

# Wind River Workbench 3.0

Wind River Workbench 3.0 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 ThreadX
  - Target processor support for ARM, ColdFire, IA/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
- An extensible framework, based on Eclipse, 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 development suite is backed by Wind River's 20-plus years of device software industry experience, a

world-class support organization, and a specialized professional services team.

## New in Wind River Workbench 3.0

Workbench 3.0 includes significant new capabilities in support of increased development team productivity. (For a list of new capabilities in add-on products, see the respective product notes for those products.)

### General Workbench Enhancements

**Increased APIs and compatibility with Eclipse plug-ins:** The Eclipse C/C++ Development Tooling (CDT) and the Device Software Development Platform (DSDP) projects enable Workbench to adopt industry-wide standards. The benefit is that it makes Workbench more interoperable with third-party and homegrown Eclipse-based tools that leverage these standards.

**Migration to CDT Editor, Target Management, and Device Debugging views from Eclipse:** Workbench 3.0 is built on top of four Eclipse open source projects (Eclipse 3.3.1, CDT 4.0, DSDP-DD 0.9, and DSDP-TM 2.0), which makes it easier for third parties to add new capabilities by using industry-standard APIs. This strengthens the compatibility of Workbench with non-Wind River plug-ins, delivers new capabilities and fixes from the community and partners, and allows Wind River to focus on high-value commercial features on top of industry-standard Eclipse.

## Table of Contents

New in Wind River Workbench 3.0.....	1
General Workbench Enhancements .....	1
VxWorks Platform Enhancements ...	2
Wind River Linux Platform Enhancements .....	3
Included in Wind River Workbench .....	4
Application Components.....	4
Optional Ad-Ons .....	4
Eclipse .....	4
Project System.....	5
Build System.....	5
Index-Based Global Text Search-and-Replace.....	6
Wind River Compilers.....	6
Debugger .....	7
Simulators.....	8
Shell Environments .....	8
VxWorks Kernel Configurator .....	10

Linux Kernel and User Space Configuration Tools .....	10
Linux File System Configurator .....	11
Run-Time Analysis Tools .....	11
Optional Add-Ons .....	14
Wind River Workbench, On-Chip Debugging Edition .....	14
IPL Cantata++ for Wind River Workbench .....	14
Wind River Device Management .....	14
Technical Specifications .....	15
Professional Services .....	15
Workbench Acceleration Services....	15
Installation and Orientation Service .....	16
Education Services .....	16
Personalized Learning Program ....	16
Public Courses.....	16
Onsite Education.....	16
Support Services .....	17

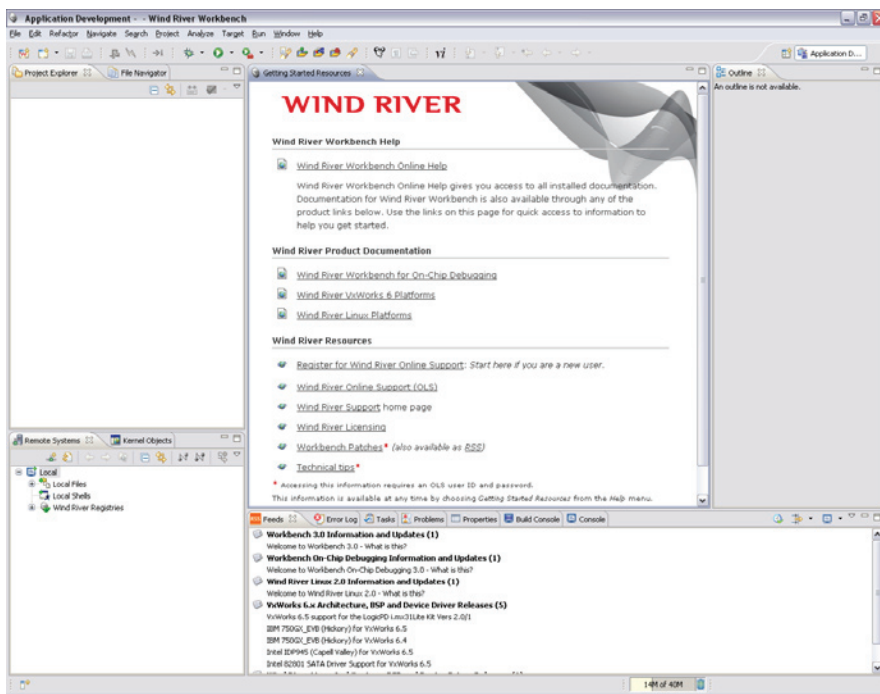


Figure 1: Getting Started Resources

**Performance and scalability enhancements for large application support:** Wind River Workbench is designed to perform in a wide range of project sizes, from individual developers to large, complex applications. Workbench is validated using performance and scalability benchmarks based on real customer applications and usage. For Workbench 3.0, performance and scalability has been validated and enhanced to meet the needs of Wind River customers.

**Workbench Getting Started Resources:** The Getting Started Resources is shown when the user starts Workbench for the first time and can remain open as a reference while the user works. The Getting Started Resources provides links to important web content and online documentation. Also included is an integrated RSS reader with preconfigured links to Wind River RSS feeds. The RSS feeds will announce the availability of updates for Workbench, tools, board support packages (BSPs), device drivers, and platforms.

