

# Wind River® Workbench On-Chip Debugging

RELEASE NOTES

3.2

---

Copyright © 2010 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](http://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](http://www.windriver.com)

For information on how to contact Customer Support, see:

[www.windriver.com/support](http://www.windriver.com/support)

*Wind River Workbench On-Chip Debugging  
Release Notes  
3.2*

27 Jan 10

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Overview .....	1
1.1.1	Features .....	1
1.1.2	Configuration .....	2
1.1.3	Latest Release Information .....	2
<b>2</b>	<b>Changes in This Release .....</b>	<b>5</b>
2.1	Introduction .....	5
2.2	New Features .....	5
2.2.1	Intel Architecture Support .....	5
2.2.2	Wind River Connect 2 .....	6
2.2.3	VxWorks 653 Platform .....	6
2.3	Enhancements .....	6
2.3.1	Flash Programming Enhancements .....	6
	Flash Library .....	6
2.3.2	Expressions View Enhancements .....	7
2.4	New Processor Support .....	7
2.5	Fixed Problems .....	8
<b>3</b>	<b>System Requirements .....</b>	<b>11</b>
3.1	System Requirements .....	11
3.1.1	Host System Requirements .....	11
	Windows Hosts: .....	11
	Solaris Hosts: .....	12
	Linux Hosts: .....	12
3.1.2	Target System Requirements .....	13

3.2	Supported Hardware .....	14
<b>4</b>	<b>VxWorks 653 Platform .....</b>	<b>15</b>
4.1	Introduction .....	15
4.2	Usage Caveats .....	15
<b>5</b>	<b>Wind River Workbench .....</b>	<b>17</b>
5.1	Overview .....	17
5.2	Changes in This Release .....	17
5.2.1	Enhancements .....	18
	Wind River Workbench .....	18
	Wind River System Viewer .....	21
	Wind River Run-Time Analysis Tools .....	22
5.2.2	Fixed Problems .....	22
5.2.3	Unsupported Features .....	23
5.3	Usage Caveats .....	23
	Workbench Issues .....	23
	Project and Build Issues .....	30
	Wind River Run-Time Analysis Tools .....	30
	Windows Host-Related Issues .....	31
	Linux Host-Related Issues .....	31
	Solaris Host-Related Issues .....	32
5.4	Known Problems .....	32
	Workbench Issues .....	32
	Project Issues .....	36
	Remote Systems View Issues .....	36
	Build Issues .....	37
	Build Properties Issues .....	37
	Debugging Issues .....	38
	Indexer Issues .....	40
	Properties View Issues .....	40
	Run-Time Analysis Tools Issues .....	41
	System Viewer Issues .....	41
<b>6</b>	<b>Known Issues .....</b>	<b>43</b>
6.1	Usage Caveats .....	43
6.2	Known Problems .....	49
6.2.1	Firmware-Specific Issues .....	49
6.2.2	Wind River ICE 2 Issues .....	50
6.2.3	Wind River Trace Issues .....	50
6.2.4	Wind River Trace 2 Issues .....	51

6.2.5	OCD Issues .....	52
6.3	Documentation Errata .....	55
<b>7</b>	<b>Processor Support Matrix .....</b>	<b>57</b>
7.1	Introduction .....	57
7.2	PowerPC Processors .....	58
7.3	QorIQ Processors .....	61
7.4	ColdFire Processors .....	61
7.5	MIPS Processors .....	62
7.6	ARM Processors .....	63
7.7	XScale Processors .....	64
7.8	Cavium Processors .....	64
7.9	PA Semi Processors .....	65
7.10	SiByte Processors .....	65
7.11	RMI Processors .....	65
7.12	Intel Processors .....	65
<b>8</b>	<b>Uninstalling .....</b>	<b>67</b>



# 1

## *Introduction*

### 1.1 Overview

Wind River Workbench On-Chip Debugging provides the Wind River ICE SX, Wind River ICE 2, and Wind River Probe emulators, the Wind River Trace and Wind River Trace 2 tools, and the Wind River Workbench development suite. Together, these products provide a fully integrated hardware and software solution that you can use for board bring-up, flash programming, production, and testing.

Wind River emulators allow developers to perform source-level debug activities such as watching memory and controlling large numbers of registers. They allow users to shorten development cycles, even for extremely complex applications.

Wind River emulators let you control a target by utilizing the on-chip debugging (OCD) services embedded in the microprocessor of that target. An emulator operates effectively as a stand alone system, communicating with the OCD services resident in the microcode of the chip.

When you access the OCD services in the chip, you gain complete control of the microprocessor, and all interaction between the emulator and the target runs exclusively through the OCD connection. This means that the emulation system is effective for the entire development process, even before board-level peripherals are stable.

#### 1.1.1 Features

Wind River Workbench for On-Chip Debugging includes the following features:

##### **High-Performance JTAG**

Eliminates slow download times and run control when developing with On-Chip Debugger (OCD) microprocessors. Performance is improved due to hardware logic that caches common JTAG scan chains.

##### **On-Chip Debug Target Control**

Allows users to start and stop the target, set internal hardware and software breakpoints, take a target snapshot, reset the target, step one statement or instruction into function calls, and step over or out of a function.

### **Built-In Hardware Diagnostics**

Wind River Workbench includes a comprehensive suite of RAM tests, scope loops, and CRC tests.

### **Additional Custom Registers**

Wind River emulators support 32 custom register groups. For the Wind River Probe, some architectures support up to 5000 custom registers. For the Wind River ICE SX, most architectures support 1600 custom registers; for the Wind River ICE 2, most architectures support 960 custom registers.

### **Target Versatility**

Allows the user to test several target architectures using the same debug hardware.

### **JTAG Server**

JTAG Server provides the ability to control and manipulate multiple devices on a single scan chain ring. JTAG Server is a single development tool that can access any device on the scan chain ring. FPGAs, EPLDs and other programmable devices can be loaded, eliminating the need for multiple device programmers and emulation hardware.

## **1.1.2 Configuration**

Tool Configuration information is available in the following books, available in HTML format from the Workbench toolbar by selecting **Help > Help Contents > Wind River Documentation**:

- *Wind River ICE SX for Wind River Workbench Hardware Reference*
- *Wind River ICE 2 for Wind River Workbench Hardware Reference*
- *Wind River Probe for Wind River Workbench Hardware Reference*
- *Wind River Trace for Wind River Workbench Hardware Reference.*
- *Wind River Trace 2 for Wind River Workbench Hardware Reference.*

A full description of all on-chip debugging configuration options is available in the *Wind River Workbench On-Chip Debugging Configuration Options Reference*.

A full description of all on-chip debugging run control commands is available in the *Wind River Workbench On-Chip Debugging Command Reference*.

You must install USB drivers to use the Wind River Probe. For information on how to do this, see the *Wind River Probe for Wind River Workbench Hardware Reference: Installing Drivers*.

## **1.1.3 Latest Release Information**

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

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

In addition, this site includes links to topics such as known 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.71.

In addition, a list of known issues and other important information is installed in *installDir/readme.txt*. Be sure to review this file before using Wind River Workbench for On-Chip Debugging.



# 2

## *Changes in This Release*

2.1	Introduction	5
2.2	New Features	5
2.3	Enhancements	6
2.4	New Processor Support	7
2.5	Fixed Problems	8

### 2.1 Introduction

This section describes changes in Wind River Workbench for On-Chip Debugging since the 3.1 release.

### 2.2 New Features

#### 2.2.1 Intel Architecture Support

This release includes support for the Intel Atom, Core 2 Duo, and Nehalem Xeon 55xx architectures.

There is no difference in the workflow of Wind River Workbench On-Chip Debugging between Intel processors and other architectures. However, due to electric-potential considerations, there are some hardware differences in the following tools.

##### **Wind River Probe**

Intel targets can have three different kinds of debug ports. The most common is the extended debug port (XDP). The Probe uses an XDP adapter for this kind of port. Some Intel targets use second-side attach (SSA) or zero insertion force (ZIF) ports.

For these kinds of ports the Probe includes attachments for the XDP adapter. See the *Wind River Probe for Wind River Workbench Hardware Reference* for details on all three adapters.

The Probe draws its power from its USB cable. Due to the power demand from the XDP adapter, one USB port does not provide enough current. For Intel targets, Wind River supplies a double-ended USB cable with two type A connectors and one mini-B connector. Attach both USB-A connectors to USB ports on your host.



---

**NOTE:** Both USB-A connectors must be attached to powered USB ports. If you connect them to ports on a USB hub, which in turn is connected to only one USB port on your host, that will not supply enough current.

---

## Wind River ICE 2

Intel targets can have three different kinds of debug ports. The most common is the extended debug port (XDP). The ICE 2 uses an XDP adapter for this kind of port. Some Intel targets use second-side attach (SSA) or zero insertion force (ZIF) ports. For these kinds of ports the ICE 2 includes attachments for the XDP adapter. See the *Wind River ICE 2 for Wind River Workbench Hardware Reference* for details on all three adapters.

## Wind River ICE SX

The Wind River ICE SX does not support Intel processors.

## 2.2.2 Wind River Connect 2

This release includes support for the new Wind River Connect 2 tool. The Connect 2 is an optional extension for the Wind River ICE 2 that allows you to connect the Wind River ICE 2 to up to four targets at the same time. For information, see the *Wind River ICE 2 Hardware Reference: Wind River Connect 2*.

## 2.2.3 VxWorks 653 Platform

The 2.2 release of Wind River VxWorks 653 Platform includes support for Wind River Workbench On-Chip Debugging. See [4. VxWorks 653 Platform](#).

## 2.3 Enhancements

### 2.3.1 Flash Programming Enhancements

#### Flash Library

Combined flash drivers added since the 3.1.1 release:

- AMD S29GL01GP (65536 x 16) 2 Devices

- Intel 28F512P33B (32768 x 16) 1 Device
- SPANSION S29GL064N02 (8192 x 8) 1 Device
- NUMONYX 28F256M29EW (16384 x 16) 2 Devices

### 2.3.2 Expressions View Enhancements

The Expressions view now allows you to assign an expression to a given debug context (so the expression's displayed value will not change when you switch debug contexts.) You can also now break up very large arrays into subsets, to speed display time. See the *Wind River Workbench On-Chip Debugging User Tutorials: Using the Expressions View*.

## 2.4 New Processor Support

Workbench now includes support for the following processors:

#### PowerPC 56xx

- SPC560B50
- SPC560S60
- MPC5606S
- MPC5607B
- MPC5607S

#### PowerPC 85xx

- MPC8569

#### PowerPC 86xx

- MPC8640
- MPC8640D

#### Freescale QorIQ

- P2010
- P4080

#### MIPS32

- Wintegra WinPath2
- RMI AU1300
- Altera MP32 (Nevada)

#### ARM9

- Freescale i.MX25
- CNS1102
- CNS1104
- CNS1105
- CNS2131

- CNS2132
- CNS2133
- CNS2181
- CNS2182

#### ARM Cortex

- Freescale i.MX51

#### Intel Atom

- 330

#### Intel Core Duo

- T9400
- L7400

#### Intel Nehalem

- Xeon 55xx

## 2.5 Fixed Problems

This section lists problems from previous Workbench releases that have been fixed in this release.

- On ColdFire 5329 targets running VxWorks, setting a breakpoint no longer generates exceptions in VxWorks when cache is enabled. (WIND00145520)
- When you use the Convert utility to convert an \*.elf file to an \*.elfc file, C++ types and values are now displayed correctly in the Variables view. (WIND00111014)
- On PowerPC 8572 targets, you can now erase an area of flash larger than 6MB without breaking up the area into several smaller areas and erasing them separately. (WIND00167108)
- The Flash Programmer view now successfully adds flat (headerless) binary files. (WIND00172006)
- On Texas Instruments OMAP3410 targets, when the **Run on Target** checkbox in the Hardware Diagnostics view is selected, the Single RAM Test now returns correct results. (WIND00164031)
- For MPC8349 targets, the bit-level field details in the Registers view now match the current revision values. (WIND00131076)
- Errors have been removed from the MPC8349 register file. (WIND00162404)
- Workbench now supports L2 cache for the AMCC 460EX processor. (WIND00161630)
- On Texas Instruments OMAP34xx and 35xx targets, you can now reset the target with an IN or INN command after a bus error. (WIND00171620)

- On ARMX (XScale) targets, you can now single step the core from address 0x0. (WIND00153442)
- For little-endian OMAP3410 targets, the compare memory operation in the Binary Upload view now succeeds for 16-bit and 32-bit as well as for 8-bit. (WIND00163964)
-



# 3

## *System Requirements*

[3.1 System Requirements](#) 11

[3.2 Supported Hardware](#) 14

### **3.1 System Requirements**

#### **3.1.1 Host System Requirements**

The host is the computer on which you do your development work. This section lists the minimum requirements for running Wind River Workbench On-Chip Debugging in the standard configuration, in which the host and target are separate computers.

System requirements for the Wind River Probe include at least 500MB of disk drive space, and a USB 1.1 or 2.0 port (1.1 minimum; Wind River recommends 2.0).

