

## Wind River Workbench 3.3

Wind River Workbench 3.3 is a collection of tools based on the Eclipse framework that accelerates time-to-market for developers building devices with Wind River Linux and VxWorks platforms. Workbench offers the only end-to-end, open standards-based suite for device software design, development, debugging, test, and management. Through its powerful combination of capabilities, integration, and availability, Workbench enables organizations to standardize on a common environment for device software development, helping developers, project teams, and enterprises improve their effectiveness.

Workbench offers the following:

- Best-in-class capability at each phase of the development process, including hardware bring-up, firmware development, application software development, advanced visualization, system diagnostics, and test
- Broad availability to support increased standardization across projects
  - Multiple-target OS support, including support for VxWorks 5.5, VxWorks 6.x, Wind River Linux, and Wind River Hypervisor
  - Target processor support for ARM, ColdFire, Intel Architecture/Pentium, MIPS, PowerPC, SH, and XScale processors

- Target simulators for VxWorks and Wind River Linux development on Windows, Linux, and Solaris
- Plug-in architecture, enabling additional target OS, target processor, and target connection support to be added by Wind River Professional Services

With an extensible framework to seamlessly integrate third-party and in-house plug-ins for total customization and scalability, Workbench addresses the challenges individual developers and project teams face by increasing productivity, enabling collaboration between hardware and software developers, and meeting diverse development

needs across an enterprise. The Workbench development suite is backed by Wind River's reputed device software industry experience, a world-class support organization, and a specialized professional services team.

### General Features

**Workbench Welcome Experience:** Wind River Workbench 3.3 provides a task and workflow-based Getting Started component that guides developer actions and provides links to important web content and online documentation. The Workbench Welcome Experience is shown when the user starts Workbench for the first time and can remain open as a reference while users are at work. Also included is an integrated RSS reader with preconfigured links to Wind River RSS feeds. The RSS feeds announce the availability of updates for Workbench, tools, board support packages (BSPs), device drivers, and platforms. A streamlined new Basic perspective helps new users get up to speed quickly with Workbench.

**Combined export and import:** Combined export and import makes it possible to prepare a single ZIP archive of projects along with related view and perspective setup, target connections, working sets, preference settings, and more. A new user can easily join an existing team and begin working in the established development environment by importing a single combined workspace ZIP package exported by another team member.

### Table of Contents

General Features .....	1	Run-Time Analysis Tools .....	12
VxWorks Features .....	2	Optional Add-Ons .....	14
Wind River Linux Features.....	2	Technical Specifications .....	15
Application Components.....	3	Host OS Support .....	15
Eclipse .....	3	Target OS Support .....	15
C/C++ Development Tooling .....	4	Target Processor Support .....	15
Target Management/Remote		Target Architecture Support.....	15
System Explore .....	4	Workbench On-Chip Debugging	
Project System.....	4	Target Architecture Support.....	15
Build System.....	5	Professional Services .....	15
Index-Based Global Text		Workbench Acceleration	
Search-and-Replace .....	5	Services .....	15
Compilers .....	5	Workbench Adoption Services.....	15
Debugger .....	6	Technical Services.....	15
Simulators.....	8	Education Services .....	15
Shell Environments.....	8	Installation and Orientation	
Configuration Tools .....	10	Service .....	16
		Support Services .....	16

**Command-line import and export:**

Command-line options instruct Workbench to open the Combined Workspace Import wizard as soon as the Workbench window opens. This is useful if, for example, you want to distribute settings among team members so they can begin work in a particular perspective and with specific projects without having to spend time configuring their own workspaces.

**Eclipse capabilities:** Through the Eclipse CDT project, Workbench adopts industrywide standards, making it more interoperable with third-party and homegrown Eclipse-based tools that leverage these standards. Workbench 3.3 is based on the most current Eclipse 3.6 Helios release.

**Rapid enterprise-scale development:**

Wind River Workbench is designed to perform in a wide range of project sizes, from individual developers to large teams building complex applications. Workbench is tuned to deliver maximum productivity through highly usable, perspective-driven workflows tailored to specific developer needs.

**Workbench debug mode:** Workbench includes a simplified mode for debugging with a streamlined user interface and optimized workflows that give users the flexibility of using it in concert with their external source code editors and build systems.

**VxWorks Features**

**GCC compiler:** The GCC 4.3.3 compiler can be used to build, debug, and analyze VxWorks 6.9 projects.

**Intel C++ Compiler for VxWorks:**

Compile Intel architecture 32-bit and 64-bit VxWorks applications on the VxWorks Simulator and real targets with Workbench.

**System Viewer support:** Wind River System Viewer features a VxWorks Source Build (VSB) configuration option so users can select a performance-oriented implementation as opposed to

a deterministic-oriented one. They can use a nondeterministic spinlock instead of a real-time spinlock. For other tools improvements see the “Run-Time Analysis Tools” section.

**VxWorks Source Build configuration**

**tool:** Wind River Workbench 3.3 provides a project type that allows the configuration and building of custom VxWorks libraries, which can be used as the basis for VxWorks image projects. This project type supports the creation and management of VxWorks Source Build (VSB) projects. These projects are based on a specific BSP and allow users to tune build options and content to match a specific application. The selected options are used to rebuild the VxWorks run-time sources to fit that configuration. Though it is not mandatory for customers to have VxWorks source code, VSB projects can only be set up if the sources are installed.

**Multi-context debugger support for**

**VxWorks AMP and SMP:** The Workbench debugger supports VxWorks debugging for single-core processors as well as multi-core processors running VxWorks in either symmetric multi-processing (SMP) or asymmetric multi-processing (AMP) mode. Multi-context debugging capabilities allow developers to debug code running in multiple contexts simultaneously. Full OS-awareness of VxWorks processes enables the developer unparalleled insight into code concurrency deadlocks and race conditions.

**ARM thumb-2 modes:** VxWorks applications built in ARM thumb-2 modes can be debugged with the Workbench debugger.

**SMP CPU usage and system load analysis:**

Wind River System Viewer contains powerful analysis capabilities to aid SMP kernel and application developers. Analysis suite views, derived from System Viewer data, show how tasks and processes migrate across cores during an instrumentation session.

**VxWorks 6.x Kernel Configurator:** This graphical utility simplifies and accelerates the task of selecting the operating system components to be included in a bootable VxWorks image. In addition to configuring VxWorks images for single-core processors, the Kernel Configurator allows the creation of SMP and AMP projects on any supported SMP or AMP target.

**High-speed global text search-and-replace:**

The Search view provides high-speed text search-and-replace features based on project settings. The advanced contextual search-and-replace system provides a far more intuitive and faster solution than traditional command-line tools.

**Wind River Linux Features**

**Git support:** Use Git to maintain project-specific kernel patches and export them to a layer for use in other projects, directly from the Workbench user interface. This project-specific support directly to the Git repository lets developers create customized layers from any number of kernel patches they need.

**Configuration management system**

**support:** ClearCase, CVS, and SVN (Subversion) support is available from the Workbench user interface. This support enables developers to retrieve and update projects without having to leave Workbench.

**RPM debug:** Workbench provides support for creating a target file system with debug symbols to facilitate package debugging. Use the fs-debug option in the Project Explorer to generate debug symbols in export/dist/usr/lib/debug. This feature provides for production-level debugging on the host and target.

**64-bit target debugging and analysis**

**tools support:** Workbench 3.3 provides both kernel and user space debugging in addition to profiling and analysis capabilities for Wind River Linux with a 64-bit kernel on x86-64 and MIPS64 architectures.

