Overview

The Wind River Compiler is a complete toolkit for embedded application development, including C and C++ compilers, assemblers, linkers, utilities, and standard libraries for a variety of target CPU architectures. The compiler version shipped with this release is 5.8.0.

1.1 Features

The Wind River Compiler provides:

- a wide range of optimizations to produce fast, compact code
- hundreds of command-line options, pragmas, preprocessor directives, and special keywords for flexible operation, precise control of output, and full utilization of architecture-specific features
- special features for device software optimization, such as
 - **-Xsize-opt**, which optimizes for code size
 - Support for user-defined linker command files
 - Support for customized section placement via **#pragma section** or **__attribute__((section))**
 - Attributes to control alignment and packing of data structures
- compile-time checking to catch suspicious or non-portable constructs
- profiling to find bottlenecks in code and identify problems that cannot be detected by static analysis
- full ANSI C++ support, including templates, exceptions, and run-time type information
- integration with other Wind River tools and products

The Wind River Compiler includes the following tools:

- optimizing C and C++ compilers
- assembler
- linker

- standard C and C++ libraries
- archiver/librarian
- profiling block counter
- file dumper
- make utility
- instruction-set simulator and disassembler

1.2 Installation and Licensing

For information on installing the Wind River Compiler and configuring your product licenses, see the Wind River product installation and licensing guides. They are accessible from the following URL:

<http://www.windriver.com/licensing/documents>



NOTE: Make sure that you are using the licensing software (in particular, the **wrsd** daemon) that comes with the current version of the compiler. Using mismatched licensing software—for example, by installing from a machine running an older licensing daemon—may cause compilation to fail.

Special Note: Installing on Linux Hosts

32-bit Run-Time Libraries

In order to install the Wind River Compiler on a Linux host, your host system must include 32-bit run-time libraries. Your installation will fail if you do not have the libraries installed before running the Wind River installer (**setup_linux**). Most Linux distributions provide the libraries in their base install feature set. For some hosts, you must install additional packages. See [3.2 Installation Requirements and Issues](#), p.10 for details.

Permissions Error with **setup_linux**

Note that some Linux distributions either do not automount media, or automount with a **noexec** option that prevents execution of files found on the media.

If **./setup_linux** fails with a permissions error, you may need to remount the DVD. To do so, use the following steps:

1. Log in as root by entering the command **su** in your terminal and entering the root password.
2. Enter the following command:

```
umount dvd_mount_point
```

Where *dvd_mount_point* is the location of the DVD mount. For example, a typical Red Hat or Fedora DVD mount point might be **/media/DVD-12345-67890**.

3. Then enter the following command:

```
mount /dev/cdrom dvd_mount_point
```

Where *dvd_mount_point* is the location of the DVD mount.

4. Exit root mode with the **exit** command.
5. Set your working directory to the location of the DVD mount:

```
cd dvd_mount_point
```
6. Retry the **./setup_linux** command.

1.3 Configuration

Configuration information is available in the getting started guide.

The directory containing the tools must be included in your path variable. Windows users should map a drive letter to the remote directory where the tools reside and use the drive letter when setting the path.

Selecting a Target

If you want to configure a default target and execution environment, run **dctrl -t** from the command prompt. For more information about **dctrl**, see the getting started guide or the *Wind River Compiler User's Guide*.

Using **dctrl** is optional. A target and execution environment can also be specified with the **-t** command-line option when the compiler is invoked. For information about **-t**, see the *Wind River Compiler User's Guide* for your target architecture.

1.4 Migration and Backward Compatibility

General information on migrating your installation and applications for use with this product is available in the “Converting Existing Code” section of the *Wind River Compiler User's Guide* for your architecture.

Binary Compatibility with Previous Releases

When migrating a project to a new compiler release, it is best to recompile all source-code modules—especially C++ modules. If this is not possible, legacy object and archive files can usually be linked against output from the new compiler. Note, however, the following exceptions:

- All C++ modules built with pre-5.4 releases of the compiler (release 5.3.1 or earlier) must be recompiled.
- C modules that contain packed structures and were built with pre-5.3 releases of the compiler must be recompiled.
- C and C++ modules built for pre-5.6.1 SH targets must be recompiled.

1.5 Latest Release Information

The latest information on this release can be found in the Wind River Compiler area of the Wind River Online Support Web site:

<http://www.windriver.com/support>

This site includes links to topics such as known problems, fixed problems, documentation, and patches.

