WIND RIVER

Wind River®
Diab Compiler

RELEASE NOTES

5.9.2

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 Diab Compiler
Release Notes
5.9.2

13 Aug 12

Contents

Wind River Diab Compiler

RELEASE NOTES

5.9.2

1. Introduction

The Wind River Diab Compiler 5.9.2 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.

1.1 Installation and Licensing

For information on installing the Wind River Diab 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.

CAUTION: Installing both the standalone and bundled version of the Wind River Diab Compiler can result in unexpected results, e.g., older versions of some installed files may be overwritten over newer versions. If you choose to install both versions, be sure to install the tools in different locations.

Special Note: Installing on Linux Hosts

32-bit Run-Time Libraries

In order to install the Wind River Diab 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.

1.2 Latest Release Information

The latest information on this release can be found in the Wind River Diab 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.

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 compiler.

2. Changes in This Release

2.1 Enhancements

Supported Processors

For a current list of supported processors for your architecture, use the **dctrl** utility or see the *Wind River Diab Compiler User's Guide: Target Configuration*.

ARM

New support for the ARM architecture includes the following:

- ARM Unified Assembler Language support.
- Cortex M4 instructions enabled for MT2 targets.

In addition the following floating point support has been added for MT2 targets:

Code	Description
F	Single Hardware, Double Software Floating Point (for ARMT2, ARMV6MT2, ARMV7MT2, ARMV7AT2, ARMV7RT2 only)
G	All Single Hardware Floating Point

PowerPC

Support has been added for the following processors:

- PPCE200Z225N3V
- PPCE200Z420N3V
- PPCE200Z425N3V
- PPCE200Z0HN2PV
- PPCE200Z210N3V
- PPCE200Z720N3V
- PPCE200Z759N3V
- PXD10
- PXD20
- PXN20
- PXN21
- PXR40
- PXS20
- PXS30

TriCore

The following processors are now supported as part of TriCore ISA 1.3.1 support:

- TC1167

- TC1197
- TC1337
- TC1367
- TC1387
- TC1724
- TC1728
- TC1736
- TC1767
- TC1782
- TC1784
- TC1797
- TC27xAstep16P
- TC27xAstep16E
- TC2Dx16E
- TC2Dx16P
- TC16P
- TC16E
- TC26x16E
- TC26x16P
- TC131

TriCore Default Compiler Front End Changed

The default front end of the TriCore compiler has been changed from **ctoa** to **etoa**. The default compilation for this release therefore behaves as if invoked with **dcc -Xc-new**. The **ctoa** front end can be used by invoking **dcc -Xc-old**.

Tricore Alignment of long long and double

The alignment of **long long** and **double** data types has been changed from double-word-aligned (8 byte) to word-aligned (4 byte) in conformance with the *TriCore 32-bit Unified Processor Core Embedded Applications Binary Interface (EABI) User's Manual* (v2.3, Feb. 2007).

Note that TriCore code compiled with previous releases is not binary-compatible with code that is compiled with the 6.9.2 compiler.

TriCore SFR Header Files

To simplify the programming of low level software like device drivers and startup code, header files that define various structures are provided to simplify access to TriCore special function registers (SFRs). The files can be found in *versionDir***/include/sfr/**.

Square Root Instructions

The compiler generates the **efssqrt** instructions where appropriate.

Static Initialization of Flexible Length Arrays

The Wind River Diab Compiler supports static initialization of flexible array members when using C99.

Compiler Attributes and Keywords

use_frame_pointer attribute. Instructs the compiler to copy the stack pointer of the function that is marked to another register, and to perform all stack accesses through this register rather than using the actual stack pointer. 68K, ARM, ColdFire, MCORE, MIPS, PowerPC, SH, SPARC, and TriCore architectures only.

__bit keyword. Used for the **bit** data type. For all but the TriCore architecture, it is supported with the new compiler front end (**dplus** or **dcc** with **-Xc-new**); for TriCore it is supported by default.

Command-Line Options

The following are new or modified command-line options. For detailed information see the *Wind River Diab Compiler User's Guide*.