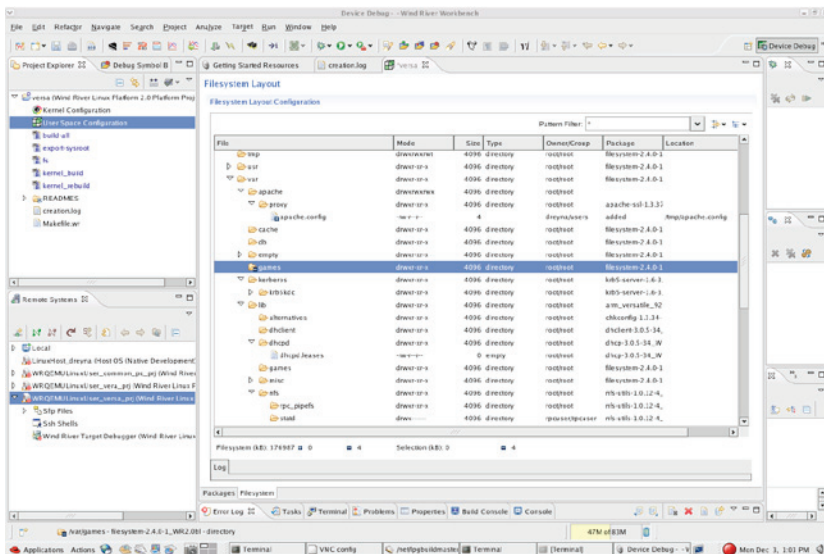


Figure 1: Linux File System Configurator

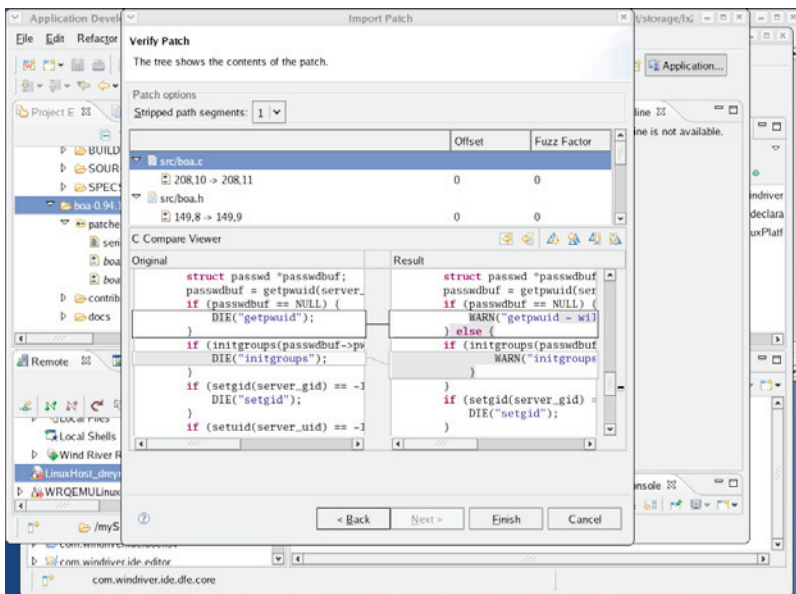


Figure 2: Enhanced Patch Manager

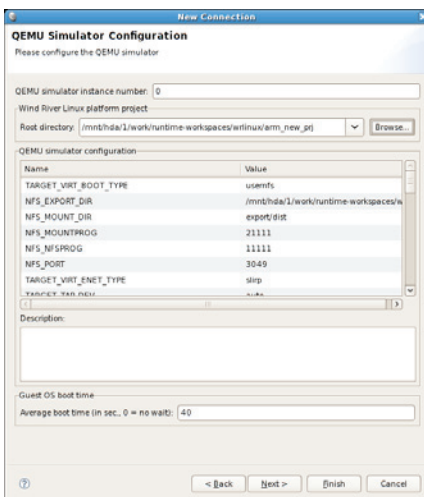


Figure 3: QEMU control from Workbench

command-line tool. It allows users to manage a series of patches by keeping track of the changes that each patch makes. Patches can be viewed, applied, unapplied, and refreshed. The Workbench Editor provides information about the patch a particular piece of code is derived from. The Patch Manager integrates the functionality of quilt into the Workbench user interface and greatly enhances ease of use.

### QEMU emulator control from the GUI:

Easily connect to virtual hardware models via the Workbench GUI. The simulator connection wizard now supports QEMU, a free Linux emulator, as well as third-party simulator tools with more advanced features.

## Application Components

### Eclipse

Because of its openness, capability, and strong community support, Eclipse was chosen as the framework for the Wind River Workbench development suite. Open, extensible, and backed by a strong community of commercial and open source developers, the Eclipse framework provides developers using Workbench with a wide range of additional integrated functionalities.

Eclipse integrated capabilities are provided by commercial development tool providers (such as IBM, Hewlett-Packard, and Borland) and an active developer community. As a result, developers have access to a wide range of value-added plug-ins from third-party and in-house sources that can be used to extend the capabilities of Wind River Workbench. Examples include Eclipse-integrated configuration management (CM) systems and editors, which offer simple plug-in integration with Wind River Workbench through standard Eclipse interfaces.

More information on Eclipse and available third-party plug-ins is available from the Community Projects and Plug-Ins section of the Eclipse website, <http://www.eclipse.org>.

In many cases, users will need to validate on their own the utility and compatibility of these plug-ins with Wind River Workbench.

**Linux File System Configurator:** This provides fine-granular control over what goes onto the target and the ability to tune the file system to contain exactly what is required and avoid any unnecessary files. The File System Configurator allows you to view, add, and remove files on the target file system just as easily as using a traditional file manager.

**Enhanced Patch Manager:** The Workbench Patch Manager makes handling patches easy and efficient. In particular, Linux kernel development involves applying patches. The patch management tool commonly used by the community is Quilt, a text-based

## C/C++ Development Tooling

Workbench 3.3 includes features from the Eclipse CDT project, many of which were developed and contributed by Wind River for the benefit of Wind River customers. The features provided by CDT 7.0 include the editor, the source code parser, and the indexer.

Workbench C development tools offer the most productive development environment for embedded programmers who are focused on writing applications for Wind River Linux and VxWorks. Comprehensive visual tools allow for simple writing, editing, and debugging code, either on simulated or real hardware targets.

C/C++ editing in Workbench is fully supported by the CDT. The editor provides syntax highlighting, content and code assistance, code folding, and code formatting, as well as integrated debugging features. Many of the advanced features of the code editor are customizable to individual tastes.

The source code analysis features include active vs. inactive code analysis, visualization of source and header file include hierarchies, call hierarchy browsing, and class hierarchy browsing.

## Wind River Workbench Components

- Eclipse
  - Eclipse platform 3.6
  - Eclipse CDT project 7.0
  - Target Management/Remote System Explorer 3.2
- Project System
- Build System
- Index-based global text search-and-replace
- Wind River compilers
  - Wind River Diab Compiler for VxWorks
  - Wind River GNU Compiler
  - Intel ICC/IPP
- Debugger
  - Target debugging agents for Wind River Linux
  - Target debugging agent for VxWorks
- Simulators
  - VxWorks Simulator
  - QEMU open source emulator

- Shell environments
  - Host shell
  - VxWorks kernel shell
- Configuration tools
  - VxWorks Kernel Configurator
  - Linux kernel and user space configuration tools
  - Linux File System Configurator
- Run-time visualization and analysis tools
  - System Viewer
  - Performance Profiler
  - Memory Analyzer
  - Data Monitor
  - Code Coverage Analyzer

## Add-on Solutions

- Wind River Workbench On-Chip Debugging 3.3
- Wind River ICE 2
- Wind River Probe
- IPL Cantata++ for Wind River Workbench
- Wind River Test Management

## Target Management/Remote System Explorer

Target Management provides information about remote and local systems. Using the Remote System Explorer, the user can discover, analyze, and interact

with a variety of service providers. Target Management provides Workbench users with a powerful and fast Terminal view for interacting with Wind River platforms; the ability to filter information based on custom settings; and file system access using FTP, SSH, and other protocols.