In addition, most Wind River documentation is accessible from the following URL:

<http://documentation.windriver.com>



NOTE: Wind River strongly recommends that you visit the Online Support Web site before installing or using this product. The Online Support Web site may include important software patches or other critical information regarding this release.

For information on accessing the Wind River Online Support Web site, see [Customer Services](#), p.15.

In addition, a list of known issues and other important information is installed in *installDir/readme_wrcompiler.txt*. Be sure to review this file before using the Wind River Compiler.

2. Changes in This Release

All components in a toolchain release (compiler, assembler, linker, and utilities) have the same version number, even when a particular component is unchanged. Changes are further organized by language and target:

- Changes to the C or C++ compiler apply to all targets unless otherwise noted.
- Changes to the C compiler also apply to C++, unless otherwise noted.
- Changes marked with a target name, such as ARM or PowerPC, apply to all languages unless otherwise noted.

2.1 Enhancements

Architecture-Independent Compiler Enhancements

The following enhancements apply to all supported architectures.

New Linker Options: Library Dependencies and RAM Usage

The linker option **-m8** prints library dependency information.

The linker option **-m16** generates a table in the linker map, showing the RAM usage for functions and symbols.

Updates to **dctrl** Utility

The **dctrl -t** option has been updated to show all currently supported cores and processors. It is a good idea to run **dctrl -t** to verify a given configuration of processor, object file format, floating point support, and execution environment.

The **dctrl -s** option is new. This provides a quick way to see recommended compiler flags for optimization and debugging.

-I@ Option Implemented for C++ Compiler

The **-I@** compiler option to control the search order for user-defined header files now works for both C and C++ compilers.

New Value for -Xkeep-assembly-file Option

The **-Xkeep-assembly-file** compiler option now provides a way to set the filename of the generated assembly file.

With **-Xkeep-assembly-file=1**, the name of the **.s** file is the same as the name of the source file.

With **-Xkeep-assembly-file=2**, the name of the **.s** file is the same as the name of the object file. Use this with the **-o** flag, which creates an object file with a different name than the source file.

Whole-Program Optimization

Whole-program optimization has replaced cross-module optimization as a more effective method to enable modules to inline functions defined in other modules

Like cross-module optimization, whole-program optimization enables functions to be inlined across modules. Unlike cross-module optimization, whole-program optimization does not require two compilation passes, and it does not rely on a database that can get out of sync.

New Example, Shared Symbols in Multicore Systems

The installed example files include a new example demonstrating a proof of concept for sharing symbols across different cores in a multicore system. See the source files at the following location for details:

```
installDir/diab/releaseDir/example/multi_core_share
```

Documentation Files

The product documentation, in PDF and HTML, is installed in the following directory:

```
installDir/docs/extensions/eclipse/plugins/  
com.windriver.ide.doc.wr_compiler_standalone
```

On a Windows host, the documentation is available from **Start > All Programs > Wind River > Documentation**.

On many Linux hosts, the documentation is available from the desktop menu under **Wind River > Documentation**.

For an optional documentation patch to integrate the documentation into an online help system using the Eclipse IDE, visit the Wind River Online Support Web site. See [1.5 Latest Release Information](#), p.4.

Enhancements, Multiple Architectures**New Linker Options for Generating Branch Islands**

This enhancement applies to PowerPC, M68K, ColdFire, MIPS, and ARM/THUMB/THUMB2.

Previous releases of the Wind River Compiler suite have generated branch islands only when creating a fully-linked image. The following linker options now generate branch islands with incremental links: **-r5**, **-r9**, **-r12**.. See the *Wind River Compiler User's Guide* for details.

Enhancements, PowerPC Architecture

Graph Coloring for Register Allocation

Use the **-Xcoloring** compiler option to optimize register allocation using a graph coloring algorithm.

Note that this affects stack frame usage. See the *Wind River Compiler for PowerPC User's Guide* for details.

Support for Additional VLE Processors

Additional PowerPC VLE processors are now supported. For the complete list, see the *Wind River Compiler for PowerPC User's Guide* or run the **dctrl -t** utility.

Support for Freescale P4080

The Wind River Compiler now supports the Freescale P4080 and other e500mc processors.

Enhancements, ARM Architecture

New -Xsplit-l-long Option

Under certain circumstances, the compiler for ARM will generate the following error:

Expression too complex. Try to simplify.

This can be caused by running out of available registers used to store temporary results in long long expressions.

To avoid this error, use the **-Xsplit-l-long** option. This causes the compiler to split long long expressions into intermediate expressions using temporary values. Note that this may result in slower performance because the temporary values may become stack locations instead of registers, creating extra load and store instructions.