#### **Windows Hosts:**

- One of the following versions of Microsoft Windows:
  - Windows XP Professional (Service Pack 2 or 3) 32-bit
  - Windows Vista Business (Service Pack 2) 32-bit
  - Windows Vista Enterprise (Service Pack 2) 32-bit
  - Windows 7 32-bit
- Administrator privileges.
- 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.
- At least 1 GB of RAM for normal development; 2 GB of RAM is recommended for larger projects.

- 3.8 GB of disk space (for a typical installation), including a minimum of 40 MB in the user's home directory. 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 Internet browser.
- A network interface card for debugging over a network (recommended).
- TCP/IP must be installed on the host system, even if it is being used as a stand-alone PC.

Wind River recommends that you have an active Internet connection during initial installation in order to access patches, documentation, and other important information from the Wind River Online Support Web site.

### Solaris Hosts:



---

**NOTE:** Solaris is not supported for the Wind River Probe.

---

- Sun Solaris 10 (with GTK), Update 11/06, Sparc 32-bit
- A monitor capable of displaying 1024 x 768 @ 16 bpp or better.
- A Blade 150 workstation with a 500MHz processor, or a workstation with higher performance.
- At least 1 GB of RAM; 2 GB of RAM is recommended for larger projects.
- 3.8 GB of disk space (for a typical installation), including a minimum of 40 MB in the user's home directory. When calculating the amount of disk space needed, be sure to reserve space for your own applications and development.
- A current version of a standards-compliant Internet browser.
- A local DVD-ROM drive or access to network for installation.
- CDE Window Manager recommended.
- TCP/IP must be installed on the host system, even if it is being used as a stand-alone PC.

Wind River recommends that you have an active Internet connection during initial installation in order to access patches, documentation, and other important information from the Wind River Online Support Web site.

### Linux Hosts:

- One of the following Linux distributions:
  - Red Hat Enterprise Linux Workstation 4 (Update 6 or 8), x86 32-bit
  - Red Hat Enterprise Linux Desktop (with Workstation Option) 5, Update 2, x86 32-bit
  - Red Hat Enterprise Linux Desktop (with Workstation Option) 5, Update 3, x86 32-bit and x86 64-bit
  - Fedora 9, x86 64-bit

- Fedora 11, x86 32-bit and x86 64-bit
- openSUSE 11.1, x86 32-bit and x86 64-bit
- Novell SUSE Linux Enterprise Desktop 10 (Service Pack 2), x86 32-bit and x86 64-bit
- Novell SUSE Linux Enterprise Desktop 11, x86 32-bit and x86 64-bit
- Ubuntu Desktop 8.04 Update 4, x86 64-bit
- Ubuntu Desktop 9.04, x86 32-bit and x86 64-bit
- Root privileges.
- A monitor capable of displaying 1024 x 768 @ 16 bpp or better.
- Intel Pentium 4 class computer with a 1 GHz processor, or a computer with higher performance.
- At least 1 GB of RAM for normal development; 2 GB of RAM is recommended for larger projects.
- A DVD drive or access to a network for installation.
- GNOME Window Manager.
- 3.8 GB of disk space (for a typical installation), including a minimum of 40 MB in the user's home directory. 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, even if it is being used as a stand-alone PC.
- A network interface card for debugging over a network (recommended).
- A current version of a standards-compliant Internet browser.

The Wind River Probe USB driver must be rebuilt for the kernel being used; therefore the kernel source is required. If the kernel source is unavailable from the installation CD, you can obtain it from:

<http://rpmfind.net/linux/RPM>

Also, the **rpmbuild** and **gcc** packages must be installed on the system.

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

### 3.1.2 Target System Requirements

The target is the computer for which you are developing. This section lists the minimum requirements for running Wind River Workbench On-Chip Debugging in the standard configuration, in which the host and target are separate computers.

Your system should include a supported target board with the following:

- 8 MB of RAM (more if using a RAMdisk.) All supported reference boards from Wind River include at least 8 MB of memory.
- Space for a file system on hard disk, RAMdisk, flash memory, or floppy disk.

- An Ethernet cable or null crossover cable (needed for initial board setup.)
- An RS-232 null modem cable (needed for initial board setup.)
- A floppy disk drive or network installation card for the installation of a file system.
- A keyboard (recommended for configuration if using network booting.)
- A monitor (recommended for configuration if using network booting.)

## 3.2 Supported Hardware

The following architectures can be used with Wind River Workbench for On-Chip Debugging.

- AMCC 40x (formerly IBM 40x)
- AMCC 44x (formerly IBM 44x)
- PowerPC 5xx
- PowerPC 5xxx
- PowerPC 55xx
- PowerPC 6xx
- PowerPC 7xx
- PowerPC 74xx
- PowerPC 8xx
- PowerPC 82xx
- PowerPC 83xx
- PowerPC 85xx
- PowerPC 86xx
- Freescale ColdFire (formerly Motorola ColdFire)
- ARM9, ARM11, ARM Cortex
- ARMX (IOP, IXP and PXA processors)
- MIPS32
- MIPS64
- SiByte
- PA Semi
- Cavium
- RMI
- Intel

# 4

## VxWorks 653 Platform

[4.1 Introduction](#) 15

[4.2 Usage Caveats](#) 15

### 4.1 Introduction

The 2.2 release of Wind River VxWorks 653 Platform includes Wind River Workbench On-Chip Debugging. This chapter describes issues that VxWorks 653 Platform users new to Workbench On-Chip Debugging (OCD) should be aware of.

### 4.2 Usage Caveats

- VxWorks 653 does not support all of the target boards available for Workbench for OCD. For instance, the OCD documentation set gives many examples for a PowerPC 750FX target, which VxWorks 653 does not support.
- Path names given in the OCD documentation are specific to VxWorks. Many of them are different for VxWorks 653. For instance, the documentation gives the path to the VxWorks boot ROM. However, the pre-built VxWorks 653 boot ROMs are located in your install directory under **vxworks653-2.x/target/proj/yourTarget\_bootApp/buildSpec\_romCompressed**.
- Instead of **bootrom.bin** or **bootrom.hex**, VxWorks 653 boot ROM files are named **bootApp\_romCompressed.bin** or **bootApp\_romCompressed.hex**.
- Throughout the OCD documentation set, the tutorials given for creating projects will not work with VxWorks 653.
- Wind River board support packages (BSPs) for VxWorks record board-specific information (such as switch and jumper settings) in files named **target.ref**. BSPs for VxWorks 653 record the same information in files named **target.nr**. These files are located in your install directory under

`vxworks653-2.x/target/config/yourTarget/target.nr`. You can also visit <http://www.windriver.com/support> for online help for your BSP.

- VxWorks 653 Platform provides OCD support only for PPC750GX and PPC7457.
- VxWorks 653 Platform does not support standalone projects.
- VxWorks 653 Platform does not support the diab compiler.
- VxWorks 653 Platform provides only kernel-level and single-partition OCD support.
- Most of the examples given in the OCD documentation set are specific to VxWorks 6.x. Some steps are different for VxWorks 653. For example, if a tutorial prompts you to select **Wind River VxWorks 6.x Target Server Connection** in the New Connection Wizard, you should instead select the appropriate **VxWorks 653 Target Server Connection**.

# 5

## *Wind River Workbench*

- 5.1 Overview 17
- 5.2 Changes in This Release 17
- 5.3 Usage Caveats 23
- 5.3 Usage Caveats 23
- 5.3 Usage Caveats 23
- 5.4 Known Problems 32

### 5.1 Overview

This chapter describes changes, new features, and known issues with Wind River Workbench that are not specific to Wind River Workbench for On-Chip Debugging.

Wind River Workbench is an Eclipse-based development suite that facilitates creating and building projects, establishing and managing host-target communications, and developing, debugging, and monitoring operating system kernels as well as device software applications running on a simulator or a target.

Workbench is the successor to Tornado, SNIFF+, visionCLICK, SingleStep, and WIND POWER IDE. It incorporates many capabilities that formerly required external tools, such as target management and object browsing.

For tutorials demonstrating many Workbench features, see the *Wind River Workbench User's Guide* and *Wind River Workbench By Example*.

### 5.2 Changes in This Release

This release of Workbench is based on the Eclipse 3.4 platform release, and supports the following versions of VxWorks:

Table 5-1 VxWorks Versions

VxWorks Version	Service Pack	Notes
5.5		PowerPC and x86 only.
6.3	SP2	
6.4	SP1	
6.5		
6.6	SP1	
6.7		

This release also supports Wind River Linux 2.0, 2.0.2, and 3.0.

## 5.2.1 Enhancements

Workbench includes the following enhancements.

### Wind River Workbench

The Host Shell, Workbench build infrastructure, and BSP validation test suite now use Tcl 8.5 instead of Tcl 8.4, thereby providing increased speed and support for Windows.

### Project and Build Systems

#### VxWorks Source Build Within Workbench

Workbench provides a new project type that allows you to configure and build custom VxWorks libraries, which you can then use as the basis for VxWorks image projects.

#### VxWorks Image Projects Support Linked Resources

VxWorks image projects now support linked resources, so the Project Explorer now displays the complete list of files that make up the project.

#### VxWorks Kernel Configuration Editor Improvements

The Bundles tab of the Kernel Configuration Editor now displays the components in a selected bundle. You can navigate to an individual component in the component tree from the Bundle Component pane's context menu.

#### Quilt Patch Manager

The new default model, called the *Quilt patching model*, is based on **patch**, but provides an additional, rich set of patching features directed to exploring, merging, and importing patches. These features include pushing new patches to go forward, popping patches to go backward, capturing development changes into new patches, deriving patch annotation data, and more. The Quilt patching model

is the default for both command-line and Workbench-configured projects as it provides many advanced patching features.

### **QEMU Target Emulation**

QEMU, an open source machine emulator, allows you to configure the kernel and root file system for emulated boards for which Wind River has created board support packages (BSPs), such as:

- arm\_versatile\_926ejs
- common\_pc
- common\_pc\_64
- mti\_malta32\_74k\_be
- mti\_malta32\_74k\_le
- mti\_malta32\_be
- mti\_malta32\_le
- mti\_malta32f\_74k\_be
- mti\_malta32f\_74k\_le
- mti\_malta64f\_be
- mti\_malta64f\_le
- qemu\_ppc32
- si\_is8620

See the Wind River Online Support site for a definitive list.

QEMU loads the kernel found in the project build directory's **export** directory and NFS mounts the filesystem within the **export/dist** directory. (Unlike configuring hardware targets, for which you can place the kernel and filesystem in any location on your host system to be NFS mounted, QEMU requires specific locations.)

## **Debugger**

### **Debugger Enhancements**

#### **Dynamic printf Available**

In this release, Workbench provides support for adding dynamic **printf** commands using the Workbench Editor, the host shell, and the VxWorks kernel shell. This allows you to insert **printf** commands in your code without recompiling and relinking your application. Dynamic **printf** commands also generate an event in the System Viewer log.

#### **Integration of Wind River Debugger with Emacs**

It is now possible to access the Wind River debugger from Emacs, using the host shell's gdb mode.

#### **Support for MIPC Target Connections**

Workbench now includes MIPC as a possible WDB proxy connection type, making it possible to debug asymmetric multiprocessing (AMP) systems over a MIPC connection.

Workbench also supports simulation and debugging of an AMP system using multiple VxWorks simulator instances with MIPC connections between them.

### **Support for MIPS34k Additional Instructions**

Workbench supports the MIPS 34k extensions for supporting hardware threads.

### **Enhanced Processor Autodetection**

The capacity for Workbench to automatically detect the CPU on the target board has been improved.

### **Debugging Tools Can Now Coexist**

On-Chip Debugging tools and WDB agent-based debugging tools are no longer mutually exclusive. However, it is still the case that you should not place hardware breakpoints using both tools. This may lead to unpredictable debugging behavior, since both facilities access hardware breakpoint registers.

### **Processes May Run on Linux Targets Without Debugger Control**

When you run a process on the target using the **Run** button or its equivalent (instead of **Debug**), the process runs detached from the debugger. You can spawn any kind of process, including those having the sticky bit.

## **Launch Configuration**

Workbench is better at detecting when an existing launch configuration's settings match the program you are trying to launch, and presents a dialog that allows you to specify whether to use the existing launch, to modify it, or to create an entirely new launch configuration.

## **Source Analysis**

You can now specify the active, default, or fixed build spec to be used by the indexer when parsing a project.

## **Search Improvements**

Improvements to the Search view include highlighting of matched search terms, coloring of their line numbers, and display of the number of matches found (if greater than one).

## **Views**

### **Editor**

You can now select either Wind River or default Eclipse code and file templates when creating a file using **File > New > File from Template**. You can modify these templates from the New File wizard as necessary.

Macro expansion is available by hovering the cursor over a macro reference in the code. In addition, you can step through individual steps of a macro expansion using keyboard shortcuts.

### **Outline**

The Outline view is now populated when editing assembly files.

## Documentation

The *Wind River Workbench User's Guide* is now target platform-agnostic, and contains information useful to all Wind River Workbench users. For VxWorks-specific information, see *Wind River Workbench By Example*.