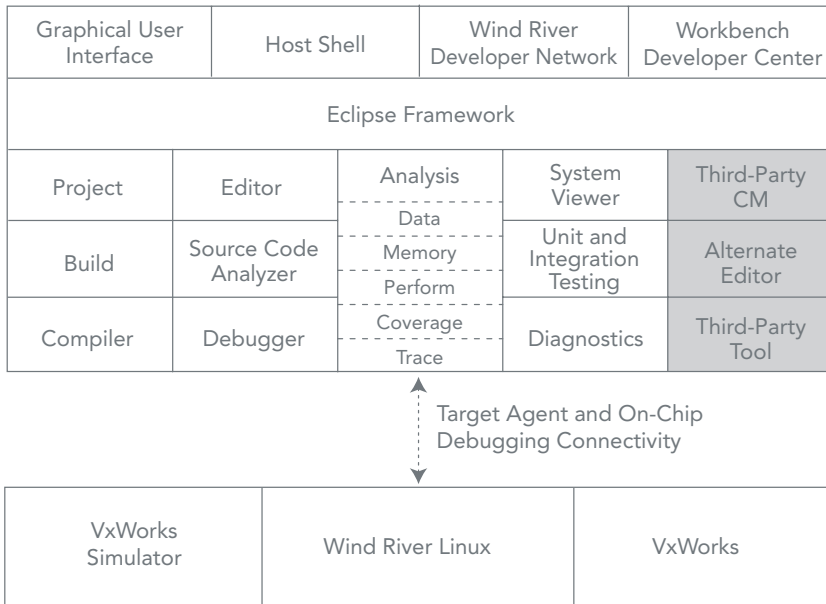


Figure 4: Wind River Workbench components

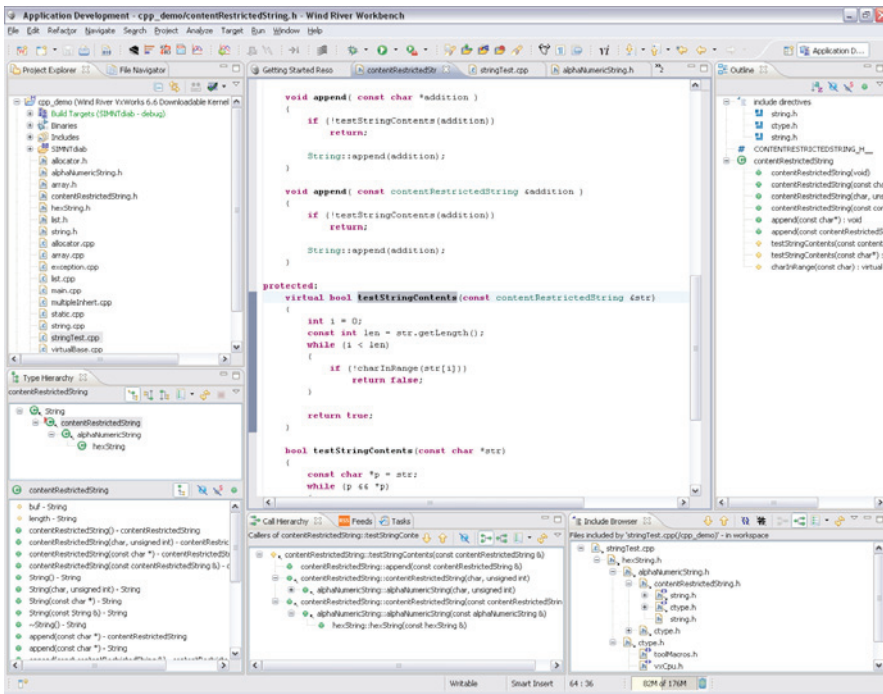


Figure 5: Wind River Workbench

- Work with multiple projects in a workspace simultaneously, reducing redundant setup and configuration and eliminating wasted time between projects.
- Automatic version control keeps track of changes, reduces errors, and allows restoration of previous project environments.
- Structured projects reflect the build hierarchy by linking subprojects to multiple parents, so you can easily see the relationship between the project and the way it is built.
- Provide and associate different build rules with different components of a project.
- Easily identify, visualize, and manage changes to the hierarchy of files, folders, and projects.
- Update workspace and project data via the command line.

### Build System

The Workbench Build System specifies the tools, options, and parameters to use when building device software projects, enabling you to set build parameters easily from the project level down to the individual file level. The Build System allows for use of simple global build-setting, fine-grained control at the level of an individual file, and everything in between.

Features of the Build System include the following:

- Specify how you want to generate makefiles; the managed application build process, based on make, is simplified through a series of wizards and self-guided dialogs.
- Automatically generate dependencies (compiler-dependent).
- Generate include search paths through source code analysis.
- Designate GUI configuration of build tools and build specifications at the project, build target, folder, or file level, with property sheets for advanced customization.
- Specify custom compiler target and optimization settings at the project, build target, folder, or file level with the compiler configuration GUI.
- Easily switch between build specifications.
- Support multiple targets per project, including specific build settings and flexible content definition (including exclusion patterns).
- Use the command-line build capability for scripted and nightly builds.
- Set default settings, which shortens the time to an initial build.
- Use VxWorks shared library build support.
- Get expert-level control and customizability with flexible content definition, file-level build, settings, and multiple build targets.

- Create new projects with wizards, such as shared libraries, VxWorks file systems, downloadable modules, bootable ROMs, and RTPs.
- Use existing makefiles for user-defined builds.
- Get support for remote builds.

### Index-Based Global Text Search-and-Replace

The Search view provides high-speed text search-and-replace features based on project settings. The searches can operate on text or regular expressions and can find matches within context (such as comments, literal text, source code, etc.). The resulting replacements can be previewed individually or accepted for all occurrences.

### Compilers

Wind River provides the following compilers for use in Wind River Workbench when developing with VxWorks 6.x: Wind River Diab Compiler, Wind River GNU Compiler, and the Intel C++ compiler.

#### Wind River Diab Compiler

Wind River Diab Compiler is the default C/C++ compiler configured for building the VxWorks 6.x kernel, libraries, BSPs, and applications in Wind River Workbench. It also supports standalone (no RTOS) development. This compiler's optimization capabilities are based on and extend the industry-hardened Diab Compiler technology, which has been in use in mission-critical applications for more than 20 years. Wind River Diab Compiler produces reliable, tight, and fast-executing code.

Features of Wind River Diab Compiler include the following:

- Superior optimization technology to generate fast, compact, high-quality code
- Proven performance with VxWorks
- Profile-guided and cross-module optimizations
- 100 percent compatibility with the latest ANSI C++ specs (ISO/IEC 14882:1998(E) C++ standard), the ANSI C spec (X3.159-1989), and the ANSI C (ISO/IEC 9899:1999) spec for RTP
- Standards conformance (ANSI and EABI) for maximum tool interoperability
- Complete control of code and data memory allocation

- Position-independent code (PIC) and position-independent data (PID) support
- Compile-time and link-time lint utility to find common programming mistakes

Wind River Diab Compiler also supports run-time error checking that detects and corrects hard-to-find problems, such as memory leaks and out-of-bounds pointers, to aid in producing higher-quality code.

Wind River Diab Compiler 5.8.0 includes new enhancements such as new linker options for library dependencies and RAM usage, new compiler options for ARM and x86 architectures, and new features for PowerPC architecture.

For more detailed information, see the Wind River Diab Compiler product note.

### Wind River GNU Compiler

Wind River GNU Compiler is based on the Free Software Foundation (FSF) distribution of GCC and g++. It provides support for VxWorks 6.x and Linux. Wind River has modified an off-the-net version of GCC specifically for use with VxWorks 6.x. The primary areas of modification deal with support for RTPs and shared libraries.

Features of Wind River GNU Compiler include the following:

- cpp, the C preprocessor
- GCC, the C and C++ compiler
- ld, the programmable static linker
- as, the portable assembler
- Binary utilities

### Intel C++ Compiler/Intel Integrated Performance Primitives

The Intel C++ Compiler processes C and C++ source language to generate code for IA-32 or Intel 64 architecture applications. The Intel Integrated Performance Primitives (Intel IPP) library includes general signal and image processing, computer vision, data compression, cryptography, string manipulation, audio processing, video coding, realistic rendering, and 3D data processing. It also includes more sophisticated primitives for construction of audio,

video and speech codecs such as MP3 (MPEG-1 Audio, Layer 3), MPEG-4, H.264, H.263, JPEG, JPEG2000, JPEG XR, GSM-AMR, and G.723.

By supporting a variety of data types and layouts for each routine, the Intel IPP library delivers a rich set of options for developers to choose from when designing and optimizing an application. To speed up performance, Intel IPP routines are optimized to use all benefits of Intel architecture processors. In addition, most of the Intel IPP routines do not use complicated data structures, which helps reduce overall execution overhead.

### Debugger

The Workbench debugger addresses the common and unique needs of developers involved with hardware bring-up, firmware/driver/BSP development, kernel development, and application development. It incorporates the feature set of best-in-class on-chip debugging environments, source-level debuggers, and target OS-aware development environments. Wind River combines the power of a direct hardware on-chip

debugging connection and a target agent connection, providing the ability to debug complex environments and complex device software applications. Multi-context debugging capabilities allow developers to debug code running in multiple contexts simultaneously. Multiple contexts means any of the following:

- Multiple cores
- Multiple tasks/processes/threads
- Multiple physical processors
- Multiple processor types
- Multiple boards
- Multiple target operating systems

With support for on-chip-debugging-based and target-agent-based debugging in the same environment, Workbench provides more access to a wider range of debugging options than any single debugging method. For example, you can use the debugger connection to on-chip debugging to bring up new hardware designs and develop board initialization code, code for ISRs, code for device drivers for the kernel, and code for boot loader applications. You can use the debugger target agent to task-mode debug a user application—

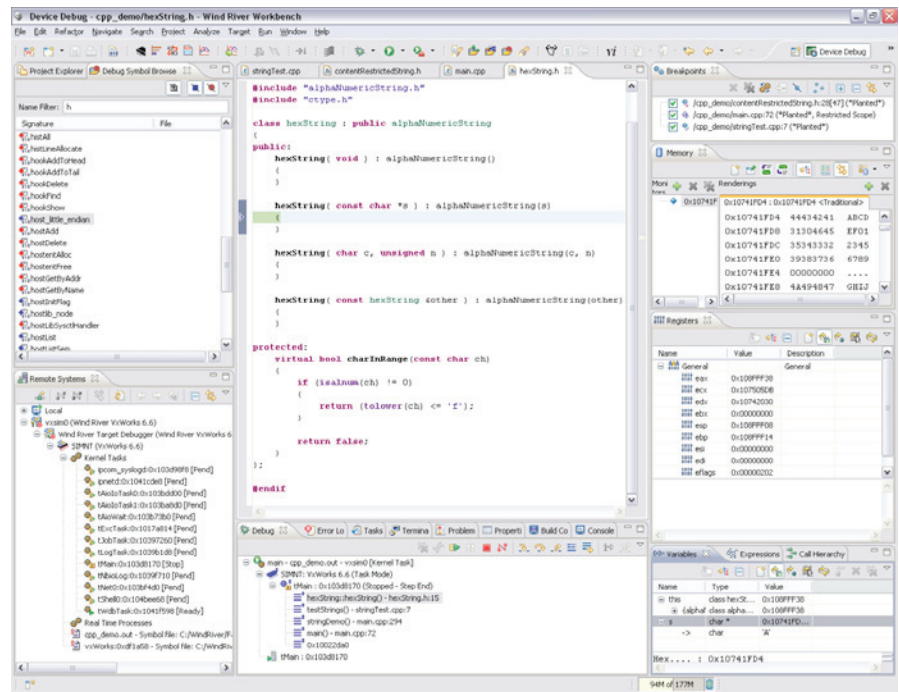


Figure 6: Workbench debugger

and also switch to the JTAG connection and determine the state of the target if it suddenly crashes and takes the target agent down with it.