Enhancements, x86 Architecture

Defer Stack Pops in Functions

By default, the compiler will emit code that pops the callee's frame off the stack after it returns to the calling function. With the **-Xdefer-pop** option, all these stack frame pops can be aggregated into a single pop and deferred until the end of the calling function. Deferred pops can be profitable in terms of code size when the C program consists of functions which in turn call a modest number of functions, and stack space is not too constrained.

2.2 Fixed Problems

For a list of problems fixed in the Wind River Compiler, visit the Online Support Web site (see [1.5 Latest Release Information](#), p.4).

2.3 Deprecated Features

Cross-module Optimization

Whole-program optimization has replaced cross-module optimization. See [Whole-Program Optimization](#), p.5.

2.4 Unsupported Features

M32R Architecture

The Wind River Compiler no longer supports the M32R architecture.

3. System Requirements

This section lists the minimum requirements for running the Wind River Compiler where the host and target are separate computers.

3.1 Host System Requirements

The host is the computer on which you do your development work. This section lists the minimum requirements for running the Wind River Compiler in the standard configuration.

These system requirements are for the Wind River Compiler only; they do not take into consideration any other software you are running on the host computer.

Windows Host

- One of the following:
 - Microsoft Windows XP Professional (Service Pack 2 or 3), x86-32
 - Microsoft Windows Vista Business, x86-32
 - Microsoft Windows Vista Enterprise, x86-32
 - Microsoft Windows 7, x86-32
- Administrator rights.
- Monitor capable of displaying 1024 x 768 @ 16 bpp or better.
- Intel Pentium 4 class computer with a 2 GHz processor, or a computer with higher performance.

- 1 GB of RAM (2 GB of RAM is recommended for larger projects).
- 500 MB disk space for installing all supported architectures. Installing a subset of supported architectures will require less disk space. When calculating the amount of disk space needed, be sure to reserve space for your own applications and development.
- A local DVD-ROM drive or access to a network for installation.
- A current version of a standards-compliant Web browser.
- TCP/IP must be installed on the host system, even if it is being used as a standalone PC with a serial connection to the target.
- A network interface card for debugging over a network (recommended).
- An active Internet connection is recommended during initial installation to access patches, documentation, and other important information from the Wind River Online Support Web site. (See [1.5 Latest Release Information](#), p.4.)

Solaris Host

- Sun Solaris 10 (with GTK), Update 11/06, Sparc 32-bit
- Monitor capable of displaying 1024 x 768 @ 16 bpp or better.
- A Blade 150 workstation with a 500 MHz processor, or a workstation with higher performance.
- 1 GB of RAM.
- 500 MB disk space for installing all supported architectures. Installing a subset of supported architectures will require less disk space. When calculating the amount of disk space needed, be sure to reserve space for your own applications and development.
- A local DVD-ROM drive or access to a network for installation.
- A current version of a standards-compliant Web browser.
- CDE Window Manager (recommended).
- An active Internet connection is recommended during initial installation to access patches, documentation, and other important information from the Wind River Online Support Web site. (See [1.5 Latest Release Information](#), p.4.)

Linux Host



NOTE: There are usage caveats associated with installing on Linux Hosts. Refer to [3.2 Installation Requirements and Issues](#), p.10 before installing the Wind River Compiler.

- One of the following:
 - Red Hat Enterprise Linux Workstation 4, Updates 6&8, x86-32
 - Red Hat Enterprise Linux Desktop with Workstation option 5, Updates 2&3, x86-32 and x86-64

- SUSE Linux/openSUSE 11.0 and 11.1, x86-32 and x86-64
- Novell SUSE Linux Enterprise Desktop 10, Service Pack 2, x86-32 and x86-64
- Novell SUSE Linux Enterprise Desktop 11, x86-64
- Fedora 9, x86-64
- Fedora 11, x86-32 and x86-64
- Ubuntu Desktop 8.04 Update 4, x86-64
- Ubuntu Desktop 9.04, Update 4, x86-32 and x86-64
- Monitor capable of displaying 1024 x 768 @ 16 bpp or better.
- GNOME Window Manager.
- Intel Pentium 4 class computer with a 1 GHz processor, or a computer with higher performance.
- 1 GB of RAM.
- 500 MB disk space for installing all supported architectures. Installing a subset of supported architectures will require less disk space. When calculating the amount of disk space needed, be sure to reserve space for your own applications and development.
- A local DVD-ROM drive or access to a network for installation.
- TCP/IP must be installed on the host system.
- A network interface card for debugging over a network (recommended).
- A current version of a standards-compliant Web browser.
- An active Internet connection is recommended during initial installation to access patches, documentation, and other important information from the Wind River Online Support Web site. (See [1.5 Latest Release Information](#), p.4.)
- 32-bit run-time libraries.