## Wind River System Viewer

### Linked Views

*Linked Views* are a versatile replacement/extension of the **Analysis Pack** used in previous releases. Linked views can be *any* visualization of the data in the System Viewer log, including, but not restricted to, the standard analyses (Core Usage, System Load, Memory Usage, Running State, Ready State, and so on). Furthermore, linked views are always synched to each other; that is, any changes to cursor measurement, range selection, bookmarks, and so on, are immediately visible in every linked view. If you notice something unusual in one view, you can, for example, set the cursor in that view and thereby immediately home in to the problem area in all the other linked views.

### New Overview (previously known as 'Radar') Features

#### Overlays

Any analysis can contribute an additional visualization overlay layer to be displayed on the Overview.

#### Cropping

Cropping removes data on either side of a selected range in the Overview so that the full range of the log is equal to the selection, allowing you to focus entirely on the range that is of most interest. Cropping can also be undone.

### Filtering/Searching Improvements

The following Filtering/Searhing improvements are implemented in this release:

- The query interface is now a tabbed dialog; results are displayed in an integrated Workbench view that you use to navigate matches.
- A **Search/Filter History** of previously created searches/filters is maintained. Existing custom filters/searches can be renamed at any time.
- Search/filter results can be viewed both as a table or as an Event Graph and are extendable to any analysis.
- New, ready-made filters to hide unnecessary clutter include:
  - hide inactive contexts
  - hide non-executing states
  - “flatten” the context tree (unnest nested contexts)
- Higher filter granularity lets you selectively include event types within event classes.
- You can now filter on objects.

## General Interface Improvements

Right-click context menus have been enriched throughout. As well, several features previously presented on dialogs are now in integrated views and therefore persistently visible/accessible.

## Wind River Run-Time Analysis Tools

This section describes the enhancements and improvements to the Run-Time Analysis Tools (formerly ScopeTools) provided with this release.

### All Tools

#### Analysis Views Integrated into Workbench

The previously released Analysis Suite has been fully integrated in the System Viewer application, which has in turn been fully integrated into Workbench. This new architecture facilitates extensibility and enhances usability.

#### Reworked Interfaces

The interfaces of Performance Profiler, Code Coverage Analyzer, and Memory Analyzer have been reworked to simplify workflow and improve usability.

#### Port Number is Now Configurable

Rather than always connecting to a known port number, Workbench allows you to specify a port number.

### Performance Profiler

#### Target Port Selection

Target port selection is now configurable in the Properties dialog box.

#### Simplified Profiling Options

Profiling options have been simplified and are no longer buried in the Preferences; they have been moved to a dedicated Properties dialog box. To access the Properties dialog box, right-click into any Performance Profiler view and select **Properties** on the context menu.

### Code Coverage Analyzer

Source code is now opened in the CDT editor (with symbol and syntax highlighting).

The maximum number of command line arguments for the instrumenter as well as for the invoked compiler command has been increased to 1024.

## 5.2.2 Fixed Problems

For a list of problems fixed in Wind River Workbench, visit the Online Support Web site (see [1.1.3 Latest Release Information](#), p.2).

### 5.2.3 Unsupported Features

#### Wind River Function Tracer

Support for Wind River Function Tracer has been discontinued with this release.

#### VxWorks Scalable Kernel Profiles

The VxWorks profiles that allowed for scaling of the VxWorks kernel image have been deprecated. This functionality is now available through VxWorks Source Build projects. For more information, see *Wind River Workbench By Example*.

#### Support for DDC-I Ada Compiler

Workbench no longer supports Ada applications built with the DDC-I Ada compiler.

## 5.3 Usage Caveats

### Workbench Issues

#### Red Hat Enterprise Linux 5.2 Must Use Latest Firefox Packages

Red Hat Enterprise Linux 5.2 ships with a beta library as part of the Firefox 3.0 web browser. This beta library may cause the Eclipse environment provided with Workbench to crash on startup. The crash logs will generally show a segmentation fault in **libxul**.

To avoid this problem, update the Firefox/xul packages to the latest available version. Use the following steps:

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

```
yum update
```
3. Exit root mode with the **exit** command.

#### Workbench 2.x Projects Migrated to Workbench 3.x Must Be Reindexed

Beginning with Workbench 3.0, a new technology is used for source analysis. The database formats are different, and the format of Workbench 2.x is not supported by the new indexer. All projects must be parsed again in Workbench 3.x, regardless of whether you use the same workspace or import old projects.

#### Binary Incompatibility for UP and SMP Downloadable Kernel Modules

Downloadable kernel modules built for uniprocessor systems are no longer binary compatible with those built for symmetric multiprocessor systems. Starting with this release, you must specify the type of system that the module will run on during project creation, so Workbench can apply the correct build settings.

### Workbench Supports US English Language Only

Workbench supports and is tested only with US English. Support for other languages is not available at this time.

### Double Brackets Not Supported in Makefiles Run in the VxWorks Development Shell

The `[[` construct is no longer accepted in VxWorks makefiles after issuing the `sh.exe` command in a VxWorks development shell. You must use the standard `[` construct. For example:

This syntax will fail:

```
if [[ ! -d $(TODIR) ]]; then ...
```

To make it work, use a single `[`:

```
if [ ! -d $(TODIR) ]; then ...
```

This will provide the same functionality as the double brackets construct, but might be slightly slower because the double bracket test syntax is built into the Korn shell. The single bracket test calls a program named `test` to test the values, which is slower.

The double bracket test will not work in the Bourne shell. (WIND00117962)

### -RedirectIO Flag is now Obsolete

In Tornado, you needed to use the `-redirectIO` flag with the Target Server to redirect I/O to the Wind Shell. In Workbench, the `-redirectIO` flag has been made obsolete, and I/O is now redirected to the `wtxConsole`.

### Workbench 2.6.1 Registry Does Not Support Workbench 3.x

The Workbench 3.x registry will support running Workbench 2.6.1, but the reverse is not true. So if you launch Workbench 2.6.1, then close it and start Workbench 3.x without shutting down the running registry, an error will appear.

To get around this, always shut down the Workbench 2.6.1 registry before starting Workbench 3.x.

### Multiple Users and Installations

Different configurations of the number of Workbench users and the number of Workbench installations on a single host are possible, and the following considerations apply.



---

**NOTE:** Each user should work in a user-specific workspace. Sharing of workspaces is not possible. To specify a workspace, either use the `-data` startup option, select a workspace in the Workspace selection dialog during startup, or select **File > Switch Workspace** in Workbench.

---

### Single User with Single Installation

This configuration presents no special installation or use considerations, assuming the users on each host perform their own installation and have standard access permissions to the installation location. No special startup arguments are necessary and the default workspace may be used.

### Multiple Users with Multiple Installations

The same conditions apply here as they do for a single user with a single installation, with the exception that a single administrative user often performs the different Workbench installations. In this case it is important that the proper permissions be granted to each user for access to their particular installation. Linux and Solaris root users performing multiple installations should refer to the **umask** and **chmod** manual (**man**) pages.

### Multiple Users with a Single Installation

Multiple users can share a single Workbench installation as long as the access rights allow them to read all files of the installation (same group as the user who installed Workbench). Linux and Solaris users should refer to the **umask** and **chmod** manual (**man**) pages.

For performance reasons, it is desirable to have the workspace on a local file system. Some Eclipse-specific data is stored in the user's home directory by default. If this is not desired because of slow network access, use the **-configuration** startup option to redirect this data. Refer to *Eclipse Workbench User Guide: Running Eclipse* in the online help for more information on startup options.

### Single User with Multiple Installations

If a single user is installing Workbench more than once, it is important that the configuration area (%USERPROFILE%\workbench-3.x.build\_ID in Windows and \$HOME/workbench-3.x.build\_ID in UNIX) does not get corrupted. Different versions of Workbench will not conflict, but for multiple installations of the same version for the same user, different configuration areas should be specified at startup with the **-configuration** option.

### Eclipse Team Features

Workbench supports all the team features of the standard Eclipse installation as documented in the online help supplied with Workbench.

### Workspace Location

Wind River strongly recommends placing your workspace on a local file system. Remote file systems can cause significant performance problems. If you use ClearCase, place the workspace outside a dynamic view, otherwise you will face significant performance problems.

### Turning off Static Analysis for Large Projects

Static analysis is turned on by default. For large projects (such as kernel build projects) static analysis can slow down Workbench functionality considerably.

To turn off static analysis:

#### For existing projects:

Right-click the project and select **Properties**, then select **C/C++ General > Indexer**. From the **Select Indexer** drop-down list, select **No Indexer**. Click **OK** to close the dialog.

#### For new projects:

Select **Window > Preferences** and choose **C/C++ > Indexer**. From the **Select Indexer** drop-down list, select **No Indexer**. Click **OK** to close the dialog.

#### By project:

If you would prefer not to turn off static analysis for all new projects, you can turn it off for a particular project as you create it. Click **Next** in the New Project wizard until you reach the Indexer page, then select **Enable project specific settings**. From the **Select Indexer** drop-down list, select **No Indexer**, then click **Finish** to create the project.

#### Supported Make System

The Workbench make system is based on features of the GNU Make utility (**gmake**). Other make systems are not supported.

#### Increasing Virtual Memory

Building large applications can cause Java to run out of memory. To avoid this, increase the size of memory for the Java virtual machine to 512 MB or higher, either manually (when you launch Workbench from the command line) or by editing a launch command.

#### From the command line:

##### On Windows:

From a shell, type the following:

```
C:\> cd installDir\workbench-3.x\wrwb\platform\eclipse
C:\> .\wrwb-x86-win32.exe -vmargs -Xmx512m
```

##### On Linux and Solaris:

Use the values as parameters to the **startWorkbench.sh** script:

```
% ./startWorkbench.sh -vmargs -Xmx512m &
```

where 512 corresponds to 512 MB of RAM for the virtual machine.

#### From the Windows Start menu:

1. Select **Start > Programs > Wind River > Workbench 3.x**, then right-click **Wind River Workbench 3.x** and select **Properties**.
2. From the **Shortcut** tab, move the cursor to the end of the command in the **Target** text box.
3. After **wrwb-x86-win32.exe**, type **-vmargs -Xmx512m**.
4. Click **OK**.

Each time you launch Workbench using the **Start** menu, you will allocate the appropriate amount of memory for the Java virtual machine.

#### From a desktop shortcut:

Right-click the shortcut, select **Properties**, and edit the **Target** command line as described above.

## Accessing Online Support from the Workbench Welcome Page

### On Windows

If Internet Explorer is installed on your system, you must enable cookie support for the Internet Explorer ActiveX control before you can access the VxWorks 6.x Release Web site from the Workbench Welcome page. If your host machine is configured not to accept cookies for other sites, follow these steps to override:

1. From the Windows **Start** menu, select **Settings > Control Panel > Internet Options**.
2. Select the **Privacy** tab.
3. Under **Web Sites**, click **Edit**.
4. Under **Address of Web site**, type **secure.windriver.com**, click **Allow**, then click **OK**.

You should now be able to access the Online Support login page.

If you have removed Internet Explorer from your system, Workbench opens your default browser and your regular cookie settings apply.

### On Linux and Solaris

Workbench opens your default browser. Your regular cookie settings apply.

## Viewing Workbench Help and Other Documentation on Solaris

To view Workbench documentation in Netscape on Solaris, you must adjust your Workbench preferences to run Netscape (the default browser is Mozilla):

1. From within Workbench, select **Window > Preferences > Help > Custom Browser Command**, then enter the full path to Netscape:

```
/usr/dt/bin/netscape
```

If Netscape is already in your path, you can type just **netscape**.

2. Netscape asks for a cookie by default. Accept the internal cookie.

You can now view the Workbench documentation in Netscape, either context-sensitive help from Workbench itself (by pressing the **CTRL+F1** and selecting a link from the help view) or by navigating to it from **Help > Help Contents > Wind River Documentation**.



---

**NOTE:** The **Help** button on Solaris keyboards does not open Workbench help due to a problem in Solaris/GTK+. Instead, use **CTRL+F1** to access help.

---

## Using the Help Browser Independently of Workbench

For a full-featured help browser that runs independently of a particular instance of Workbench, create a script with the commands given below, and then start it in the proper environment for your operating system.

### On Windows

1. Open a VxWorks Development Shell by selecting **Start > Programs > Wind River > VxWorks 6.x and General Purpose Technologies > VxWorks Development Shell**.



---

**NOTE:** Starting this shell sets the appropriate environment variables for you.

---

2. Create and save a batch file with a descriptive name (such as **showhelp.bat**) with the following content. Replace *installDir* with your Workbench installation directory path and *3.x* and *version* with the appropriate version number.

```
echo off
echo Please close this shell with CTRL+C when you have finished browsing the
documentation.
java -classpath installDir\workbench-3.x\wrwb\platform\eclipse\plugins\org.eclipse
.help.base_version.jar org.eclipse.help.standalone.Help -command
displayHelp /com.windriver.ide.doc.globals/toc.xml -eclipsehome installDir\work
bench-3.x\wrwb\platform\eclipse -noexec
pause
```

3. Run your new batch file in the shell; it may take a minute or two for the help browser to appear.
4. When you are finished browsing the documentation, close the help browser window, then press **CTRL+C** in the shell. When asked if you want to stop the batch file, press **Y** for yes.
5. At the shell prompt, type **exit** to close the shell.