Wind River also allows you to connect to a target running VxWorks 6.x using transparent mode driver (TMD) technology, in which a virtual connection is made to the target agent through the on-chip debugging hardware and connection. Workbench includes the ability to debug VxWorks 6.x targets via full kernel, task, and RTP debugging, including visibility and synchronous control of tasks associated to an RTP, as well as full kernel and user mode debugging of Linux targets.

Tracking down kernel and interrupt handler bugs is difficult and time-consuming. A solid kernel mode debugging solution can save days or weeks in comparison to printf/printk, which requires multiple edit-compile-debug cycles. Workbench offers both on-chip and agent-based solutions for Linux kernel debugging.

The dynamic printf() feature enables more productive debugging sessions. By allowing developers to dynamically insert printf() functions into compiled code without the need to recompile or download modified images to the target device, the Workbench debugger enables heightened productivity for all developers to rapidly instrument code and diagnose hard-to-find bugs.

Workbench developers using dynamic printf() can also benefit in applications where using traditional breakpoints to halt the processor is not possible or practical, such as in avionics or networking applications.

Workbench debug mode allows the use of Wind River's platform-integrated debugging capabilities in concert with editors and build systems that are external to the Workbench development environment. It includes a simplified user interface and optimized workflows for debugging. With Workbench debug mode, users can quickly connect to a target and start debugging without creating a workspace or project.

User applications (both VxWorks and Linux) often involve multiple interacting tasks/processes. Debugging these applications is greatly enhanced if more than one process or task can be debugged concurrently. Workbench allows this functionality, and it also offers the ability to debug a task or a thread independent of a process (Linux) or real-time process (VxWorks 6.x). The development suite also enables users to set breakpoints that can stop a single process/task or a specified group of processes/tasks.

Workbench enables up to eight instances of each debugger view to be present within the perspective at one time. This capability allows you to view relevant data for any processor, process, or thread in time, or simply placing the data from multiple debug targets into a single view.

Debugger administration features enable the following:

- Multiple window instantiation, with color coding and user-controlled update
- Color highlighting of changed value in any display
- Ability to save and restore the state of the debugging environment
- Option to disable window updates on breakpoints or when execution stops
- Context-aware breakpoints
- Data tip support when in an appropriate stopped mode

C++ debugging capabilities include the following:

- Support resolution of ambiguous namespaces
- Debugging of derived classes
- Breakpoint support for a single inline function
- Support for stepping inline functions
- Debugging of templates
- Breakpoint support
- Inline template functions
- Debugging of implicit and explicit instantiations
- Debugging interrupt handlers using Workbench debugging agent and KGDB for VxWorks and Wind River Linux

Workbench provides extensive browsing and inspection capabilities on objects in the target platform. This awareness of operating system objects provides an

enhanced debugging experience that allows greater insight and productivity when debugging. With the object-browsing capability of Workbench, the user can inspect the following:

- RTPs
- Objects
- Tasks
- Shared libraries
- Semaphores
- Watchdog timers
- Message queues
- Memory partitions
- Modules
- Symbols
- ISR objects
- Triggers
- File descriptors
- I/O devices

### *Target Debugging Agents for Linux*

Wind River Workbench and Wind River Linux platforms together provide extensive debugging capabilities for the kernel, kernel modules, and user mode applications. User mode applications are supported using an advanced agent based on the ptrace API. Advances in Linux 2.6 kernel multi-threading enable developers to debug individual threads reliably inside processes without stopping the entire process. Because ptrace is applicable only with the debugging of applications in user mode, another method is required for debugging the kernel.

Kernel debugging for Linux 2.6 kernels, including device drivers, kernel modules, and interrupt handlers, is achieved by using the open source KGDB-2 agent, or the kernel-debugging version of the GNU Project debugger (GDB). With standard debugging tools, this would be an issue, as connecting to KGDB is not typically performed using the same debugger as the one used for user-mode applications. Workbench's multiple-context technology allows simultaneous connection to KGDB-2 and the target agent through the same interface. Workbench provides the synchronization necessary to debug a multiple-agent connection in a synchronized manner, despite the fact that multiple target debugger agents are in use.

### **Target Debugging Agent for VxWorks**

VxWorks 6.x requires a robust debugging agent capable of debugging many tasks within one or more RTPs. The debugging agent must provide this functionality across memory boundaries protected by hardware memory management units (MMUs). The debugging agent provides the ability to debug and control multiple tasks in multiple RTPs. At a high level, RTPs can be selected and controlled with specific actions. For example, issuing a “run” or a “stop” command results in the overall control of each task state within the RTP. Tasks can also be individually selected within an RTP and debugged without stopping any other tasks. Breakpoints can be created that are only detected when code is executed within the context of a specific task or RTP, greatly simplifying the debugging of a complex device design.

Capabilities include the following:

- Basic execution control (step into, step over, step out, go, and stop)
- Advanced execution control (go all, stop all)
- Comprehensive debugging views
  - Expressions view (formerly Watch)
  - Breakpoint view
    - Software breakpoints
    - Hardware breakpoints
    - Number of software breakpoints limited only by host and target resources
    - Counted breakpoints
    - Context-sensitive breakpoints
  - Variables view
  - Registers view
  - Debug view
  - Memory view
  - Debug symbol browser view
- Target selector facility/target connection dialog/wizards
- Debugging code in shared libraries
- Kernel and user space debugging
- Debugging over MIPC with WDB proxy for VxWorks AMP systems
- Forked-process debugging
- Ability of the debugging agent to stop the entire system when it panics, allowing attachment of Workbench to track the cause of the panic (typically, memory access error in a kernel thread or interrupt handler)
- System and task mode debugging on SMP systems

- Process fatal exception debugging—the agent can attach to a process before it generates a core file, and the debugger can display all process and stack back trace information
- Kernel fatal exception debugging—the agent can remain in system mode after the exception occurs, allowing the debugger to display kernel stack trace and other useful information
- Debugging of “stripped” target applications in which symbolic information only exists in the host copy (in order to save target memory resources)
- Support for hardware and software breakpoints
- No impact on the behavior of the scheduler
- Proven stability and robustness through 20-plus years of use as the debugging agent for VxWorks in all major processor architectures

### **Simulators**

Workbench supports both the VxWorks Simulator for VxWorks users and the QEMU open source emulator for Wind River Linux users.

#### **VxWorks Simulator**

VxWorks Simulator 6.9 is a complete prototyping and simulation tool for VxWorks 6.x applications. VxWorks Simulator supports simulation of SMP systems. It also provides a simulation environment for VxWorks AMP development using MIPC. It enables you to develop and test significant portions of your application earlier in the development cycle, before hardware is available. It can also lower your development cost by allowing developers to share fewer hardware targets by enabling host-based development. The simulator is fully integrated into the Wind River Workbench development environment as a target connection, allowing complete configuration and debugging control through standard interfaces. Enhancements for this release include an increase to the number of parameters added to the vxsim command, access of the VxSim console through a telnet client, and display of the VxSim console in Workbench.

### **QEMU Open Source Emulator**

QEMU is an open source processor emulator supported by Workbench for Wind River Linux users. Users can easily connect to QEMU virtual hardware models via the Workbench GUI. The simulator connection wizard supports QEMU as well as third-party simulator tools with more advanced features. QEMU is provided for ARM, MIPS, Intel Architecture, and PowerPC single CPU targets.

### **Shell Environments**

To enhance developers’ work environments and improve their effectiveness in developing VxWorks 6.x–based device software applications, Wind River Workbench provides two command-line environments, or shells: the Host Shell, supported for both VxWorks and Wind River Linux targets, and VxWorks Kernel Shell.

#### **Host Shell**

The Host Shell provides a command-line interface that lets you download application modules and invoke platform and application module subroutines. This facility has many uses:

- Ability to work with VxWorks, Wind River Linux, and targets without an operating system Emacs integration for debugging and building VxWorks and Wind River Linux targets
- Interactive exploration of the operating system by calling any platform routines and APIs
- Interactive exploration of VxWorks 6.x RTPsDebug and monitor processes
- Prototyping
- Interactive development by calling any application (RTP) routines
- VxWorks 6.x application (RTP) and kernel testing
- Error management support through the output of error dumping; the ability to turn on/off error management on a per-task or per-RTP basis
- Wind River Architecture for Messaging Protocol (WRAMP) support through text dump of the message traffic

The Host Shell executes on the development host, not the target—but it enables you to spawn tasks, look at RTPs, read from or write to target devices, and exert

full control over the target. The Host Shell receives your commands, executes them locally on the host, and dispatches requests to the target server for any action involving the symbol table or target-resident programs or data.

In order to simplify and streamline working with remote code running on a target device, Workbench supports the use of Host and Kernel shells. Application developers can use the Host Shell functionality to rapidly prototype C code or quickly access symbol tables, all without having to launch or configure a debug session. Shell access provides developers with the ability to save time and effort in deploying and testing code on a device.