NOTE: In order to install the Wind River Compiler on a Linux host, your host system must include 32-bit run-time libraries. The Wind River product installation process fails if you do not have the libraries installed before running the Wind River installer (the **setup_linux** program). Most Linux distributions provide the libraries in their base feature set; for others, you must install the 32-bit libraries.

3.2 Installation Requirements and Issues

In order to install and run Wind River products on certain Linux hosts, you must resolve some package dependencies. Before installing the Wind River Compiler, perform the steps described in this section for your particular host type.

Fedora 9 64-Bit Host

Installer Requirements

Install the 32-bit libraries that are required in order to run the installer:

```
# sudo yum install glibc.i386
# sudo yum install libgtk-java.i586
```

Fedora 11 64-bit Host

Installer Requirements

Install the 32-bit libraries that are required in order to run the installer:

```
# sudo yum install glibc.i686
# sudo yum install libgtk-java.i586
```

Ubuntu 8.04 64-Bit Host

Installer Requirements

Install the 32-bit libraries that are required in order to run the installer:

```
$ sudo apt-get install ia32-libs
$ sudo apt-get install lib32nss-mdns
```

Ubuntu 9.04 64-Bit Host

Installer Requirements

Install the 32-bit libraries that are required in order to run the installer:

```
$ sudo apt-get install ia32-libs
$ sudo apt-get install lib32nss-mdns
```

3.3 Target System Requirements

The target is the processor for which you are developing. The Wind River Compiler is separately licensed for a variety of target architecture families. For a list of specific target CPUs supported by the tools, use the `dctrl` utility, or see the *Wind River Compiler User's Guide* for each architecture family.

4. Usage Caveats

This section describes usage issues that you should be aware of when working with the Wind River Compiler.

C and C++ Compilers

Treatment of Invalid Command-Line Options

In most cases, passing an unrecognized option flag to the tools generates a warning. This behavior, however, is not completely consistent. In some cases, no warning is generated; in other cases, such an invalid option may cause compilation to stop.

-Xmismatch-warning Overrides -e Option

-Xmismatch-warning and **-Xmismatch-warning=2** override the **-e** option, used for changing the severity of a message. If either form of **-Xmismatch-warning** is used, mismatched types will only produce a warning, even if **-e** is used to increase the severity level of the diagnostic.

#pragma weak Usage

Using **#pragma weak** may occasionally produce unforeseen behavior. Two instances are worth mentioning:

- **#pragma weak** is incompatible with local data area (LDA) allocation; using **#pragma weak** with **-Xlocal-data-area** or **-Xlocal-data-area-static-only** enabled will produce a warning and temporarily disable LDA.
- A global definition will override a weak definition when it is encountered. (A symbol may be defined in more than one module as long as the following are true:
 - No more than one of the definitions is global.
 - All the other definitions are weak.

Consider the following scenario. Function **foo()** uses *x*, which is declared weak in library 1 and global in library 2. If library 1 is searched first, the weak version of *x* will be used. On the other hand, if library 2 is subsequently linked (because, for example, another function uses it), then the global version of *x* will replace the weak version.

Using __attribute__((section)) May Not Always Be Honored

In some cases, the compiler may not honor an attempt to use the **section** attribute to place initialized data into a section intended for uninitialized data, and vice-versa.

For example, in the following code, the compiler does not honor the attempt to put *x* into the **.bss** section:

```
__attribute__((section(".bss"))) int x = 3;
```

x will be assigned to the **.data** section, not **.bss**.

Far Relative Addressing and VLE

The following applies to PowerPC code using the VLE (Variable Length Encoding) instruction set.

Programs compiled to use far (32-bit) relative addressing, either for code or data (for example, programs compiled with **-Xcode-far-relative** or **-Xdata-far-relative**), must explicitly reference the symbols **_SDA_BASE** and **_SDA2_BASE**. If these symbols are not referenced anywhere in the program, the linker will generate

incorrect code. Specifically, it will try to use “absolute SDA,” in which **r0** is used as a base register to indicate a base location of zero. (See the user’s guide section on ELF Relocation Information for more on absolute SDA.)