#### On Linux/Solaris

1. In a terminal window, create and save a script with a descriptive name (such as **showhelp.sh**) with the following content. Replace *installDir* with your Workbench installation directory path, and *3.x* and *version* with the appropriate version number.

```
#!/bin/sh

echo Please close this shell with CTRL-C when you have finished browsing

exec installDir/jre/version/x86-linux2/bin/java -classpath
installDir/workbench-3.x/wrwb/platform/eclipse/plugins/org.eclipse.help.base_versi
on.jar org.eclipse.help.standalone.Help -command displayHelp
/com.windriver.ide.doc.globals/toc.xml -eclipsehome
installDir/workbench-3.x/wrwb/platform/eclipse -noexec pause
```



---

**NOTE:** The **exec** command shown above is all one line to the final **pause**.

---

2. Save the script and make it executable:  

```
$ chmod +x showhelp.sh
```
3. Copy the script to your installation directory:
4. 

```
$ cp showhelp.sh installDir
```
5. Change to your Workbench installation directory.  

```
$ cd installDir
```
6. Run the **wrenv** command to initialize your Wind River environment:  

```
$ ./wrenv.sh -p workbench-3.x
```
7. Run your new script in the shell; it may take a minute or two for the help browser to appear.

```
$ ./showhelp.sh
Please close this shell when you have finished browsing
<<<press CTRL-C when you are done with the help browser>>>
```

8. When you are finished browsing the documentation, close the help browser by entering **CTRL+C** in the terminal window. When asked if you want to stop the batch file, press **Y** for yes.
9. At the shell prompt, type **exit** to close the shell.

### Viewing Individual Wind River Documents in a Browser

It is possible to view a single document at a time using the HTML files in `installDir\docs\extensions\eclipse\plugins`. This does not require that you have Workbench running, nor must you run a batch file.

For example, to view the *Wind River Workbench User's Guide, 3.1*, enter the following URL into your browser, substituting your Workbench installation path for `installDir`:

```
file://installDir/docs/extensions/eclipse/plugins/com.windriver.ide.doc.wr_workbench_vxworks/wr_workbench_vxworks_users_guide_3.1/html/index.html
```



**NOTE:** This method does not provide all the features of the help browser provided by Workbench help, such as full-text search, bookmarks, and a fully expandable table of contents for the whole documentation set.

### Setting the Number of Open Files in Editor

Workbench allows you to open as many files as you like in the Editor.

To limit the number of files you have open at one time, select **Window > Preferences > General > Editors**, and select **Close editors automatically**. Increase or decrease the number in the **Number of opened editors before closing** field, then click **OK**. This causes Workbench to close one of the open files as soon as you try to open another file that exceeds your set limit.

If you only want to keep those files open that you are particularly interested in, set the **Number of opened editors before closing** field to **1**, causing Workbench to close each file as it opens the next one. For those files you want to keep open, click the **Pin Editor** icon on the main Workbench toolbar. Workbench keeps that file open, and opens a new editor the next time you open a file.

### Dragging and Dropping in Workbench

When dragging information from the Editor and dropping it into the Expressions view or other data views, press the **Ctrl** key to ensure that you copy the information to the destination rather than cutting and pasting it.

This also applies if you are dragging and dropping the information to an application outside of Workbench.

### Coprocessor Support

The build mechanism provided for process-based applications does not support pre-defined compiler options for coprocessors.

In Workbench, these options must be specified in the `CC_ARCH_SPEC` build macro. For more information, see *Wind River Workbench User Interface Reference: Build Properties Dialog*.

From the command line, you can add build options to an existing makefile with the `ADDED_CFLAGS` macro. This example illustrates usage for Altivec:

```
include $(WIND_USR)/make/rules.rtp
```

```
ADDED_CFLAGS += -tPPC7400EV:rtp
```

For the Wind River Compiler, the last **-t** option specified in the makefile overrides any **-t** option specified on the command line. For the GNU compiler, **-fvec-eabi** can be added with the **ADDED\_CFLAGS** macro. For coprocessors other than AltiVec, a similar approach can be used to specify the corresponding compilation options.

### VxWorks 5.5 Support

For VxWorks 5.5, only task objects are shown in the Kernel Objects view.

### VxWorks 5.5 Debugging Using OCD with Workbench

When using Workbench with Tornado 2.x tools over an OCD connection, the Tornado 2.x registry is shut down and only the Workbench registry is accessible.

### Renaming Projects on Windows

Projects can be renamed, even with an open connection and the debugger attached to a build target output file of the project, as long as the file size does not exceed the setting in the **Maximum host-side Symbol File Size to load fully into In-Memory Cache** field (see **Window > Preferences > Wind River > Target Management > Debug Server Settings > Symbol File Handling Settings**).

### Disabling Startup Code Can Impact Workbench

Workbench launches a background process called the Wind River Registry during **org.eclipse.ui.startup**. The Registry manages multiple connections from Workbench to target boards. The launch process does not block further execution of other components.

Disabling the startup code in Workbench will result in a diminished capacity for Workbench.

This functionality is only executed if you have the Application Development perspective or the Remote Systems view open.

## Project and Build Issues

### Running Workbench Remotely

Wind River recommends using Virtual Network Computing (VNC) software (<http://www.realvnc.com/>) when running Workbench on a remote Linux or Solaris server. User interface responsiveness is much better compared to the standard remote X access method.

The **Remote Workspace Location** field of the **Remote Connections** dialog must contain the absolute path to the root directory of the workspace, as seen on the remote host. Environment variables are not supported in this field.

## Wind River Run-Time Analysis Tools

### Changing Timeouts to Accommodate RTP Initialization Task

If you want to run an RTP on your target after starting one of the Analysis tools, the RTP spawn time limit should be set at **120** seconds or greater, and the back end request time limit should be set to **30** seconds. With the target disconnected, edit

these values in the **Advanced target server** panel in the **Target Server Options** tab view of the target **Properties** dialog box.



---

**NOTE:** If you do not attend to both of these items, the RTP initialization task may not receive sufficient CPU time to complete its execution before the RTP spawn time limit expires and causes the host to stop all tasks running in the RTP.

---

### Configuring a Custom VxWorks Simulator with Networking Support

The Workbench default simulator (Wind River VxWorks Simulator), which uses a **wdbpipe** back end connection (**INCLUDE\_WDB\_COMM\_PIPE** enabled), connects to a VxWorks target but does not successfully connect to any of the Analysis Tools tools using TCP/IP. To connect using TCP/IP, you *must* configure a custom simulator with **INCLUDE\_WDB\_COMM\_END** enabled instead. You will also have to set up a **wrtap** adapter, and install a network daemon.

For detailed information on setting up a VxWorks simulator with full networking support, see the *Wind River VxWorks Simulator User's Guide*.

## Windows Host-Related Issues

### Defragment Hard Disks to Decrease Workbench Startup Time on Windows

If it takes Workbench a minute or more to open on Windows, this could be because your hard disk is very fragmented. Defragmenting your hard disk could reduce Workbench's startup time by half or more.

## Linux Host-Related Issues

### Welcome Page Requires Supported Browser

The default Welcome implementation is HTML-based and requires a supported browser in order to work. If no supported browser is found, the Welcome page displays in its Forms-based implementation, which has a different (simpler) appearance. Consult the SWT FAQ (<http://www.eclipse.org/swt/faq.php#browserplatforms>) for supported browsers and instructions on setting up your browser to work with Eclipse.

### Workbench Crashes on SuSE Running KDE

A problem exists in the **gtk-qt-engine** system library provided by SuSE/Novell. If this package is present, Workbench may crash on startup. The only workaround is to remove this library. This will have consequences for any program that uses the library.

To remove the library, issue the following command:

```
$ rpm -e gtk-qt-engine
```

## Licensing

The licensing system for Wind River Workbench assigns a license based on a combination of a unique machine ID and a unique name. If multiple Linux machines share the same name (such as **localhost.localdomain**), installation and

licensing only works on the first machine to install it. To correct this problem, do the following:

1. Select **Start > System Settings > Network** on your Linux machine, and select the DNS tab.



---

**NOTE:** You are prompted to log in as **root** to view these settings.

---

2. In the **Hostname** field, replace the existing host name with a unique name for your system.
3. Close the **Network Configuration** dialog.
4. Add your new host name to the list of host names associated with address 127.0.0.1 in the **/etc/hosts** file.
5. Reboot your system so that the new network settings take effect.

## Solaris Host-Related Issues

### Starting the VxWorks Simulator on Solaris

In order to start the VxWorks simulator on Solaris, the path environment variable must include **/usr/openwin/bin** so that it can find **xterm**. If **xterm** is not in the path, the simulator connection fails.

## 5.4 Known Problems

This section lists some known problems with Wind River Workbench. For a complete list of known problems in Wind River Workbench, visit the Online Support Web site (see [1.1.3 Latest Release Information](#), p.2).

### Workbench Issues

#### Help Unavailable for Eclipse Plug-in Registry and Error Log Views

Workbench provides the Plug-in Registry and Error Log views, but it does not include the entire Plug-in Development Environment (PDE) so the Workbench help system cannot display help for either of these views.

To find the documentation for the PDE, see <http://www.eclipse.org/documentation/>.

#### Workbench sh.exe Hangs During Automated Build

If a make process hangs on Windows, check all involved Makefiles for any complex pipe (“|” operator) usage. Due to a general problem with Windows' pipe handling, Wind River recommends that you do not use the “|” operator for combined commands in Makefiles. Instead, split them into explicit commands.

Makefiles created by Workbench managed builds do not contain any “|” operators by default.

### Workbench Does Not Pass IPv6 Build Flag to VxWorks Image Project Builds

Information for how libraries are built is not visible when a VxWorks image is built, so even if IPv6 is set to **true** in **config.mk**, Workbench does not pass the IPv6 build flag.

To workaroud this problem, add the following to the top of the header:

```
#ifndef INCLUDE_IPCOM_USE_INET6
#define IPCOM_USE_INET6
#endif

<app includes go here>
```

### Problem Interpreting Unquoted Numbers as Arguments

When creating launch configurations, unquoted numbers in **Arguments** fields are interpreted as address values by the underlying debugger components. The corresponding argument value will be read from the memory specified by this address value.

To read a number as a string instead of an address, enclose the number in quotation marks (for example, "1").

### Inter-Project Dependencies

Inter-project dependencies in Workbench do not work for Shared Library projects and User-Defined projects. (WIND00196929)

### Workbench May Crash when Using KDE with BSD Automount Daemon (amd)

When using Workbench on Red Hat Enterprise Linux (RHEL) with KDE, Workbench may crash when you use a native directory dialog to browse to a file system which is mounted by the BSD automount daemon **amd**. (WIND00069984)



---

**NOTE:** You will have similar problems when using any system application, such as **gedit**. Wind River recommends that you do not use **amd** with RHEL and KDE since it will crash any standard application.

---

### Undefined Symbols with C++ Applications

You may get undefined symbols when linking or downloading C++ applications. C++ applications must be compiled with the same compiler used to build the VxWorks image.

For example, if the VxWorks image is built with the Wind River Compiler, the C++ applications must also be built with the Wind River Compiler. Likewise, if the VxWorks image is built with the GNU compiler, the C++ applications must be built with the GNU compiler.

### C++ Code Completion

Code completion for C++ templates (including shared pointers) does not work. (WIND00196919)

### Workspace Locks After an Unexpected Reboot

If your host unexpectedly reboots during a Workbench session, the **.lock** file may remain in your workspace. This means that when you restart Workbench, your workspace is locked and you cannot access your projects.

To unlock your workspace, use a shell window to navigate to your Workbench install directory, then delete the *workspaceDir/.metadata/.lock* file.

### Workspace Remains Locked After Forced Quit of Workbench on Linux

If Workbench is not responding and you close the Workbench window, some Linux window managers may not actually kill the Workbench application. The Workbench window may disappear, but the application remains running.

After you click the **Close** icon on the title bar, the window manager may ask if you want to quit the application:

**The window Application Development - Wind River Workbench is not responding. Forcing this application to quit will cause you to lose any unsaved changes.**

If you choose to quit, Workbench may still be running and you will not be able to open the same workspace. The workaround for this problem is to manually kill the Workbench Java process using the **kill** command.

### Workspace in Use Errors

Several issues can cause Workbench to display the error **Workspace in use, choose a different one**.

Possible reasons are:

- Workbench or Eclipse does not have write access to the workspace or **.metadata** folder.
- Another instance of Workbench is running in the same workspace. Only one instance of Workbench can use a workspace at a time.
- The workspace may not have unlocked after an abnormal exit. To unlock it, kill the remaining Workbench and Java processes. On Linux, check which process still has an open file handle on the file *installDir/workspace/.metadata/.lock* by using the command **ls -lsof**.
- Your workspace is on a file system that does not support locking. Several issues in Eclipse and Java prevent the locking mechanism from working properly on some NFS-exported file systems such as HP. In such cases, after trying all the suggested fixes, use another Java Virtual Machine (JVM) argument to start Workbench:

```
./startWorkbench.sh -data /nfs/exported/hp/filesystem -vmargs  
-Dosgi.locking=none
```



---

**NOTE:** This will allow multiple users to open the same workspace, which may result in potentially unrecoverable data loss.