Because the shell executes on the host system, you can use it with minimal intrusion on target resources. As with other tools, only the target agent is required on the target system. Thus, the Host Shell can always remain available; you can use it to maintain a production system, as well as to experiment and test during development. Since you do not need to rebuild the VxWorks 6.x image, the Host Shell is useful on targets with restricted memory and permits system-mode debugging, which is critical to debug device drivers and interrupt handlers.

Capabilities of the Host Shell include the following:

- RTP debugging
- Kernel task debugging
- Integration with Emacs for GDB debugging and source build
- Shell access to remote VxWorks AMP core with tip
- Setting and managing breakpoints
- Task-specific breakpoints
- Task-specific single-stepping
- Symbolic disassembler
- Task/process and system information utilities
- Ability to call user routines
- Ability to create and examine variables symbolically
- Ability to examine and modify memory
- Exception trapping for debugged processes

- Ability to run on all supported host operating systems: Linux, Windows, and Solaris
- Use of the debugging agent to access the target through three connection types: network, serial, or a pipe connection
- Access to the target through the network for process debugging or the serial or network for kernel debugging
- Support of different interpreters depending on target OS (C, CMD, Tcl, GDB)
- Support of Tcl scripting and backward-compatibility with Tcl 8.0
- GDB command-line mode
- Command mode (CMD)

#### **VxWorks Kernel Shell**

The VxWorks 6.x Kernel Shell, formerly known as the Target Shell, runs within the VxWorks 6.x kernel and provides direct access to VxWorks 6.x through a console or a network connection, such as Telnet. While the VxWorks Kernel Shell and the Host Shell have similar capabilities, there are several key differences:

- The VxWorks Kernel Shell implements fewer commands than are provided in the Host Shell.
- Both shells include a C and CMD interpreter; the Host Shell also provides a Tcl and a GDB interpreter.
- The Kernel Shell, as well as its associated target-resident symbol tables and module loader, must be configured into the VxWorks 6.x image by including the appropriate components; the Host Shell is always ready to execute, provided the debug target agent is included in the system.

The Kernel Shell's input and output are directed at the same view by default, usually a console connected to the board's serial port; for the Host Shell, these standard I/O streams are not necessarily directed to the same view.

Features of the VxWorks 6.x Kernel Shell include the following:

- Updated monitoring commands to support VxWorks SMP and AMP
- Save and restore Kernel Shell command history in an external file

- Multiple session support to enable a user to spawn multiple kernel shells
  - Use for Telnet, rlogin access, and virtual consoles
  - Enable independent and simultaneous interaction with the target from several remote connections and from the console
  - Create a new shell session redirected to the host computer through the debug virtual I/O
  - Create new Kernel Shell sessions from the host tool wtxConsole using the "-s" option
  - Set up a VxWorks 5.5-compatible mode; one shell session can be created and it is shared between the connections
  - Global standard I/O of VxWorks is set to one of the shells
- Multiple interpreter support—integrate your own interpreters (Tcl, GDB-like interpreter), dynamically switch between them, and have one interpreter evaluate statements by another one
- Inclusion of the shell in the kernel without spawning an initial shell task
- Provided symbol name, task name, and path completion
- Configurable shell prompt with format strings like the current path and target name are provided
- RTP support
  - Launch and delete an RTP
  - List the RTPs and RTP tasks
  - Stop or continue an RTP (stop and continue all RTP tasks)
  - Add or remove a breakpoint on an RTP application
  - Disassemble, display, and modify RTP memory
  - View or modify the value of an existing symbol in an RTP
  - Shared library support
  - Execution path settings
  - Environment variable settings
- Command interpreter mode—a UNIX-like interpreter is available to access the RTP; also available in the kernel
- Fault management support—Kernel Shell provides commands to display information reported by the operating system error management capabilities, as well as help debug applications using the fault management capabilities

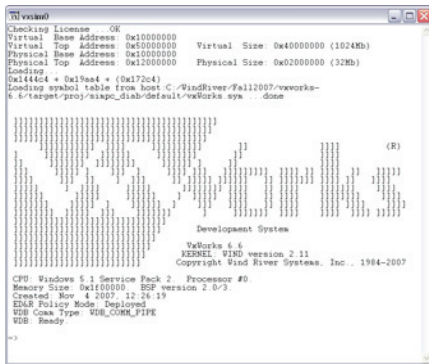


Figure 7: VxWorks Kernel Shell

- Protected access to the Kernel Shell with a login and password; available for remote connections and for a connection over the console/serial line; configure an inactivity time-out delay, after which the shell returns to the login/password prompt
- Multiple line edit (LED) mode support—integrate LED modes (an Emacs-like LED mode and traditional vi-like LED mode) and dynamically switch between them
- Kernel Shell access via a Workbench virtual console

## Configuration Tools

### VxWorks Kernel Configurator

The VxWorks 6.x Kernel Configurator is a graphical utility that simplifies and accelerates the task of selecting the operating system components that need to be included in a bootable VxWorks image. A command-line utility, vxprj, is available; it enables the ability to create a configuration and perform a kernel build (e.g., for testing purposes) within scripts used as part of nightly builds.

When creating a new bootable kernel image, Workbench analyzes available kernel components and BSP and compiler selections and provides BSP-specific capabilities plus a set of general configuration profiles to select from. The Kernel Configurator displays a summary of key configuration data, such as number of selected components or data and text size. A bundle selector allows users to quickly and easily include or exclude dedicated configuration bundles composed of multiple components from a kernel image. Sample configuration bundles include components needed for POSIX compliance, real-time process development, and error management.

Selecting kernel components individually gives you greater flexibility and control over your VxWorks image. The Kernel Configurator analyzes component dependencies and highlights conflicts when components are required but not selected or if components are not compatible with one another. The AutoScale feature analyzes the entire VxWorks image and removes unused kernel components that may unnecessarily increase the size of a bootable image.

It is also possible to include custom component definitions for specialized purposes or from third parties; the Kernel Configurator verifies whether component selections are valid and free from conflict.

Workbench offers the ability to assemble projects in hierarchies as a means of designing the structure of a target system. A bootable kernel project may include a file system project, as well as several DKM and RTP projects. Each DKM or RTP project may consist of one or several libraries and other source code modules. The advantage of creating a project structure over defining loosely coupled projects is that the Build System will create bootable kernel images in a single run, compared to multiple build stages that must each be started individually. It

is also a more logical and natural representation of system components.

Features of the VxWorks Kernel Configurator include the following:

- Component bundle selector
- Automatic scaling of the kernel size (AutoScale)
- Dependency control and conflict resolution
- Ability to create project hierarchies to reflect the structure of the final system, including bootable file system DKM and RTP projects
- Command-line availability of all GUI features
- VxWorks Image Builder

Workbench includes configuration tools for VxWorks and Linux kernels, allowing dramatic simplification of setting up a device for the first time. Kernel options and settings are laid out in a visual, logical format allowing a bird's-eye view of the fundamental system facilities available for use. Kernel developers and system integrators can easily select, deselect, modify, and configure the entire image in order to optimize it for a particular set of tasks or operational modes. Using the kernel configuration tools dramatically eases the complexity of bringing up bare metal for the first time under Wind River Linux or VxWorks.

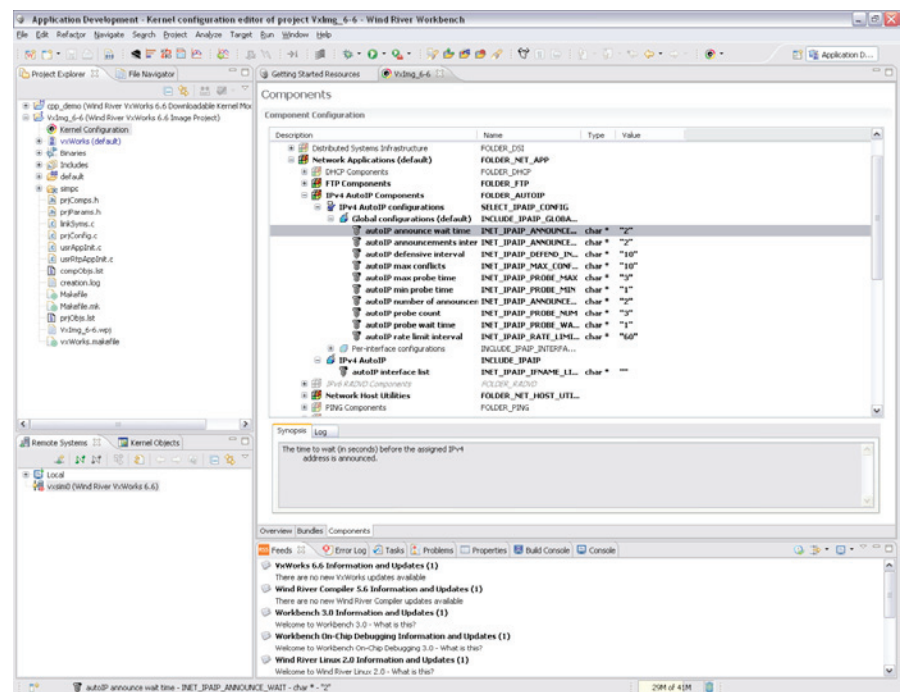


Figure 8: VxWorks Kernel Configurator

## Linux Kernel and User Space Configuration Tools

For Wind River Linux platform projects, the Kernel Configuration Tool facilitates Linux kernel configuration by providing an improved GUI, compared to the traditional xconfig and menuconfig configuration tools. The GUI is organized as two tabs for the Kernel Configuration view: One tab provides read-only summary information about the current configuration, including the configure command arguments used to create it; while the other tab provides the ability to include or exclude kernel items and also allows users to view and modify kconfig files, get help, and view dependencies. A search capability is provided, and it is possible to filter the view and to show disabled items.