- **-H**. Print pathnames of header files.
- **-speed**. Optimize code for speed.
- **-speed-debug**. Optimize code for speed and enable debugging.
- **-size**. Optimize code for size.
- **-size-debug**. Optimize code for size and enable debugging.
- **-balanced**. Optimize code and balance between size and speed optimization.
- **-balanced-debug**. Optimize code and balance between size and speed optimization, and enable debugging.
- **-Xapu-lsp**. Enable LSP instructions and intrinsic functions. PowerPC architecture only.
- **-Xlsp-int**. Enable use of LSP instructions for integer data types. PowerPC architecture only.
- **-Xcompress-prolog-epilog**. Reduce size of function prologue and epilogue.
- **-Xreverse-inline**. Optimize code by moving some code to new functions. Now supported for PowerPC.
- **-Xdebug-mode 0x200** *mask* option. Do not generate debug information to describe source code labels.
- **-Xdiagnose-inline**. List all functions that have not been inlined.
- **-Xdiagnose-inline-verbose**. List all functions that have not been inlined and explain why they are not inlined.
- **-Xmetadata**. Activate or deactivate metadata generation.
- **-Xuser-metadata**. Add key-value pairs to the metadata of an object file.
- **-Xdump-metadata**. Display metadata found in any of the linked object files.
- **-Xdump-metadata-output-file**. Specify output file for **-Xdump-metadata** reports.
- **-Xsection-spli**t. New *split-val* options (4-7).

- **-Xsection-split-name**. Specify name pattern for split sections.
- **-Xgen-coldfire-header**. With the assembler, the ELF header of the output object file is **EM_COLDFIRE**. With the linker, the ELF header of the executable file is **EM_COLDFIRE** when the input object files and library files are all **EM_COLDFIRE** format (otherwise the header of the generated executable file is **EM_68K**). Without **-Xgen-coldfire-header** the ELF header of the files generated from the assembler and linker is **EM_68K(4)**, and not **EM_COLDFIRE (52)**, despite the fact that the target is Coldfire. Coldfire architecture only. See also *6. Documentation Errata*, p.11.
- **-Xno-builtin**. Disable Diab built-in functions (like **__diab_memset** and **__diab_memcpy**). See also *6. Documentation Errata*, p.11.
- **-Xdisable-intrinsic-mask**. Disable specific intrinsic functions. See also *6. Documentation Errata*, p.11.
- **-Xkeep-function-asm**. Generate function body if it contains an assembly string or calls an assembly macro. See also *6. Documentation Errata*, p.11.
- **-Xsmall-data-registers** and **-Xsmall-const-registers**. Support the automatic extension of small data and small constant areas. See also *6. Documentation Errata*, p.11.
- **-Xlegacy-c-attribute-aligned**. Use legacy aligned attribute implementation. See also *6. Documentation Errata*, p.11.
- **-Xefpu-spfs**. For a target with an embedded floating-point APU that only supports single precision floating point instructions (no double precision instructions and no vector float instructions). For PowerPC only. See also *6. Documentation Errata*, p.11.

Assembler

The intrinsic assembly feature that available for certain architectures is now supported for TriCore.

The following TriCore 1.6.x (AURIX) instructions are now supported:

- **cmpswap.w**
- **swapmsk.w**
- **crc32**
- **wait**

Linker

CLONE(*original*:*clone*). Instructs the linker to make a copy of the section that includes the symbol *original* and to include it in the link.

ALIAS(*reference*, *definition*). Instructs the linker to bind all references to the symbol *reference* in the specified input section(s) to the symbol *definition*.

Configuration Files

remove statement. Used in the **user.conf** file to suppress warnings from Diab tools when they encounter a user-specified option.

ddump Utility

The **ddump -m** flag generates a metadata report for all provided object files, as the linker would do when linking the object files.

The **ddump -Td** flag is similar to **-T**, but removes only debug information (ELF only).

Multi-Channel Sequencer Support

The compiler toolchain supports building code for Multi Channel Sequencer (MCS) sub-modules of the Bosch GTM-IP v1.4 module (for Tricore and PowerPC architectures).

Eclipse CDT Integration

Wind River provides a Diab compiler plugin for the Eclipse CDT, which is available on your installation media. The plugin allows you to use the Wind River Diab compiler within the CDT environment, including a graphical interface for changing toolchain settings. See also *Documentation*, p.7 and *6. Documentation Errata*, p.11.

Documentation

The *Wind River Diab Compiler Plugin for Eclipse CDT Getting Started* has been added to the documentation set, as well as the **dmake** command reference.

The *Wind River Diab Compiler User's Guide* and the *Wind River Diab Compiler Error Message Reference* have been updated for new features and enhanced for existing features.

Note in particular the following new material in the user's guide:

- *SDA Optimization* in chapter 10. See also *6. Documentation Errata*, p.11.
- *Creating a System Image with Multi-Channel Sequencer Code* in chapter 15.
- *Rebuilding Libraries* in chapter 32.
- The example for removing unused sections in the *-Xunused-sections* section of appendix D.