---

### Workbench Startup Reports “An Error Has Occurred”

If Workbench starts with the message **An Error Has Occurred**, try starting Workbench with the **-clean** option. On Windows, modify the shortcut that starts Workbench to **wrwb.exe -clean**. On Linux, start Workbench with:

```
$ startWorkbench.sh -clean
```

If that does not help, remove the Workbench configuration directory `$HOME/.workbench-3.x.build_ID` (Linux) or `%USERPROFILE%\workbench-3.x.build_ID` (Windows).

If neither of these methods works, try using a new workspace.

### Workbench Reports Fork Problems on Linux and Solaris

On Solaris and Linux, Workbench may report a problem if it fails to execute an external process. For example:

```
java.io.IOException: Not enough space at  
java.lang.UNIXProcess.forkAndExec(Native Method)
```

The Java run time has the limitation that external processes are launched using **fork**, which requires that enough free memory (including swap space) is available. The fork tries to reserve the same amount of memory as the parent process, in this case, the Workbench process. After the successful launch of the process, only the actual required process memory will be allocated. If the maximum heap size is set too high, the child process cannot be started and Workbench operations will fail.

Make sure that:

- Your system is configured with enough swap space.
- Workbench is started with a reasonable heap size, which is not unnecessarily high. Choose a heap size which is always lower than half of the freely available memory (physical and swap). Note that every started process will reduce the available memory.

### Resetting Workbench to its Default Settings

If Workbench crashes, some of your settings could get corrupted, preventing Workbench from restarting properly.

To reset all your settings to their defaults, delete the `.workbench-3.x.build_ID` directory in your home directory. It is recreated when Workbench restarts.

### Workbench Font Size Problem with Exceed 8

Running Workbench with Exceed 8 causes font size problems. Workbench appears correctly when running with Exceed 7.1 or Exceed 9.0.

### Online Update

It is not possible to use the **Help > Software Updates > Find and Install** feature of Eclipse to update the Eclipse platform or Workbench itself. However, the feature can still be used to update any third party plug-in that you may have added to the installation.

To update Workbench, please download and install available patches from the Wind River Online Support site (see [1.1.3 Latest Release Information](#), p.2).

### Printing in Workbench

On Windows, **File > Print** is enabled only for the Source Editor and the Disassembly view.

### Ada Editing is No Longer Supported

Because of the switch from the Workbench Editor to the Eclipse Editor, Ada syntax highlighting is no longer available. Debugging still works, but you must use the default text editor for Ada files.

## Project Issues

### Rebuilding While Debugging

If you want to rebuild a project after a debugger session has started, you may need to disconnect from the debugger (depending on the type of project):

- To rebuild an RTP, the RTP must not be running, either having exited or been terminated from Workbench. You do not need to disconnect from the debugger.
- To rebuild the VxWorks kernel image, you must disconnect from the debugger.
- To rebuild a kernel module, you can either disconnect from the target, or you can make sure the kernel module is not being used by any tasks, right-click the kernel module file, then select **Delete** (dialog prompts you with further options and cautions regarding rebuilding while still connected to the target).

### Non-English Character Restrictions

Workbench does not support the use of spaces (blank characters) or non-English language characters in project names. The usage is not prohibited, but it leads to errors.

Non-English language characters are also not supported for user names on Windows.

## Remote Systems View Issues

### Update of Display is Slow when Listening to Execution Life Cycle Events

When more than 100 threads are running on the target, selecting **Listen to execution context life cycle events** can severely delay the update of the Remote Systems display. Do not use this feature when there are more than 100 contexts on the target.

### Run RTP Exec Path Should be Checked for Accuracy

Each time you run an RTP by right-clicking the executable and selecting **Run RTP** from the Remote Systems view or the Project Explorer, check the exec path in the **Run RTP** dialog for accuracy. The path that automatically appears may not be correct.

### Workbench Requires a Host Name to be Set

Workbench cannot connect to the target registry or debug when the host name of the local machine is set to **localhost** (which is the default on some Red Hat Linux distributions).

On Red Hat Linux, the host name can be changed in the **System Settings > Network > DNS** tab. If you cannot change the local host name, you can work around the problem by setting the **-host** option in the file *installDir/workbench-3.x/foundation/version/resource/wtxregd/wtxregd.conf*. (SPR 108938, WIND00013729)

#### Cannot Connect Registry or Debug Server (All Hosts)

When Workbench cannot connect to the Wind River Registry, or the debug server connection fails (logged as **Failed to create Target Control**) check your network configuration.

1. Check the error messages for host names and IP addresses, and do a **ping** at a command prompt to check whether the corresponding host names and addresses are actually reachable. Also, try **nslookup** with the host names to check if the DNS service is configured properly.
2. Try **ping localhost** and **ping 127.0.0.1** to verify that the loopback interface is up and configured properly.

#### Drag and Drop is not Supported for Wind River Target Connections

It is not possible to drag images from the Project Explorer to Wind River target connections.

### Build Issues

#### Problem Building User Mode Agent on Different Linux Host Version

When building the Linux user mode agent (**usermode-agent**), it is important to build it on the version upon which you intend to run it.

- When built on a Red Hat Linux Workstation 4 host, the x86-64 agent does not work reliably on a Red Hat Linux Workstation 5 machine.
- When built on a Red Hat Linux Workstation 5 host, the x86-64 agent does not start at all on a Red Hat Linux Workstation 4 machine.

The version of the agent shipped with this release is built for Red Hat Workstation 5. To debug on a Red Hat Workstation 4 host, rebuild the agent on that platform.

### Build Properties Issues

#### Manual Addition of New Build Macros

New build macros you define in the Build Properties of a managed build project are not automatically added to any of the build tools. You must manually add the macro to the command line of the appropriate build tool(s) where appropriate. An **Add to all** button is available in the Build Paths settings which allows you to add a new Include Path to all build specs of the project in one step.

#### Build Property Adjustments for CPP Demo Kernel Module Sample Project

Launching the cpp demo Kernel Module sample project on an ARM target causes the error **WTX Loader error: relocation offset too large**.

A workaround for this problem is to modify your C++ build tool properties.

- For `diab`, include `-Xcode-absolute-far`.
  - For `gnu`, include `-mlong-calls`.
1. Modify these properties by right-clicking on the project, selecting **Properties**, and then selecting the **Build Tool** Tab.
  2. Select **C++ Compiler** in the **Build Tool** pull-down box.
  3. Edit the **Command** field to include the above directive after the `$(ADDED_CFLAGS)` text.

Note that the *VxWorks Architecture Supplement: Building Applications* states to use the `$(LONGCALL)` macro to be tool agnostic, but this macro does not resolve. (SPR 109859)

## Debugging Issues

### Problem Removing Breakpoint on KGDB Target

When a KGDB target is running and you remove a breakpoint that is installed on the target, the debugger can fail to remove that breakpoint if the target is in process of hitting the breakpoint that is supposed to be removed.

In this situation, the target will continue stopping at this breakpoint even though it is no longer present in the Breakpoints view. The only way to remove the breakpoint is to disconnect from the target and then to re-connect to it.

### Casting an Address to a Structure Requires Extra Parenthesis

When passing an expression containing parentheses to the debugger, you must add extra parentheses to the expression. For example, if you place a watch on the following expression and expand it, the members get errors:

```
*(GRID *)0x20658
```

However, if you add parentheses around the address, the entry expands correctly.

```
(* (GRID*) 0x20658)
```

In order to access members, you must also surround the cast with parentheses:

```
(* (GRID *) 0x20658) .x
```

```
(WIND00016848)
```

### Consecutive Launches Using F11

Issuing multiple launch commands of the same application consecutively (using F11) may cause delays in Workbench view updates. This is due to the network traffic required to fetch stack information for each launched application.

### Configuring WDB for Debugging a Large Number of Tasks

The WDB agent's default Gopher configuration calls for 14,000 bytes of Gopher tape. This accommodates debugging about 300 tasks with short names, but if you use longer task names or more than 300 tasks, the Remote Systems view, the shell, and the debugger will fail.

To avoid this problem, set `WDB_GOPHER_TAPE_NB` to a higher value (such as 20). This run-time parameter can be located and modified using the Kernel Configuration Editor.

### Debugger Link Exception when Processing Symbol Request Return Value

When using Performance Profiler, the debugger occasionally stops resolving symbols after a reconnect. Disconnecting the target server from the target, then reconnecting the target server and Performance Profiler may resolve the error. (WIND00070739)

### Target Server and ICE Connection Conflicts

Though the Remote Systems view allows you to connect an emulator and a target server to a single board and use both connection types simultaneously, you must set the target server time-outs to longer default values to avoid the target server automatically disconnecting when the OCD connection stops the target.

For information about setting target server time-outs, see *Wind River Workbench User's Guide*.

### Debugger Views

#### Debug View

In GTK+ on Linux, the Debug view icons do not wrap when the view is smaller than the width of the toolbar. So if you do not see all the icons discussed in the *Wind River Workbench User's Guide: Debugging Projects*, expand the Debug view horizontally until all the icons appear on the toolbar.

#### Run Control

Step return is not fully working for recursive functions. For example, consider the following simple recursive function:

```
int factorial( int x )
{
    if ( x > 1 )
    {
        int result = x;
        result *= factorial(x-1);
        return result;
    } else
        return 1;
}
```

If the **factorial()** function is called with an argument of 6, then you perform some **Step** operations until **factorial()** is called with an argument of 5, the call stack now contains two function calls: **factorial(6)** and **factorial(5)**.

If you step to the statement **result \*= factorial(x-1)**; and then request that the debugger perform a **Step Out** operation, you would expect to end up in the calling function, which is **factorial(6)**.

However, since the debugger sets a breakpoint at the statement **return result**; and then performs a **Go** operation, this recursive algorithm actually stops in the **factorial(2)** call. This is because the breakpoint isn't actually reached until the recursive function has completed calling itself all the way down to **factorial(1)** and then started returning values (unwinding the call stack).

### Data Views

Changing the value of a variable or expression in data views (such as the Expressions view) may return an error, depending on the nature of the variable or expression.

For example, if the expression refers to a WRITE-ONLY register, a write succeeds but a read always fails. (SPR 103034)

STL string types are not decoded/displayed in the Variables view when debugging. (WIND00196185)

## Indexer Issues

### Macro Limitations

No macro references appear in the call hierarchy (for MACROs that look like functions).

### C++ Limitations

- Template support is limited.
- There are no references to constructors, destructors, or implicit type conversions in the symbol list. However, Declarations and Definitions are in the symbol list.
  - An implicit conversion (sequence) is generated when the type of an argument in a function call does not match the type of the parameter. It can make use of constructors to convert the argument to an object of the expected type.
  - Implicit constructor calls may be needed for the initialization of base classes, automatic variables, or returned values.
  - An implicit destructor call is necessary when an automatic variable goes out of scope.

### Call Tree Limitations

There are no read/write flags for variables, and polymorphic method calls are not honored.

### External APIs Cannot Be Configured

The preferences page for configuring external APIs no longer exists, because these APIs can no longer be configured by the user (they are associated with a project depending on the project type). External APIs are supplied for Wind River-specific project types (such as VxWorks project types).

## Properties View Issues

### Quotes Not Preserved in Parameter Values

In the Properties view, Eclipse does not preserve embedded quotes in parameter values. As a workaround, use \" before and after the parameter.

For example, to include `“ppp”`, type `“\”ppp\”\”ppp\””`. (SPR 98394)

## Run-Time Analysis Tools Issues

### WDB\_TIPC Connections Not Supported with Run-Time Analysis Tools

The Run-Time Analysis Tools do not support connecting to a target over a WDB\_TIPC connection. This means that if you are working in an AMP environment, you cannot connect the Analysis Tools to cores that do not support networking over Ethernet, such as core 1 on the Broadcom BCM1480 board.

### Code Coverage Analyzer Issues

#### Coverage Instrumenter and Unusually Obfuscated C++ Language

The coverage instrumenter may not be able to handle unusually obfuscated C++ language constructs. If so, the instrumenter writes a parse error message to the build console and ignores the corresponding statement. It is not necessary to add the source file to the **ignore** list, unless the instrumented file is no longer buildable.

#### VxWorks Target Heap Memory

On VxWorks targets the available heap memory may be limited.

#### Java VM Heap Size

The heap size of the Java VM is a limiting factor when handling a large number of tags. To workaroud this issue, you can increase the VM size by appending **-vmargs -Xmx2048m** to the Workbench commandline (**2048m** stands for 2048 megabytes).

Note that the specified heap size should not be larger than half of the available system memory on Linux.

## System Viewer Issues

### Continuous Upload from Target not Recommended

Continuous upload from a target with a slow connection link is problematical. It is strongly recommended that you use a different upload method for slow connections.