The User Space Configuration Tool provides a GUI for RPM package configuration in the target root file system. The tool also offers an easy way to add or remove packages from the existing configuration, and it enables users to view package dependencies and get help information.

### Linux File System Configurator

The Linux File System Configurator gives fine-granular control over what goes onto the target and provides the ability to tune the file system to contain exactly what is required and to avoid any unnecessary files. The File System Configurator allows you to view, add, and remove files on the target file system (as it is placed in the distribution archive file that is expanded on the target), just as easily as using a traditional file manager. It can handle all types of files (including block and character devices and symlinks). It not only shows file attributes such as owners and permissions but also shows which RPM a file belongs to and how much space it will need on the target file system. Powerful filter and search features make file handling and system footprint tuning easy and efficient.

For kernel developers and system integrators the Workbench Kernel Configuration tools provide a graphical perspective within Eclipse to rapidly configure and deploy kernels for Wind

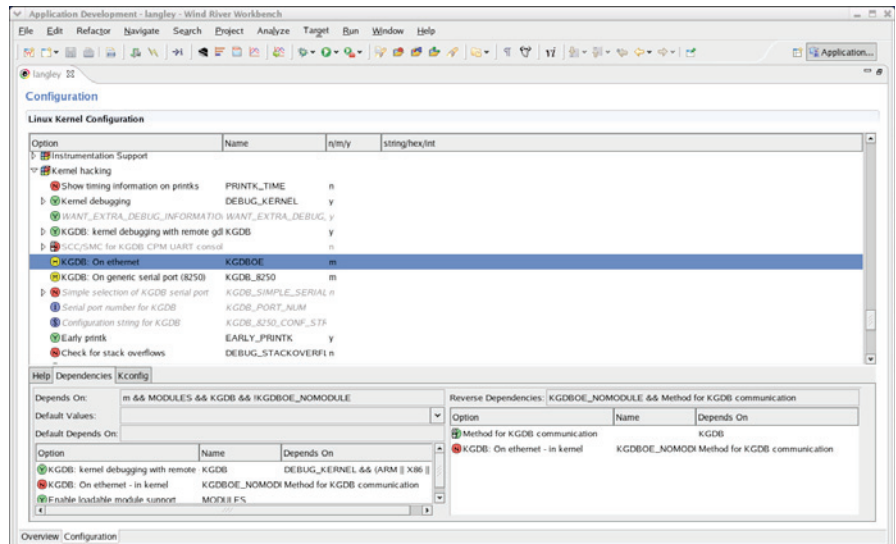


Figure 9: Linux Kernel Configuration Tool

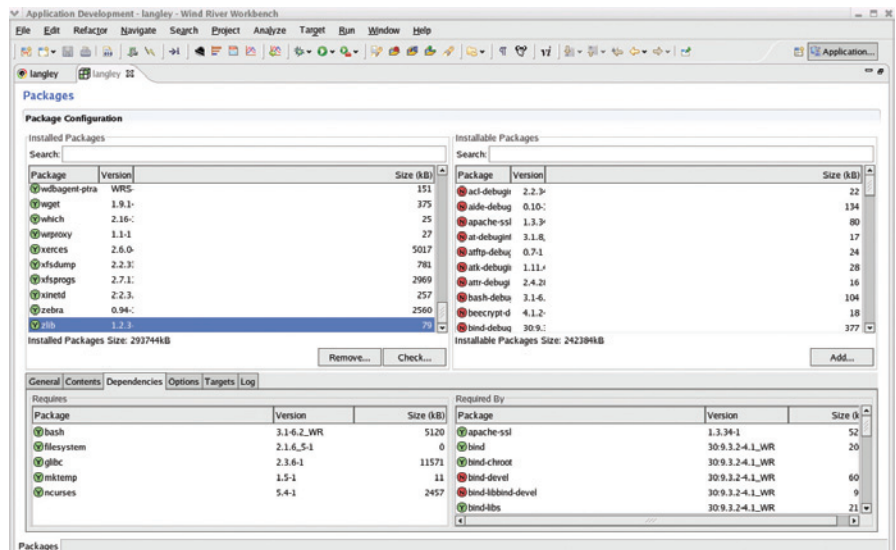


Figure 10: Linux User Space Configuration Tool

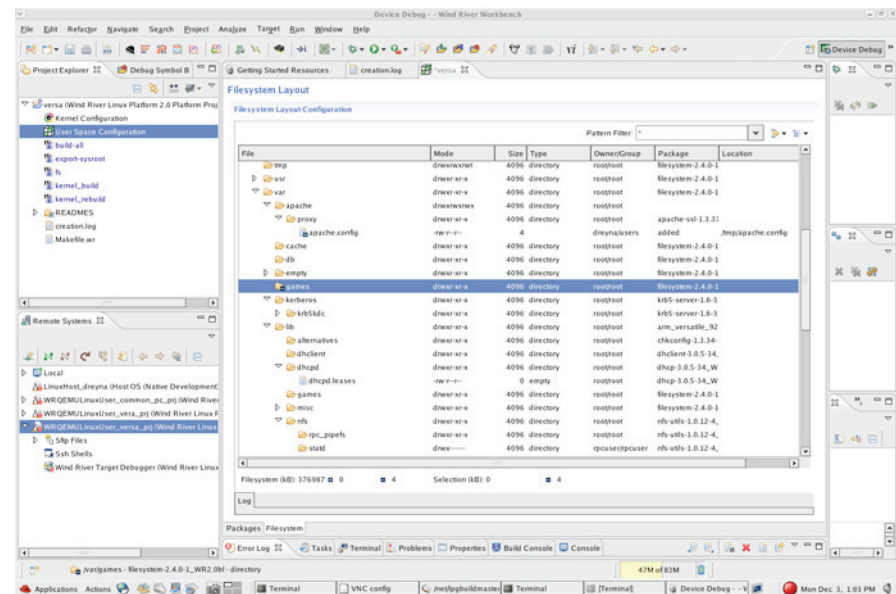


Figure 11: Linux File System Configurator

River Linux images. The kernel configuration tools dramatically reduce the traditional level of complexity associated with command-line approaches to patching, integrating, building, and deploying kernel images for embedded devices. Coupled with the comprehensive Workbench development environment, board bring-up and system optimization are easier to achieve than ever before.

### Run-Time Analysis Tools

Workbench contains six run-time analysis tools. These tools are powerful and dynamic visualization tools for device software applications. They provide developers with visibility into the entire platform: application code, third-party libraries, and the operating system. You can monitor variables, optimize

performance, and find memory problems—all while the system is still running.

An integral part of Workbench, the run-time analysis tools support both VxWorks 6.x and Wind River Linux platforms. Benefits include the following:

- Efficiently investigate defects and tune performance of complex systems, including multi-core processors.
- Learn where the processor spends its time, so you can focus on optimizing areas that contribute the most to overall performance.
- Analyze all the code running in the system, including the kernel and third-party applications.
- Understand more about the behavior of your platform through clearly diagrammed details of application and system behavior.

- Quickly uncover and resolve system problems faster through superior data analysis and display.
- Change variables on the fly, allowing more flexible experimentation and immediate observation of the results.
- Easily install tools with minimal system impact because there is no hardware setup or code instrumentation required (except for Code Coverage Analyzer).

### System Viewer

System Viewer captures the dynamic interactions of the operating system, device software applications, and target hardware. It is used to inspect the dynamic behavior of device software systems to detect run-time problems and improve system performance.

System Viewer provides detailed analysis and graphical visualization of system events, revealing the complex interactions of tasks, threads, interrupts, and system objects of applications executing on a target. Context changes are clearly shown, as well as such system events as semaphores, message queues, signals, tasks, timers, and user events. This tool is best suited for use when developers need to diagnose and solve one or more of the following problems:

- Task/process/thread scheduling problems, such as deadlocks and starvation and race conditions, including those conditions that are more pronounced under SMP
- Performance problems, such as priority setting, resource contention, and mutual exclusion
- Timing problems arising from the interaction of interrupts and tasks

### Performance Profiler

Profiling is critical for real-time systems. Once you understand performance bottlenecks, it becomes easier to optimize application code. Wind River Performance Profiler is a dynamic execution profiler that provides detailed function-by-function performance analysis, specifying individual routines within the program that are consuming the CPU cycles. Performance Profiler pinpoints inefficiencies and shows how performance changes over time. For developers targeting SMP systems, the Workbench Performance Profiler is a valuable tool.