Documentation is available in PDF and HTML form in the **docs/extensions/eclipse/plugins** directory under the top-level installation directory.

2.2 Fixed Problems

For a list of problems fixed in the Wind River Diab Compiler, visit the Online Support Web site (see *1.2 Latest Release Information*, p.2).

2.3 Unsupported Features

For the ARM Thumb architecture, formal testing is performed with the Thumb 2 architecture—it is no longer performed with Thumb 1. For more information on Diab compiler support for the Thumb architecture, contact Customer Support.

-Xcomdat-off has not been supported since 5.9.1. The new C++ ABI requires COMDAT support to correctly implement the semantics of the language.

3. System Requirements

This section lists the minimum requirements for the Wind River Diab 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 Diab Compiler in the standard configuration.

These system requirements are for the Wind River Diab 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, x86-32/64
- 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 *1.2 Latest Release Information*, p.2.)

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 *1.2 Latest Release Information*, p.2.)

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 Diab Compiler.

- One of the following:
  - RHEL WS 4,5,6 32/64-bit.
  - SUSE Linux 11,12 32/64-bit.
  - Fedora 9 - 17 32/64-bit.
  - Ubuntu 8-12 32/64-bit.
- 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 *1.2 Latest Release Information*, p.2.)
- 32-bit run-time libraries.

NOTE: In order to install the Wind River Diab 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 Diab Compiler. For example:

Fedora 9 64-Bit Host

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

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

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

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 Diab 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

Wind River Diab Compiler Plugin for Eclipse CDT

The Wind River Diab Compiler Plugin for Eclipse CDT has been tested with the Eclipse Indigo release. The *Wind River Diab Compiler Plugin for Eclipse CDT Getting Started* document can also be used with the Juno release, as long as the **C/C++ Projects** view is selected (from the menu bar, select **Windows > Show View > C/C++ Projects**).

C and C++ Compilers

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.

5. Known Problems

For a complete list of known problems in the Wind River Diab Compiler, visit the Online Support Web site (see *1.2 Latest Release Information*, p.2).

6. Documentation Errata

For a detailed list of documentation errata for the Wind River Diab Compiler, visit the Online Support Web site (see *1.2 Latest Release Information*, p.2).

Wind River Diab Compiler Plugin for Eclipse CDT Getting Started

The Wind River Diab Compiler Plugin for Eclipse CDT has been tested with the Eclipse Indigo release. The *Wind River Diab Compiler Plugin for Eclipse CDT Getting Started* can also be used with the Juno release, as long as the **C/C++ Projects** view is selected (from the menu bar, select **Windows > Show View > C/C++ Projects**).

Wind River Diab Compiler User's Guide: SDA Optimization

For 5.9.1, SDA optimization was enabled by default when whole-program optimization (WPO) was enabled. For 5.9.2, SDA optimization must be enabled explicitly.

For 5.9.2, the introductory paragraph for the *SDA Optimization* section is incorrect. It includes the following sentences (ellipses are used here for abbreviation):

> SDA optimization takes place as an optional part of whole-program optimization... Enabling WPO enables SDA optimization, but SDA optimization can be turned off independently with the **-Xdisable-SDA** switch...

It should instead say:

> SDA optimization is an optional part of whole-program optimization (WPO). It is not, however, enabled by default when WPO is enabled.
>
> To enable SDA optimization, set bit **0x100** in the **-Xwhole-program-optim** flag. For example use:
>
> **`-Xwhole-program-optim=0x101`**
>
> SDA optimization can be disabled for particular classes of small data area with the **-Xdisable-SDA** switch.
>
> Note that the SDA optimization is incompatible with the link-time object-file cache (mask bit **0x8**), and is automatically disabled if the cache is enabled.

Wind River Diab Compiler User's Guide: Command-Line Options

The following command-line options are missing from the *Wind River Diab Compiler User's Guide*.

- **-Xgen-coldfire-header**. Used with the assembler, the ELF header of the output object file is **EM_COLDFIRE**. With the linker, the ELF header of the executable file is **EM_COLDFIRE** if the input object files and library files are all **EM_COLDFIRE** format (otherwise the header of the generated executable file is **EM_68K**). Without **-Xgen-coldfire-header** the ELF header of the files generated from the assembler and linker is **EM_68K(4)**, and not **EM_COLDFIRE (52)**, despite the fact that the target is Coldfire. Coldfire architecture only.