### Out-of-Memory Problems with Large Parameters in the Log

If extremely large parameter payloads are encountered in the log, out-of-memory problems can occur. To prevent this, only the first 1024 bytes of either **STRING** or **BLOB** event parameters are now displayed by default. You can increase or decrease this value by setting the environment variable **WRSV\_PARAM\_VALUE\_MAXBYTES** before you start Workbench; a value of **-1** will display all bytes.

### Events in Nested Interrupts Can Generate Warnings in Log Load Report

On some boards, events in nested interrupts may cause warnings to appear in the log load report, and some events will be shown in an interrupt context instead of a task. This can happen if interrupt controller drivers emit several events in the interrupt demultiplexor. In such cases, the number of **INT\_ENT** events and **INT\_EXIT** events emitted by the interrupt demultiplexor must satisfy:

```
number(INT_EXIT events) == number(INT_ENT events) - 1
```

A workaround has been published on the Wind River online support site; see WIND00046653.

# 6

## Known Issues

- 6.1 Usage Caveats 43
- 6.2 Known Problems 49
- 6.3 Documentation Errata 55

### 6.1 Usage Caveats

#### Mounting the Workbench DVDs on Linux Hosts

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. Login 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.

### 32-bit Run-Time Libraries Necessary for Linux Hosts

In order to install Wind River Workbench On-Chip Debugging 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 the following hosts, you must install additional packages.

#### Ubuntu 8.04 x86-64

Install the required 32-bit run-time libraries using the following command:

```
sudo apt-get install ia32-libs
```

#### Fedora 9 x86-64

The 32-bit run-time libraries are installed with the base installation. To launch the installer in GUI mode, install the following additional packages:

```
sudo yum install gtk2.i386  
sudo yum install libXtst.i386
```

Additionally, to remove warnings from GTK that appear when **setup\_linux** is launched, you can install the following packages. Note that this is optional. The missing libraries do not affect the functionality of the installer.

```
sudo yum install gtk-nodoka-engine.i386  
sudo yum install bug-buddy.i386
```

### Do Not Use OCD Tools and WDB Agent at the Same Time

For any architectures that support hardware breakpoints, do not use On-Chip Debugging tools at the same time as Workbench debugging tools that use the Wind River Debug (WDB) target agent. Hardware breakpoint registers are accessed by both facilities, and simultaneous use may lead to unpredictable debugging behavior.

### Installing Wind River Probe Drivers Manually

If the Wind River Probe drivers do not install correctly when you install Workbench, you must install the drivers manually.

To install the drivers manually, open an explorer window and navigate to *installDir\ocd-3.x\version\host\x86-win32\USBDriverFiles*. Double-click on **WRProbeSetup.exe**. Continue clicking **Next** until the **Finish** button becomes enabled, then click **Finish**.

### Wind River Probe Drivers Not Backwards Compatible

Wind River Workbench 3.x uses Wind River Probe drivers that are not compatible with Probe drivers from Workbench 2.x releases. If you have installed Workbench 3.x, but you want to continue to use a 2.x release of Workbench with your Wind River Probe, you must download and install a patch from Wind River Online Support. The patch is available at the following link:

<https://portal.windriver.com/windsurf/products/hwTools/windRiverProbe/legacydriverpatches>

### Wind River Probe

- On Windows systems, Administrator privileges are required to install the drivers for the Wind River Probe.
- Windows XP users may see a screen prompt that states that the drivers for Wind River Probe are not logo-certified by Microsoft. This will not affect the performance of the Wind River Probe. You should continue when prompted to do so.
- The Wind River Probe must be connected to a high-powered USB port. If connected to a low-powered port, for example on a bus-powered USB hub, the Wind River Probe may not function correctly.

### Wind River ICE SX

Proper grounding may be required to prevent damage to the Wind River ICE SX and the target board. If the Wind River ICE SX and your target board are not at similar voltage potentials, damage can occur when you connect them. Potential differences occur when the AC and DC grounds of a target do not have a common reference. For proper grounding procedure, see the *Wind River ICE SX for Wind River Workbench Hardware Reference: Grounding the Target*.

Though the Remote Systems view allows you to connect the Wind River ICE SX and target server to a single board and use both connection types simultaneously, you must set the target server time-outs to longer default values to avoid the target server automatically disconnecting when the OCD connection stops the target.

For information about setting target server time-outs, see the *Wind River Workbench User's Guide*.

### Wind River ICE 2

- The ICE 2 does not currently support the following processor families:
  - PowerPC 5xx
  - PowerPC 8xx
  - ColdFire
  - XScale
- For the Wind River ICE 2, the operation of the target console port is not under firmware control. Therefore, the configuration option CF TGTCONS does not do anything, and should be ignored. Configure target console redirection using the Ethernet Setup menu, accessible by entering the command **ETHSETUP** at the **>NET>** prompt.
- To use target console redirection (TGTCONS) on the Wind River ICE 2, you must have either a standard DB9 connection on your target, or an adapter to convert the connection to a DB9. The ICE 2 expects the signals on the following pins:
  - RX – Pin 2
  - TX – Pin 3
  - GND – Pin 5
- The ICE 2 is configured to use DHCP by default. In the rare case where the ICE 2 fails to acquire an IP address from the DHCP server, you can reconfigure the ICE 2 to send additional requests for an IP address by using the following commands:

- To enable repeated retry requests:  
`>NET>fw_setenv netretry`
- To reset the ICE 2 to the default 3 requests:  
`>NET>fw_setenv netretry no`
- To make space on the ICE 2, remove files from the flash file system. This can be done from the `>NET>` prompt with the following commands:
  - To remove all firmware files already loaded:  
`>NET>rm $OCD_LIB/*.so`
  - To remove only temporary files:  
`>NET>rm /tmp/WindRiver/*`
  - To check the disk space available:  
`>NET>du -cm /`

Note that the total capacity of the flash file system is 256 MB.

- Any attempt to downgrade the ICE 2 platform to a version older than 0144 will render the unit inoperable and require a factory reload. Please contact the technical support department for assistance if you need to perform this type of downgrade.

#### Wind River Trace

- For trace records that do not have any symbol information, attempting to display trace in the **Functions** or **Source** code levels causes very slow record trace view updating, negative time stamping, and improper text highlighting. For these kinds of records, display the trace in **Disassembly** code level only.
- Although the **Address** field in the Tracepoint dialog is configurable, this parameter has no effect on resulting trace capture for any filtered trace event type. Ignore this field during tracepoint configuration.

#### Wind River Trace 2

- The current release of the Wind River Trace 2 only supports the ETMv1 Single Connector as specified in the *Embedded Trace Macrocell Specification IHI 00140*.
- Demux and Mux clock modes are not supported.
- You must use version **pr3.0a** or later of the ARM9 firmware to use the Trace 2. Older versions of the ARM9 firmware will not work. To see the versions of loaded firmware files, connect to the Wind River ICE 2 and enter the command **QUERY** at the `>NET>` prompt.

#### Internal Software Trace

- For PowerPC 8641D targets: the 8641D is a dual-core processor; Workbench can trace either core separately, but it cannot trace both cores at the same time.
- For PowerPC 86xx targets using Revision 1.0 silicon, the **Source ID** and **Target ID** match criteria do not filter trace results correctly; the trace will return all cycles, not just the matched cycles. This is a processor limitation.

## Flash Programming

- On targets that have no flash code and no HRESET signal available, the emulator may not be able to enter background mode. For example, on MIPS Broadcom 7401 targets that have no flash code, EJTAG is not enabled until a delay after reset is released. During this delay, the processor fetches an (invalid) instruction from the blank flash and attempts to vector to a (non-existent) exception handler (which, by default, is also located in the blank flash and contains nothing but invalid instructions.) The exception priority for this invalid instruction fetch is higher than the priority of the debug exception once the delay has expired and the EJTAG has been activated, so the target cannot enter background mode.
- Flash programming cannot be interrupted prior to completion.

## Compiling Linux Without Debug Symbols

Workbench cannot show application process threads when the Linux kernel is compiled without debug symbols, because Workbench cannot properly debug processes if it does not have debug information for the kernel.

Since Workbench must be able to debug the stock Linux kernel, it cannot rely on special symbols to get offsets. The offsets also cannot be hard-coded, since Workbench would lose flexibility with regard to the way kernels are compiled.

## Reset and Download

- If you list the same file in the **Download** tab several times, but with different offsets specified for each, Workbench does not load symbols for any of the files except the last one in the list. You can make a copy of the file with a different name; then Workbench loads symbols for both.
- The Reset and Download view will detect if you have `<TABLE_TIED_RESET>ON</TABLE_TIED_RESET>` in the board file, and the reset command issued will be the **RST** command instead of the **IN** command. You can use the board file editor to set this option.
- To use the Reset and Download view for multi-core debugging:
  1. Set `TABLE_TIED_RESET` to `ON` in the board file.
  2. Set `CF HRESET` to `ENABLE`.
  3. Set `CF CMDRST` to `RST`.
- Image file download cannot be interrupted prior to completion.

## Board Files Not Always Necessary for All Target Boards

Wind River ships default board files for all Wind River target boards. For most targets, you do not need to use a board file unless you have more devices than just the processor on your scan chain; you can just select the processor variant in the connection wizard.

However, board files are necessary for some architectures, such as ARM11 targets. Workbench includes board files for some common ARM11 targets, such as the ARM1136 Integrator/CP board and the TI OMAP2430 SDP board.

For a list of all Wind River-supplied board files, see *Application Note 313: Register Files/Board Files Cross-Reference* at

<https://portal.windriver.com/windsurf/appnotes/hwTools/index.html>.

### Playing Back Register Files

If you are debugging a single core, you need to play back a register file only once. The values are stored in emulator memory and the register settings are retained until you switch to a new processor.

If you are debugging multiple cores: when you use a board file, each core gets its own register settings and they are associated with the board file name you choose. If you switch to a new board file, you need to play back the register files on each core to set up emulator memory again.

Register file playback cannot be interrupted prior to completion.

### Setting Chip Selects Not Necessary on All Processors

The *Wind River Workbench On-Chip Debugging User Tutorials* describes setting chip selects and reset configuration words. This is necessary only on some processors.

### Intel Xeon Cook Forest Reference Board

Because of the electrical properties of the Intel Xeon Cook Forest reference board, Workbench cannot make a connection using the default clock phase of 0. In order to connect Workbench to a Cook Forest reference board, you must set the clock phase to +90.

When you make a connection attempt using the default settings, the connection fails and an >ERR> prompt appears in the OCD Command Shell.

At the prompt, enter the following command:

```
>ERR>hic clkphase +90
```

You can now connect to the reference board.

### Intel Xeon Hanlan Creek board

- To connect to an Intel Xeon Hanlan Creek board using a Wind River ICE 2, you must at minimum connect to both C0\_0 and C4\_0 (if you disable multithreading and only enable a single core on each processor in the BIOS.)
- If the board is halted for 90 seconds (in background mode) it causes the platform to reboot. If this happens, use the following commands to prevent rebooting of the board:

```
>BKM>SCGA ON HALT BIOS_SMI 00000430 00000000 ON_HALT /ue:0 /no_in  
>BKM>CF GRP ON_HALT ENABLED
```

### ARM Targets

- For ARM9 MV88F5281 processors, the JTAG interface is disabled when the device is in a Wait for Interrupt state. This is by design, in order to save power, but it also prevents an emulator from taking control of the CPU.

As a workaround, either cancel the Wait for Interrupt state when the debugger is needed, or continuously assert an FIQ/IRQ interrupt signal to exit Sleep mode.

- If you are using a TI OMAP2430 SDP evaluation board, note that the primary RAM region begins at address 0x80000000. The demo applications supplied within Workbench are pre-built for the customary ARM starting addresses of 0x00000000, 0x04000000 and 0x08000000. In order to use a demo application

with the OMAP2430 SDP evaluation board, you will need to re-build the demo project with a starting address of 0x80000000.

- For ARM11 MPCORE targets, JTAG debugging requires that the **CONFIG** switch be set to **OFF**. If the **CONFIG** switch is set to **ON**, attempting to connect to the target returns an error message.

#### **MPC55xx and 56xx Targets**

- For MPC55xx and MPC56xx targets, the internal SRAM must be initialized before access, using the low-level command **SY INTSRAM INIT**. You must perform this initialization whenever a power-on reset occurs.
- To erase and program the internal data flash for MPC56xx targets, you must set the JTAG clock to 3MHz. To do this, use the low-level command **CF CLK 3**.

#### **Older Evaluation Boards may Not Enter Background Mode**

Some older Freescale MPC5516EVb evaluation boards may not enter background mode when using an auto-voltage paddle board, due to an overshoot/undershoot/reflection issue on the TCK signal. (Newer Freescale MPC5516EVb targets do not have this problem.)

#### **Cannot Set Software Breakpoints During a Reset and Download for Intel Targets**

Planting software breakpoints during a reset and download operation causes download and initialization failures. This is because when a **GO** command occurs after an **INN** command, the firmware tries to install the software breakpoints in uninitialized memory, causing the board initialization to fail.