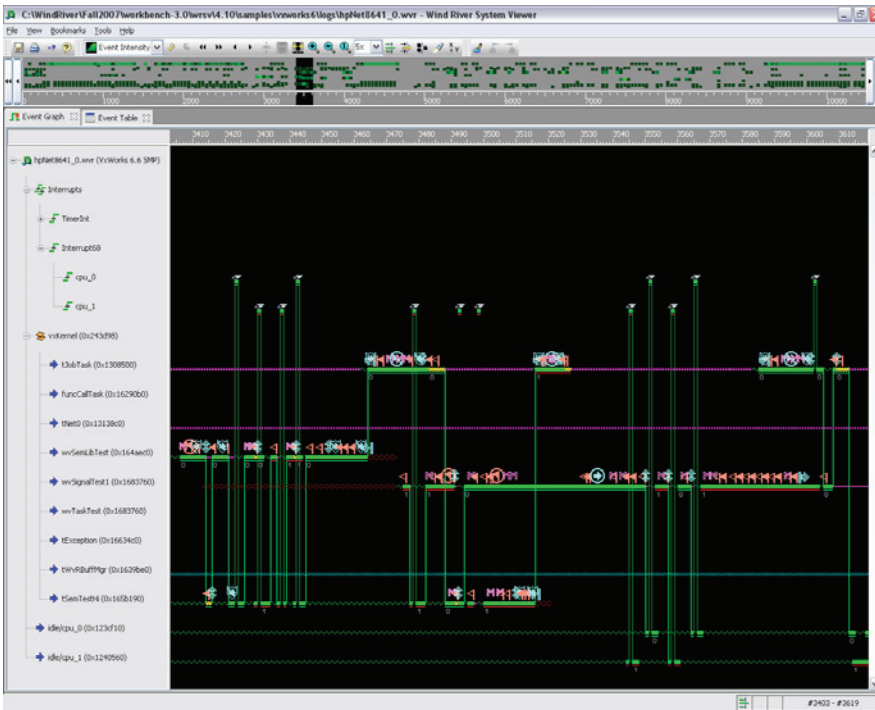


Figure 12: SMP support in System Viewer

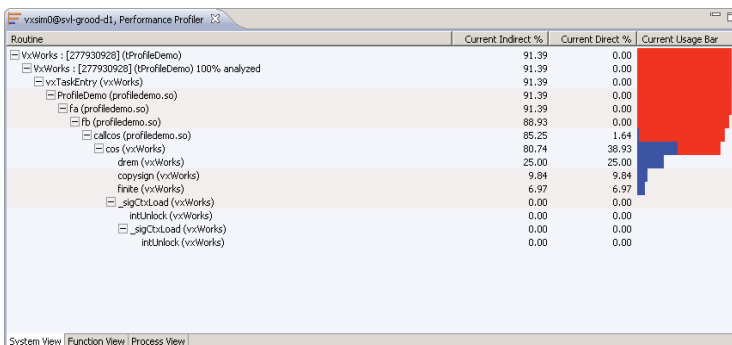


Figure 13: Performance Profiler

Performance Profiler now allows users to select arbitrary events. This feature enables users to profile events or registers exposed by the target system, such as cache misses. The tool can collect and display any register supported by the board support package.

### Memory Analyzer

Ensuring optimal use of memory is a critical activity in device software design. In many applications, memory usage is not fully understood and a large portion of available memory is wasted. Systems can run for days before failing due to noncharacterized memory leaks. Wind River Memory Analyzer provides greater visibility into memory usage. Without any special compilation or instrumentation, you can monitor available memory, detect leaks that occur due to system calls or third-party libraries, and even watch leaks as they happen.

Fragmentation view is enabled for mpatrol targets. Fragmentation view provides a graphical map representation of both VxWorks and Wind River Linux target memory allocation. Time view is also enabled for mpatrol targets.

### Data Monitor

This real-time graphical monitoring tool is used to examine variables, data structures, or memory locations in your system. You can watch any set of variables, see peak values and out-of-range settings you would otherwise miss, trigger collection on specific events, change variables while your program runs, and save collected data to disk. Wind River Data Monitor presents this live analysis of your program without stopping or slowing your code.

### Code Coverage Analyzer

Analyze your code to determine which code segments are executed during testing. Visibility into the execution of individual statements, decisions, and conditions enables you to create more thorough test scenarios, ensuring delivery of higher-quality devices. It also becomes easy to identify and remove code that is never executed, thus preventing future problems and reducing your overall memory footprint.

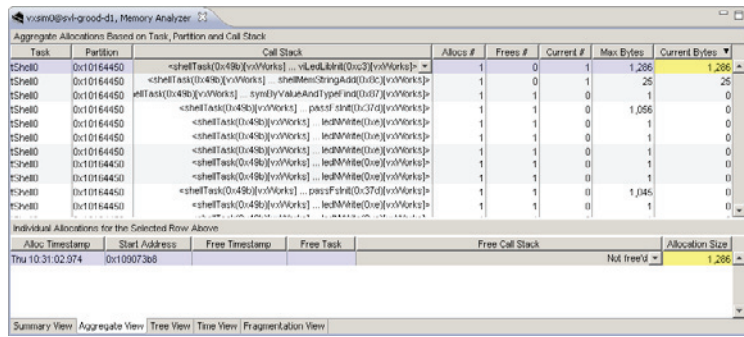


Figure 14: Memory Analyzer

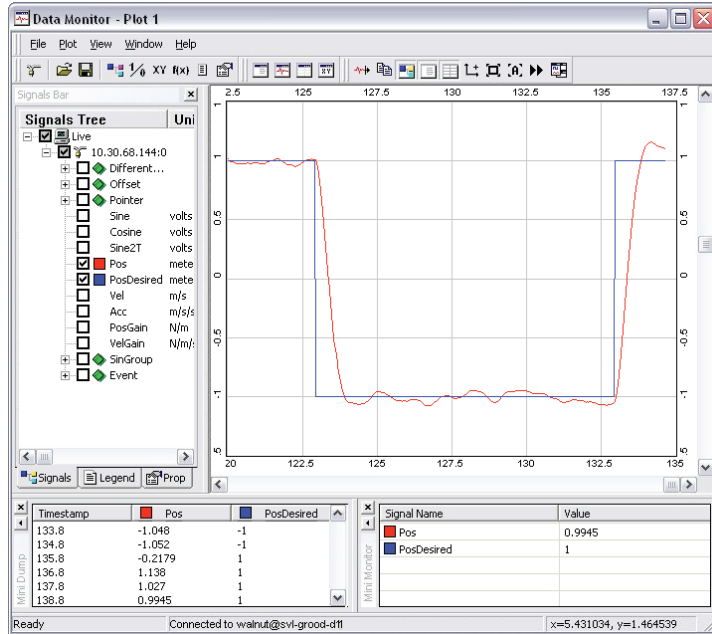


Figure 15: Data Monitor

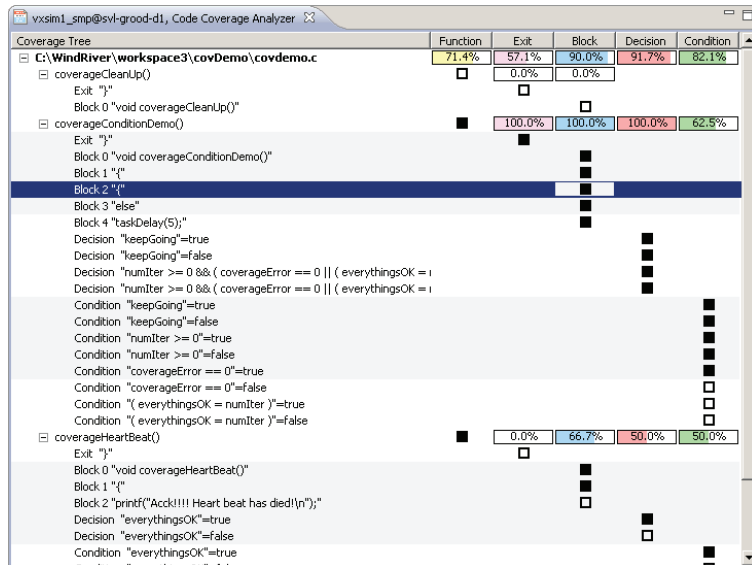


Figure 16: Code Coverage Analyzer

## Optional Add-Ons

### *Wind River Workbench On-Chip Debugging*

The Workbench development environment can be extended for on-chip debugging. Workbench On-Chip Debugging, along with Wind River ICE 2, or Wind River Probe hardware, provides access to significant additional capability within Workbench.

In the early stages of hardware and software development, a robust connection to the microprocessor through its run-control port is essential. Workbench On-Chip Debugging provides connectivity between the host development environment and the target device via the JTAG or on-chip debugging interface of the microprocessor residing on the device. The on-chip debugging interface of most microprocessors enables full control of the microprocessor itself, access to core and peripheral registers, and access to on-chip switch fabrics and memory controllers, along with access to external buses and many devices attached directly to the bus. In addition, some microprocessors support either internal or external trace buffers, allowing developers to capture information about the exact code that runs on the target and when.