This is not a problem for non-VLE code, where **r0** is interpreted as zero; in contrast, in VLE mode, **r0** is interpreted as the *contents* of **r0**.

Limited Support for RTA Profiling

Only one compiler option for generating Run-Time Analysis profiling information is supported in VxWorks. That option is **-Xrtc**. Other compiler profiling options, such as **-Xprof**, are not supported in VxWorks.

For **-Xrtc** itself, all mask options are supported except *mask = 0x4*.

Linker

Warnings for Duplicate Definitions

When SDA is enabled, the compiler changes the linkage of some symbols from **COMMON** to **BSS** data. Because of this, the linker may issue a warning if a symbol (such as **int foo**) is defined in more than one place (e.g., **sysLib.c** and **usrConfig.c**), as may be the case with custom BSPs. Although such warnings may not indicate a significant problem, nonetheless it’s a good idea to declare the symbol **extern** (e.g., **extern int foo**) in the header file and define the symbol in one C file only.

5. Known Problems

This section highlights known problems with the Wind River Compiler that may have a significant impact on installation and your experience with this product. Defect numbers, where applicable, are given in parentheses after the description of the issue.

The most current information for this release is available from the Wind River Online Support Web site (see [1.5 Latest Release Information](#), p.4).

Casting in Inline Assembly Code

The compiler fails to sign-extend certain arguments after a cast in inline assembly routines. To fix this problem, use the **-Xold-inline-asm-casting** option.

Multiple **-#** Options

If two or three **-#** options are present on the command line, the compiler acts as if the **-##** or **-###** option is present, respectively. (**-##** displays subprogram invocation lines with arguments but does not execute any subprogram; **-###** is like **-##** but quotes arguments.)

C++ Trigonometric Functions Do Not Support Complex-Number Objects

Passing instances of the complex-number class template (defined in **complex.h**) to trigonometric functions such as **cos** leads to unresolved symbols.

Using `setjmp()` and `longjmp()` in Mixed ARM and Thumb Code

If you enable interworking (`-Xinterwork`) to compile mixed ARM and Thumb code, there is a limitation to the use of `setjmp()` and `longjmp()`. The `longjmp()` routine does not support switching between 32-bit and 16-bit mode; if you call `setjmp()` in Thumb mode and make a corresponding call to `longjmp()` in ARM mode, the result will be unpredictable. (However, the reverse procedure—calling `setjmp()` in ARM mode and `longjmp()` in Thumb mode—yields correct behavior.) A workaround is to define a routine like the `__common_long_jump()` shown below and compile it with `-Xinterwork` enabled.

```
void __common_long_jump()
{
    return;
}
```

Incorrect Type-extension of Arguments for 68K Targets

For 68K targets, when a character or short integer is passed to a function, the argument is normally extended to 32 bits. But if the function prototype is present and the parameter is explicitly declared in the prototype as a character or short integer, this extension should not occur. In cases where the prototype is available but appears in a different file from the function call, however, the compiler (incorrectly) performs the extension anyway.

6. Documentation Errata

For a detailed list of documentation errata for the Wind River Compiler, visit the Online Support Web site (see [1.5 Latest Release Information](#), p.4).

CUSTOMER SERVICES

Wind River is committed to meeting the needs of its customers. As part of that commitment, Wind River provides a variety of services, including training courses and contact with customer support engineers, along with a Web site containing the latest advisories, FAQ lists, known problem lists, and other information resources.

Customer Support

For customers holding a maintenance contract, Wind River offers direct contact with support engineers experienced in Wind River products. The Customer Support program is described in the *Standard Support User's Guide* available at:

www.windriver.com/support

The guide describes the services available, including assistance with installation problems, product software, documentation, and service errors.

You can reach Customer Support by e-mail or telephone:

Location	Phone	E-mail
North and South America, Asia/Pacific (outside Japan)	800-872-4977 (toll-free)	support@windriver.com
Europe, Africa, Middle East	+(00) 800-4977-4977 (toll-free)	support-EC@windriver.com
Japan	81-3-5778-6001	support-jp@windriver.com

For detailed contact information, including contact information specific to your products, see the Support Web site shown above.

Wind River Online Support

Wind River Customer Services also provides Wind River Online Support, an online service available under the Support Web site. This is a basic service to all Wind River customers and includes advisories, online manuals, and a list of training courses and schedules. For maintenance contract holders, Online Support also provides access to additional services, including known problems lists, patches, answers to frequently asked questions, and demo code.