Because of a non-disclosure agreement with Intel, Wind River does not distribute register files for Intel targets. Since a register file can't be used for initialization, the firmware can't know when it's valid to use software breakpoints.

You can either disable all software breakpoints during a reset and download, or else use hardware breakpoints instead. (There is a limit of four hardware breakpoints.)

## **6.2 Known Problems**

For a list of known problems in Wind River Workbench, visit the Online Support Web site (see [1.1.3 Latest Release Information](#), p.2).

### **6.2.1 Firmware-Specific Issues**

- For PowerPC 8xx processors, the Trigger Out pin is not activated when a breakpoint is hit. (WIND00092464)
- For ARM9 OXETHU954 processors, the Cache view incorrectly returns "invalid number of arguments" errors. (WIND00086996)
- For Cavium OCTEON processors, the OCD Console does not always automatically open to show the progress of a reset and download operation.

The workaround is to select **Window > Show View > OCD Console**.  
(WIND00125317)

- When a PowerPC 8641D target is configured to boot from PCI, OCD run control does not function properly. Setting the program counter to a location in RAM then issuing a single step command returns the PC back to the reset vector. An additional single step at this point reports an “Unable to step” error.(WIND00188975)
- For Freescale i.MX31 targets, the supplied register file does not properly configure the target RAM. The workaround is to use the internal RAM address 0x1000\_0000 for the flash workspace. (WIND00194734)
- For Freescale i.MX51 targets, the flash algorithm does not end properly. (WIND00191983)
- For MIPS RM9000 targets, hardware data breakpoints do not break at the expected address on a write access. (WIND00189775)

### 6.2.2 Wind River ICE 2 Issues

- When using a Wind River ICE 2 to connect to all 16 cores on a Cavium 3860 target, Workbench can take up to two minutes to complete all 16 connections. (WIND00123541)
- The Wind River ICE 2 trigger input feature sometimes generates false trigger events in certain cases. As a workaround, set the configuration option TRGINFILTER to ON, either by using the CF Options view or by entering the command **CF TRGINFILTER ON** at the **>BKM>** prompt in the OCD Command Shell. (WIND00166543)
- OCD downloads do not work when the **CF CLK** option is set to **0.025**. (WIND00172469)
- When attempting to use Wind River System Viewer over a TMD connection using the ICE 2, the target takes exceptions. (WIND00166040)
- If you have an Intel adapter board plugged in without the ICE 2 powered on or connected, the Intel board will not boot. (WIND00169314)
- Ethernet over USB on the ICE 2 does not reliably support disconnect/reconnect to a Windows host; if you disconnect and reconnect several times, the port will stop working, and you must reboot the ICE 2 to reconnect. (WIND00168366)
- On PowerPC 8641D targets, the **Add Hardware Breakpoint** button is greyed out and unavailable. The workaround is to create a regular breakpoint, select properties, and enable the hardware flag. (WIND00140929)
- When an ICE 2 is connected to a PowerPC 8641D target that is configured for dual-core operation with a VxWorks SMP image, it prevents the Wind River Memory Analyzer from operating properly. (WIND00175888)

### 6.2.3 Wind River Trace Issues

- For AMCC440GX-800 targets, when both instruction and data cache are enabled on the target, the trace data generated by the processor is incorrect.

Attempting to reconstruct the trace from these values results in an invalid address. This only occurs when the instruction and data cache are enabled. (WIND00090468)

- If you create a tracepoint with invalid information – for example, an incorrect source file name – Workbench displays a warning message with a button to allow you to edit the tracepoint properties, but the button is disabled. The workaround is to open the tracepoint in the Breakpoints view and edit its properties from there. (WIND00125896)
- On AMCC 405EX targets, trace is unreliable at 400 MHz. To use trace you must slow the board down to 333MHz. (WIND00170325)
- On AMCC 460EX targets, the Start of Trace button sometimes returns a warning message even though the symbol file was downloaded correctly. (WIND00145572)
- For ColdFire 5329 targets, setting a tracepoint anywhere in the code causes an exception if the tracepoint is hit. (WIND00170292)
- For AMCC 405EX targets, using the scroll bar in the Trace view results in random occurrences of “No instruction found” in trace. Click on Start of Trace or End of Trace to reset the displayed trace information. (WIND00123249)

#### 6.2.4 Wind River Trace 2 Issues

- When trace capturing stops (either because of a breakpoint, or a **Trace Off**, or reaching the end address of a **Trace Until Address**) the last two or three instructions are sometimes missing. (WIND00137513)
- Event icons do not appear in the Trace view if there are no symbols matching the event address. The workaround is to scroll down a few lines to see the event occurrence (exception vector address). (WIND00138387)
- After a reset and download and a Trace view refresh, the Trace pane does not clear. The workaround is to use the **Clear Trace Buffer** button after every reset and download. (WIND00138719)
- In the Trace Configuration dialog, many of the parameters under the **Advanced** node do not have associated descriptions. (WIND00135510)
- Event timestamps do not always align with absolute timestamps. (WIND00137142)
- When capturing a partial trace record (fewer than 512 bytes), both absolute and execution timestamps are set to zero. The workaround is to make sure the trace capture exceeds 512 bytes. (WIND00138815)
- When using an ICE 2 and Trace 2 with a ARM926e target, the EDGELO and LEVELLO parameters for the configuration option **CF TRIGINFILTER** do not work. (WIND00148556)
- The Trace 2 sometimes reports several Undefined Exception errors when only one has occurred. (WIND00146977)
- The Trace view does not fully clear after a reset and download and a Trace view refresh. To clear trace after a reset and download, use the **Clear Trace Buffer** button. (WIND00138719)

## 6.2.5 OCD Issues

### Cache View

- On AMCC 440EPX targets, the Cache view incorrectly shows that the cache is enabled after every reset. To enable cache you must click the **Enable Instr Cache/Enable Data Cache** button. (WIND00090404)
- For MPC8349 targets, the Cache view only shows four available WAYS, although there are actually eight available. To see all available WAYS, use the **CA DUMP** command in the OCD Command Shell. (WIND00091223)
- On some ARM targets, in the **Advanced Control and Status** tab, the **Lock a Way**, **Unlock a Way**, and **Invalidate a Line** buttons cause an error to be displayed. This happens in both the Instruction and Data cache. As a workaround, use the **CA** commands in the OCD Command Shell. The command **CA** with no arguments returns a list of all available cache commands. (WIND00086996)
- The Cache view only displays the lower 32 bits of 64-bit addresses. The workaround is to enter the command **CA READ -D** (for the data cache) or **CA READ -I** (for the instruction cache) in the OCD Command Shell. (WIND00106002)
- When connected with a Wind River Probe to a BCM7401 target, and when the instruction cache is the active sub-view in the cache view, the **Refresh All** button only refreshes the instruction cache; the data cache is not refreshed. To refresh the data cache you must switch to the data cache sub-view and use the **Refresh** button. (WIND00121460)

### Convert Utility

- The Convert utility does not convert images built for SPARC processors. (WIND00110973)

### Data Breakpoints

- The **Continue on Delay** feature does not work for hardware data breakpoints. (WIND00063406)

### Flash Programming

- The Custom Flash Driver example project does not build properly on Linux for non-PowerPC builds. Other hosts fail due to a header file not being found, because one of the include paths is incorrect. The default include path is **-I./Include** when it should be **-I./include** (the path is case sensitive).  
The workaround is to change/add a build path from **Properties > Build Properties > Build Paths** to include **-I./include** (use a lower-case “i” in **include**). (WIND00170421)

### Internal Software Trace

- The Trace view sometimes returns extraneous “garbage” lines in the trace output. These lines can be ignored. (WIND00105742)

### JTAG Board File Editor

- On Linux hosts, when creating a JTAG board file for use with multi-core targets and the Wind River ICE SX, board files created using the **Custom** option are not always laid out correctly. Wind River recommends that you start with one of the pre-defined scan chains: Single Core, Dual Core, 3 Core, or 4 Core.
- Custom board files created for MPC8641D processors do not work. The workaround is to use one of the supplied board files for this processor, located at *installDir/ocd-3.x/version/BoardFiles/PowerPC/86xx*. (WIND00104802)
- The JTAG Editor may not contain an entry for all supported targets. The workaround is to create a board file manually using XML, as described in the *Wind River Workbench for On-Chip Debugging User Tutorials: Using Board Descriptor Files*. (WIND00109970)

### Memory View

- The Memory view does not display a warning when you enter invalid input. (WIND00092410)
- If you use the Memory view to export the data from a specified memory range, the resulting S-Record file cannot be downloaded using the Reset and Download view. The workaround is to use the Binary Upload view to export the memory range as a binary file. (WIND00126055)

### OCD Console

The Progress bar does not always refresh consistently when playing a register file. (WIND00124895)

### Registers View

- For MIPS64 targets, the Registers view does not display the correct decimal value for single-precision floats in 64 bit mode (FR bit in CP0 status register equals 1), or for single- or double-precision floats in 32 bit mode. (FR bit in CP0 status register equals 0). (WIND00067734)
- For WinPath2 targets, the “Save Target Register Values to File” menu option in the Registers view does not copy the target values correctly. (WIND00189775)
- For PowerPC 5200 targets, the Registers view displays some incorrect bit-level field names. (WIND00158235)
- Bit-level detail is not displayed for OMAP 3430 targets. (WIND00145201)
- Bit-level detail is not displayed for Cavium CNS110x targets. (WIND00178131)

### Reset and Download View

- Workbench sometimes does not set a breakpoint on the symbol **sysInit** on connection, even when you have specified it in the Reset and Download view. The workaround is to connect to the target first, then use the Breakpoints view to set a breakpoint on **sysInit**. (WIND00056268)
- When connected to a multicore target, clearing the checkbox for the core node does not automatically clear all the checkboxes in that core’s tree. You must clear them manually. (WIND00125588)

- If you change settings in the Reset and Download view, and then close without saving, some of the option checkboxes are not displayed when you reopen. (WIND00165959)

### Terminal View

On Linux and Solaris, you must have appropriate permissions to connect to **ttyS0** and **ttyS1** using the Terminal view; or, alternately, **ttyS0** and **ttyS1** must have appropriate permissions set to allow you to connect to them.

The Terminal view does not support terminal emulation when used in Telnet mode.

### Transparent Mode Driver

The Transparent Mode Driver is not supported for Linux targets in this release.

### VxWorks 5.5 Debugging

When using Workbench with Tornado 2.x tools over an OCD connection, the Tornado 2.x registry is shut down and only the Workbench registry is accessible.

### Multicore Debugging

- When you change your source code, recompile, and download to a core in a multi-core environment, an error message appears prompting you to unload the symbols first; but the menu pick for **Unload Symbols** in the Remote Systems view is inactive. This only happens when the symbol file is loaded into a multicore context: recompiling results in a new file that can only be reloaded in a core context if all other cores have unloaded the file first.

As a workaround, either do a second reset and download operation after the initial **Unload Symbol** error, or unload the symbol file in all attached debug contexts. (WIND00091452)

- Special issues for multi-core debugging with the MIPS64 BCM1480 Quad-Core are described in *Wind River Application Note 314: Multi-Core OCD: Using Workbench Command Shell for BCM 1480 Quad-Core 64-bit MIPS Processor* at <https://portal.windriver.com/windsurf/appnotes/hwTools/index.html>.

### Miscellaneous

- On Intel Nehalem Xeon 55xx targets, using the default JTAG clock rate, debugging the file **mthread.out** using an ICE 2 results in program-context errors. The workaround is to change the JTAG clock speed to 8 by using the command **CF CLK 8** in the OCD Command Shell. (WIND00196114)
- On MIPS targets, the initialization process does not detect whether the paddle board is connected correctly. (WIND00068744)
- For PPC8572 targets, when using Wind River Hypervisor with a Wind River Linux virtual board, the Wind River ICE 2 triggers improperly on some interrupts. (WIND00180450)
- For MPC8560 targets, when using the Instruction Set Simulator, the Debug view does not display the entire stack frame list. (WIND00180721)
- When connecting to a Linux target using **u-boot**, the target sometimes takes a TLB miss when it enters location zero (the start of the Linux kernel.) A workaround is to re-play the target's register file. (WIND00135257)

- On PPC86xx targets, snooping is halted when the core enters a freeze state due to the JTAG connection. (The freeze state is necessary for accessing SPR registers.) This results in cache coherency issues when packets are DMA from external devices while the core is halted, because the packet is not properly snooped into memory. As a workaround, either allocate the DMA buffer in uncached areas, or disable the data cache when debugging with OCD. (WIND00172460)
- When using the **UPLOADBIN** command in an OCD script, it will not allow you to specify a large area of memory. Either use the Binary Upload view, or use several **UPLOADBIN** commands with the **append** option. (WIND00153249)
- If you use an S-Record file to build a **bootrom.hex** or **bootrom\_uncmp.hex** image, and try to download the file into writable memory, Workbench returns an error. The workaround is to use the Convert utility to convert the \*.hex file to a \*.bin file and then download that. (Note that the \*.bin file must be a Wind River formatted binary file, not a raw binary file.) (WIND00098096)
- If you click the **Link Cores** button in the Debug view while the target is running, Workbench does not display an error message to tell you that is not allowed. The target continues to run and the cores are not linked. The workaround is to enable **Display Background Communications** in the OCD Command Shell. With this option enabled, clicking **Link Cores** while the target is running returns the following output in the OCD Command Shell:

```
>RUN>sctrl all enable all  
!ERROR! - [msg32052] This command requires the target to be stopped
```

(WIND00103044)

## 6.3 Documentation Errata

For a detailed list of documentation errata for Wind River Workbench, visit the Online Support Web site (see [1.1.3 Latest Release Information](#), p.2).