Workbench On-chip Debugging provides developers with complete system-level control of their environments, from hardware bring-up to kernel, BSP, and device driver stabilization and system integration. It can also be a useful alternative to agent-based debugging in applications where serial, Ethernet, or USB interfaces are not available, or in environments where agent instrumentation of the OS is not desired.

Extended capabilities offered through Workbench On-Chip Debugging include the following:

- Complete run-control solution, including start, stop, and stepping capabilities, data- and expression-based hardware and software breakpoints
- CPU and peripheral register analysis and access to bit-level detail
- Advanced symbolic debugging with conditional breakpoint support

- Simultaneous independent connection to multiple CPU cores with per-core visual indicators and cross-correlation of call stack, register view, breakpoints, and source/assembly view
- Comprehensive diagnostics including RAM tests, address and data bus tests, CRC checks, and scope loops to determine bus routing issues on densely populated boards
- Flash programming for a wide variety of flash devices
- On-chip debugging command shell interface
- JTAG editor
- Statistical performance analyzer
- Target OS awareness for VxWorks and Wind River Linux that provides visibility to kernel objects, threads, tasks, and user-mode application contexts to allow debug of complex application-kernel interactions
- Statistical performance analyzer
- Wind River Hypervisor object awareness for bring-up of virtual boards and guest operating systems
- SMP-aware debugging that simplifies multi-core configuration and minimizes system skid during stop operations
- Program flow execution trace for supported processors that reduces debug time by providing the underlying cause of specific defects

For more information, see the Wind River Workbench On-Chip Debugging product note.

### *Wind River ICE 2*

Wind River ICE 2 is a network-based hardware unit providing connection to the processor's BDM/EJTAG/JTAG hardware run-control interface. It communicates with Workbench, running on the host machine via Ethernet, allowing on-chip debugging in a remote development environment. For more information, see the Wind River Probe product note.

Through its JTAG server capability, Wind River ICE 2 supports multiple JTAG/EJTAG devices on a single scan chain as well as multiple debugger connections to these devices. Wind River ICE 2 supports connections for up to eight devices simultaneously in a scan chain of up to 128 individual EJTAG/JTAG devices.

### *Wind River Probe*

Wind River Probe uses on-chip debugging services embedded in a microprocessor and the Wind River JTAG Accelerator technology to provide the industry's most feature-rich tool for board bring-up, flash programming, and production/test. Probe provides a high-speed USB connection between a host debugger and a target microprocessor for PC-based development as well as a high-performance emulation system with quick download speeds, overall debug throughput, and unprecedented debug efficiency. For more information, see the Wind River Probe product note.

### *IPL Cantata++ For Wind River Workbench*

IPL Cantata++ for Wind River Workbench, available at an added cost for use with VxWorks 5.5 and VxWorks 6–based platforms, is an integrated set of tools that allows developers greater efficiency in completing unit testing, integration testing, and code coverage analysis on the tests. The integration between IPL Cantata++ and the rest of the development suite places these capabilities within easy reach of every developer. Using this product increases software quality, decreases time-to-market, and reduces support costs through better, faster, more automated testing in the development life cycle. For more information, see the IPL Cantata++ for Wind River Workbench product note.

### *Wind River Test Management*

The Wind River Test Management system provides an integrated suite of applications built on a collaboration platform and test information repository that can be shared by quality assurance personnel and development engineers.

- Capture and manage test plans.
- Develop reusable manual and scripted test cases.
- Manage virtual labs of test devices remotely.
- Automate test execution across multiple devices.
- Monitor code coverage at run-time.
- Profile device performance at run-time.
- Incorporate white-box testing techniques.

- Collaborate to diagnose run-time defects.
- Patch software “hitlessly” without stopping devices.
- Manage with dashboards and reports.

Wind River Test Management is designed to manage multiple devices under test at multiple lab locations, maximizing resource utilization and accelerating the testing process. Wind River Test Management helps teams adopt a repeatable process for rigorously testing embedded software. The system streamlines workflow, automates testing, and mitigates risks and delays so products can get to market faster at lower cost.

## Technical Specifications

### Host OS Support

- Fedora 13, 32-bit
- Fedora 13, 64-bit
- Novell SUSE Linux openSUSE 11, 32-bit (SP2)
- Novell SUSE Linux openSUSE 11, 64-bit (SP2)
- Red Hat Enterprise Linux Workstation 4, 32-bit (Update 8)
- Red Hat Enterprise Linux Workstation 5, 32-bit (Update 5)
- Red Hat Enterprise Linux Workstation 5, 64-bit (Update 5)
- Red Hat Enterprise Linux Workstation 6, 32-bit
- Red Hat Enterprise Linux Workstation 6, 64-bit
- Sun Solaris 10 SPARC
- Novell SUSE Linux Enterprise Desktop 11, 32-bit
- Novell SUSE Linux Enterprise Desktop 11, 64-bit
- Ubuntu Desktop Edition 10.04, 32-bit
- Ubuntu Desktop Edition 10.04, 64-bit
- Windows XP Professional, 32-bit (SP3)
- Windows 7, 32-bit
- Windows 7, 64-bit

### Target OS Support

- VxWorks 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 6.9 platforms
- VxWorks 5.5
- VxWorks 653
- Wind River Linux
- Native Linux development on supported Linux hosts
- Customizable target OS awareness capability for Workbench On-Chip Debugging; enables support for other target operating systems to be added

## Target Processor Support

Contact Wind River Sales for list of supported processor families within each architecture.

## Target Architecture Support

### VxWorks 6.x

- ARM/XScale
- IA-32 and Intel 64
- MIPS
- PowerPC
- Renesas SuperH

### VxWorks 5.x

- IA-32
- PowerPC

### VxWorks 653

- PowerPC

### Linux

- ARM
- Intel
- XScale
- MIPS
- PowerPC

## Workbench On-Chip Debugging Target Architecture Support

- ARM
- ColdFire
- Intel
- MIPS
- PowerPC
- XScale

For additional support of target hardware and operating systems not listed here, contact Wind River Professional Services.

## Professional Services

Wind River Professional Services, a CMMI Level 3–certified organization, enables you to reduce risk and focus on development activities that add value and differentiate your design. As part of our solution, Wind River offers industry-specific services practices, with focused offerings that help you meet strict market deadlines while keeping development costs down. Our experienced team delivers device software expertise that solves key development challenges and directly contributes to our clients’ success. Based on our commercial-grade project methodology, Wind River Professional Services include

requirements discovery and definition, BSP and driver optimization, software system and middleware integration, and legacy application and infrastructure migration.

## Workbench Acceleration Services

Wind River Workbench Acceleration Services offer a comprehensive approach to helping your team realize the productivity benefits of Workbench through process integration, system integration, and education services. This approach begins with an assessment of your goals, which we then apply to a service plan specific to your needs. Services may consist of engineering, consulting, education, and mentoring, all with the end goal of helping your organization blend Workbench into your day-to-day development processes.

## Workbench Adoption Services

- Integrate customized version-control system
- Customize and automate build execution
- Customize and integrate projects
- Migrate existing and customized development environment
- Optimize Workbench performance
- Mentor one-to-one or one-to-few

## Technical Services

- Extend Workbench processor support
- Extend Workbench target OS support
- Validate Workbench on Linux host environment
- Validate Eclipse plug-ins
- Integrate agents

## Education Services

- Deliver standard education courses in public facilities or at your worksite
- Tailor custom courses to your team’s aggregate needs and deliver them at your site
- Tailor individual education plans to each developer’s needs as part of our Personalized Learning Program and deliver them at your site, with results measured across each developer’s education plan

## Installation and Orientation Service

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

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

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

## Support Services

Wind River Customer Support, a Service Capability and Performance (SCP)-certified organization, provides full technical support for VxWorks, Wind River Workbench, and Wind River Linux platforms. Our global support organization is staffed with engineers who have extensive experience with Wind River products and device software development.

Visit Wind River Online Support (OLS) at [www.windriver.com/support](http://www.windriver.com/support) for fast access to product manuals, downloadable software, and other problem-solving resources for Workbench. OLS offers a comprehensive knowledge base with a robust search feature for locating product information and manuals by keyword, author, published date, document type, language, and solution category. OLS visitors can also access a community of developers to discuss their issues and experiences.

If you cannot find the information you need through Online Support, contact our global support team for access to the industry's most knowledgeable and experienced support staff.

## North America, South America, and Asia/Pacific

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

## Japan

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

## Europe, Middle East, and Africa

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

**WIND RIVER**

Wind River is a leader in embedded and mobile software. We enable companies to develop, run, and manage device software faster, better, at lower cost, and more reliably. [www.windriver.com](http://www.windriver.com)

© 2011 Wind River Systems, Inc. The Wind River logo is a trademark of Wind River Systems, Inc., and Wind River and VxWorks are registered trademarks of Wind River Systems, Inc. Other marks used herein are the property of their respective owners. For more information, see [www.windriver.com/company/terms/trademark.html](http://www.windriver.com/company/terms/trademark.html). Rev. 06/2011