- **-Xno-builtin**. Disable Diab built-in functions (like **__diab_memset** and **__diab_memcpy**).

  If the optimizer encounters a loop like this:

```
for (int j = 0; j < 128; j++) {
 for (int i = 0; i < 128; i++) {
                a[j][i] = 0;
 }
}
```

  It might transform it to the following:

```
    __diab_memset(a, 0, 128*128*sizeof(a[0][0])).
```

Use **-Xno-builtin** to disable this kind of transformation.

- **-Xbss-zero-init-const-on**. Move **const** variables initialized to zero to BSS.
- -**Xbss-zero-init-const-off**. Do not move **const** variables initialized to zero to BSS.
- **-Xdisable-intrinsic-mask**. Disable specific intrinsic functions.
- **-Xkeep-function-asm**. Generate function body if it contains an assembly string or calls an assembly macro.

  Normally, static functions that are not called and do not have their address taken are deleted by the optimizer. This option prohibits that action for any function that has an assembly string or that "calls" an assembly macro in its body.

- **-Xsmall-data-registers** and **-Xsmall-const-registers**. Use these options to support the automatic extension of small data and small constant areas to more than one register on targets that support this feature.

  -Xsmall-data-registers=*n*
  -Xsmall-const-registers=*m*

  Targets that support small data and small constant areas have a register that is is reserved by the ABI to support referencing small variables and small constants using base-register-plus-offset addressing modes. This can provide greater efficiency and smaller code size. In some cases, these areas may not be big enough to hold all of the application's data, which requires the user to take some sort of action to resolve the problem. One solution is to manually place some variables in different sections, and to reserve a register globally across the application, thus reducing the number of variables in the small areas to fit the size limitations.

  The **-Xsmall-data-registers=***n* and **-Xsmall-const-registers=***m* options tell the compiler and the rest of the toolchain to automatically extend the number of registers reserved to *n* and *m*, respectively (where *n* and *m* would be greater than 1). For example, at link time the small constant area may overflow, and the user may decide to use two registers to fix the problem by specifying **-Xsmall-const-registers=2**. Once the entire application is recompiled with this option, the linker can link the application correctly and safely. Note, however, that by reserving more registers, fewer variables are free for use and this could have a negative impact on performance and size.

  The use of these options changes the effective ABI of the target. Because this happens automatically (and by intention) to simplify the process for the user, the user does not control which variables are referenced by which registers. Any variable placed in the small data and small constant areas may be accessed through the original ABI register pair, or by other registers, as part of an implied *special ABI extension*.

  To make sure that the registers that are re-purposed as extended small area registers are not arbitrarily used for multiple purposes by the compiler, you must build the entire application—including library code— with the same **-Xsmall-data-registers=***n* and **-Xsmall-const-registers=***m* settings. The linker generates warnings (by default) when the ABIs of the object files are inconsistent and ignoring these warnings may result in applications misbehaving during run-time. Note that the **-Xdisable-all-warnings** causes the linker to ignore warnings about this potentially serious ABI incoherence.

To understand why the same **-Xsmall-data-registers=***n* and **-Xsmall-const-registers=***m* settings must be used for all of the application code (including libraries), consider the case in which two small data area registers are used: the reserved register **Rs** and an extension register **Rx,** which is normally free and used for other purposes, for some object files. Also assume that other object files are built using one small area register, and that the variable **V** goes into the small data area and is used by all of the object files. If **V** is referenced through **Rs**, the application may appear to be working correctly. However, if **V** is referenced through **Rx**, the object files that reserve **Rx** are able to access **V**, but the other object files do not reserve **Rx**, and are free to overwrite it and use it for other purposes. The code in these object files can change the value of **Rx** freely. The linker also uses **Rx** freely to access **V** (and any other variables in the extended small data area).

- **-Xlegacy-c-attribute-aligned**. Use legacy aligned attribute implementation.
- **-Xefpu-spfs**. For a target with an embedded floating-point APU that only supports single precision floating point instructions (no double precision instructions and no vector float instructions). For PowerPC only.
- **-Xdiagnose-inline-verbose**. List all functions that have not been inlined and explain why they are not inlined.

Wind River Diab Compiler User’s Guide: Appendix A Target Configuration

The following target configuration options are missing from the guide:

PowerPC

- PPCE200Z759N3V

TriCore

- TC27xAstep16P
- TC27xAstep16E
- TC2Dx16E
- TC2Dx16P
- TC16P
- TC16E
- TC26x16E
- TC26x16P
- TC131

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.