# 7

## *Processor Support Matrix*

7.1	Introduction	57
7.2	PowerPC Processors	58
7.3	QorIQ Processors	61
7.4	ColdFire Processors	61
7.5	MIPS Processors	62
7.6	ARM Processors	63
7.7	XScale Processors	64
7.8	Cavium Processors	64
7.9	PA Semi Processors	65
7.10	SiByte Processors	65
7.11	RMI Processors	65
7.12	Intel Processors	65

### **7.1 Introduction**

This chapter lists the processors supported by this release of Wind River Workbench On-Chip Debugging.

## 7.2 PowerPC Processors

Table 7-1 **PowerPC 4xx**

405CORE	405EX	405L	405STB3
403GCX	405EXR	405H	405STB4
405CR	405GP	405PM	X2VPCORE
405EP	405GPR	405STBX25	

Table 7-2 **PowerPC 44x**

440CORE	440GX-800	440SP	440SPE
440GP	440EP	440GR	460EX
440GX	440EPX	440GRX	460GT

Table 7-3 **PowerPC 5xx**

MPC505	MPC560	MPC565
MPC509	MPC561	
MPC555	MPC563	

PPC5xx processors are not supported for the Wind River ICE 2.

Table 7-4 **PowerPC5xxx**

MPC5200	MPC5200B	MPC5121E
---------	----------	----------

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-5 **PowerPC 55xx**

MPC5514 (*)	MPC5517 (*)	MPC5553	MPC5566
MPC5515 (*)	MPC5533	MPC5554	MPC5567
MPC5516 (*)	MPC5534	MPC5565	

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-6 **PowerPC 56xx**

SPC560B50	MPC5602	MPC5606S
SPC560S60	MPC5603	MPC5607B
	MPC5604	MPC5607S

PPC56xx processors are not supported for the Wind River ICE SX.

Table 7-7 **PowerPC6xx**

PPC603E	MPC603P	PPCEC603E
MPC603E	PPC603R	MPCEC603E
PPC603P	MPC603R	

Table 7-8 **PowerPC 7xx**

PPC740	PPC750	PPC750CX	PPC750FX
MPC740	MPC750	PPC750CXE	PPC750GX
PPC745	PPC755	PPC750FL	PPC750GL
MPC745	MPC755	PPC750L	

Table 7-9 **PowerPC 74xx**

MPC7400	MPC7445	MPC7450
MPC7410	MPC7447	MPC7451
MPC7440	MPC7447A	MPC7455
MPC7441	MPC7448	MPC7457

Table 7-10 **PowerPC 8xx**

MPC801	MPC850SE	MPC860DH	MPC862P
MPC821	MPC852T	MPC860DP	MPC862SR
MPC823	MPC855T	MPC860DT	MPC862T
MPC823E	MPC857DSL	MPC860EN	MPC866P
MPC850	MPC857T	MPC860MH	MPC866T
MPC850DC	MPC859DSL	MPC860P	MPC870
MPC850DE	MPC859T	MPC860SAR	MPC875
MPC850DH	MPC860	MPC860T	MPC880
MPC850DSL	MPC860DC	MPC862DP	MPC885
MPC850SAR	MPC860DE	MPC862DT	

PPC8xx processors are not supported for the Wind River ICE 2.

Table 7-11 **PowerPC 82xx**

MPC8220	MPC8247	MPC8260	MPC8270	MPC8280
MPC8240	MPC8248	MPC8264	MPC8271	
MPC8241	MPC8250	MPC8265	MPC8272	
MPC8245	MPC8255	MPC8266	MPC8275	

Table 7-12 **PowerPC 83xx**

MPC8313	MPC8323	MPC8349	MPC8377
MPC8313E	MPC8323E	MPC8349E	MPC8377E
MPC8314	MPC8343	MPC8358	MPC8378
MPC8314E	MPC8343E	MPC8358E	MPC8378E
MPC8315	MPC8347	MPC8360	MPC8379
MPC8315E	MPC8347E	MPC8360E	MPC8379E
MPC8321			
MPC8321E			

Table 7-13 **PowerPC 85xx**

MPC8533 (*)	MPC8544	MPC8555	MPC8568
MPC8533E (*)	MPC8544E	MPC8555E	MPC8568E
MPC8536 (*)	MPC8545	MPC8560	MPC8569 (*)
MPC8536E (*)	MPC8545E	MPC8565	MPC8569E (*)
MPC8540	MPC8547	MPC8565E	MPC8572
MPC8541	MPC8547E	MPC8567	MPC8572E
MPC8541E	MPC8548	MPC8567E	
MPC8543	MPC8548E		
MPC8543E			

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-14 **PowerPC 86xx**

MPC8610	MPC8641
MPC8640 (*)	MPC8641D
MPC8640D (*)	

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

## 7.3 QorIQ Processors

Table 7-15 **Freescal QorIQ**

P2010	P4080
P2020	

QorIQ processors are not supported for the Wind River ICE SX.

## 7.4 ColdFire Processors

Table 7-16 **ColdFire**

MCF5202	MCF5249	MCF5307	MCF5475	MCF52230
MCF5204	MCF5249L	MCF5307A	MCF5480	MCF52231
MCF5206	MCF5250	MCF5327	MCF5481	MCF52232
MCF5206E	MCF5251	MCF5328	MCF5482	MCF52233
MCF5207	MCF5253	MCF5329	MCF5483	MCF52234
MCF5208	MCF5270	MCF5372	MCF5484	MCF52235
MCF5211	MCF5271	MCF5372L	MCF5485	MCF52236
MCF5212	MCF5272	MCF5373	MCF52100	MCF54450
MCF5213	MCF5274	MCF5373L	MCF52110	MCF54451
MCF5214	MCF5274L	MCF5407	MCF52210	MCF54452
MCF5216	MCF5275	MCF5470	MCF52211	MCF54453
MCF5232	MCF5275L	MCF5471	MCF52212	MCF54454
MCF5233	MCF5280	MCF5472	MCF52213	MCF54455
MCF5234	MCF5281	MCF5473	MCF52221	
MCF5235	MCF5282	MCF5474	MCF52223	

ColdFire processors are not supported for the Wind River ICE 2.

## 7.5 MIPS Processors

Table 7-17 **MIPS32**

4KC	PR4450	BCM1113	BCM4704	BCM7401
4KM	PNX3001	BCM1113R	BCM4710	BCM7405
4KP	PNX7350	BCM1115	BCM5365	BCM7100
4KEC	PNX832x	BCM1115R	BCM56214	BCM7115
24KC	PNX852x	BCM1190	BCM56218	BCM7312
24KF	PNX8535	BCM3349	BCM5836	BCM7318
AU1000	PNX8550	BCM3350	BCM6348	BCM7335 (*)
AU1200	BCM1100	BCM3351	BCM6550A	WinPath2 (*)
AU1300 (*)	BCM1101	BCM3352	BCM7400	Altera_MP32 (*)
PR1910	BCM1103	BCM3360		
PR3940	BCM1104	BCM3560		

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-18 **MIPS64**

5KC	TX4925	CN3110	CN5540
5KF	TX4927	CN3120	CN5645
20KC	TX4955	CN3630	CN5740
25KF	RM7900	CN5220	CN5745
VR4131	RM9150	CN5230	CN5750
VR4133	RM9000X2	CN5434	
VR4181A	BCM7038	CN5440	
VR5432	BCM7320	CN5534	
VR5500			
VR5500A			
VR7701			

## 7.6 ARM Processors

Table 7-19 **ARM9**

ARM9TDMI	AT91RM9200	MC9328MX1 (FSL i.MX1)
ARM920T	AT91SAM9260	MC9328MXL (FSL i.MXL)
ARM922T	AT91SAM9261	MC9328MXS (FSL i.MXS)
ARM940T	OXETHU954	MC9328MX21 (FSL i.MX21)
ARM926E	TMS320DM6441	MC9328MX27 (FSL i.MX27)
ARM946E	TMS320DM6443	FSL i.MX25 (*)
MV88F5181	TMS320DM6446	M82515 (Mindspeed 920 dual-core)
MV88F5281	MP201 (ARM926E core) (*)	EPXA (ARM922 based core) KS869SPX (ARM922 based core)

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-20 **ARM11**

ARM1136	FSL i.MX31
ARM1176 (*)	FSL i.MX35 (*)
MPCORE	
Medy2	
TI OMAP2430	

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

Table 7-21 **ARM CORTEX**

LM3S801 (Cortex-M3)	TI OMAP3410 (*)	TI OMAP3530 (*)
CORTEXA8	TI OMAP3420 (*)	FSL i.MX51 (*)
	TI OMAP3430 (*)	

\* Processors marked with an asterisk (\*) are not supported for the Wind River ICE SX.

## 7.7 XScale Processors

Table 7-22 **StrongARM**

IOP310	PXA210	IXP421	IXP465	IXP2401 (IXP2400 with encryption)
IOP321	PXA250	IXP422	IXP2325	IXP2800
IOP331	PXA255	IXP425	IXP2350	IXP2801 (IXP2801 with encryption)
IOP333	PXA270	IXP450	IXP2351	IXP2850
IOP341	PXA300	IXP451	IXP2400	IXP2851 (IXP2850 with encryption)
IOP342	PXA310	IXP452		
IOP348	PXA320	IXP455		
	IXP420	IXP460		

## 7.8 Cavium Processors

Table 7-23 **Cavium OCTEON**

CN3005	CN5010	CN5830
CN3010	CN5020	CN5840
CN3830	CN5640	CN5850
CN3840	CN5650	CN5860
CN3850		
CN3860		

Table 7-24 **Cavium Econa/STAR**

CNS1102	CNS2131	CNS2181
CNS1104	CNS2132	CNS2182
CNS1105	CNS2133	

Econa/STAR processors are not supported for the Wind River ICE SX.

## 7.9 PA Semi Processors

Table 7-25 PA Semi

PA6T-1682M
------------

## 7.10 SiByte Processors

Table 7-26 SiByte

BCM1112	BCM1250	BCM1455
BCM1122	BCM1255	BCM1480
BCM1125	BCM1280	
BCM1125H		

## 7.11 RMI Processors

Table 7-27 RMI

XLR716	XLS416
XLR732	XLS616

RMI processors are not supported for the Wind River ICE SX.

## 7.12 Intel Processors

Intel processors are not supported for the Wind River ICE SX.

Table 7-28 Intel Atom

230	Z500	Z530
n270	Z510	Z540
330	Z515	Z550
	Z520	

Table 7-29 **Intel Nehalem**

---

Xeon 55xx

---

Table 7-30 **Intel Core 2 Duo**

---

T9400                  L7400

---

# 8

## Uninstalling

Wind River recommends that you uninstall the products in this release using the uninstaller provided with your installation. If you choose to uninstall the products manually (that is, by manually deleting the product directories), you must also remove the InstallShield VPD (InstallShield database). If the VPD is not removed, your system will be left in an inconsistent state which can result in a variety of system problems.

To remove the InstallShield VPD, complete the following steps:

### On Windows hosts:

1. In a command shell, change to the InstallShield directory as follows:

```
C:\> cd Program Files\Common Files\InstallShield\Universal\WindRiver
```

2. Execute the **dir** command to see a directory listing.

```
% dir
```

This should show a list of directories corresponding to the products you have installed. For example, **Wind River Workbench 3.x** or **VxWorks 6.x**.

3. Remove the product directories that correspond to the products you manually removed from your installation.

```
% del Wind River Workbench 3.x
```



---

**NOTE:** Delete only those directories that correspond directly to the product or products you removed manually.

---

### On Linux and Solaris hosts:

1. Change to the InstallShield directory as follows:

```
% cd $HOME/InstallShield/Universal/WindRiver/
```

2. Execute the **ls** command to see a directory listing.

```
% ls
```

This should show a list of directories corresponding to the products you have installed. For example, **Wind River Workbench 3.x** or **VxWorks 6.x**.

3. Remove the product directories that correspond to the products you manually removed from your installation.

```
% rm -r Wind River Workbench 3.x
```



---

**NOTE:** Delete only those directories that correspond directly to the product or products you removed manually.

---

## 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](http://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)	<a href="mailto:support@windriver.com">support@windriver.com</a>
Europe, Africa, Middle East	+(00) 800-4977-4977 (toll-free)	<a href="mailto:support-EC@windriver.com">support-EC@windriver.com</a>
Japan	81-3-5778-6001	<a href="mailto:support-jp@windriver.com">support-jp@windriver.com</a>

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.