**New host OS support:** Workbench supports several new host operating systems, including Windows Vista, Fedora Core, openSUSE, SUSE Linux Enterprise Desktop, and Solaris 10. See detailed technical specifications later in this document.

**Based on Eclipse 3.3.1 framework:** Workbench 3.0 is based on the most current Eclipse 3.3.1 Europa release, which was released in June 2007.

#### VxWorks Platform Enhancements

**Symmetric multiprocessing (SMP) support:** Workbench 3.0 has the ability to debug the VxWorks SMP kernel and real-time processes (RTP) in system or task mode. It has the ability to diagnose

race conditions and deadlocks. The VxWorks simulator can run SMP configurations, and the user can collect and visualize SMP data with Wind River System Viewer.

**SMP CPU usage and system load analysis:** Wind River System Viewer contains new analysis capabilities to aid SMP kernel and application developers. New analysis suite views, derived from System Viewer data, show how tasks and processes migrate across cores during an instrumentation session.

**Significant improvements in instrumented kernel size with System Viewer:** SMP is now supported throughout Workbench, with specific support in debugging, analysis, and simulation of SMP targets. It reduces the size of the System Viewer VxWorks kernel by 20K by optimizing the manner in which instrumented code is generated. As a result, user code executes faster and the execution overhead of the instrumentation primitives is decreased.

**Code Coverage Analyzer and Function Tracer:** These features, previously called CoverageScope and TraceScope, were available only through purchase of an add-on product called ScopeTools for Test and Validation. These products are now available to all Workbench users for no additional charge.

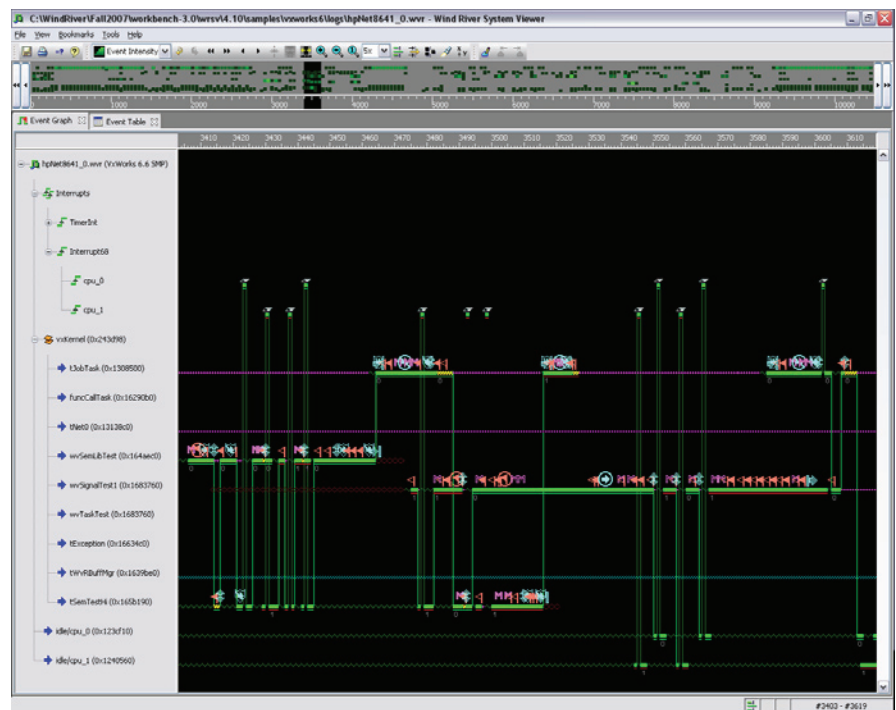


Figure 2: SMP support in System Viewer

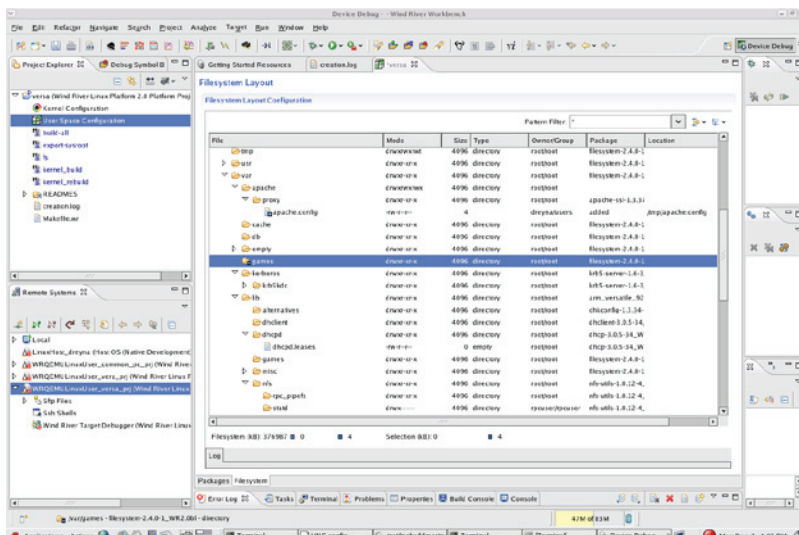


Figure 3: Linux File System Configurator

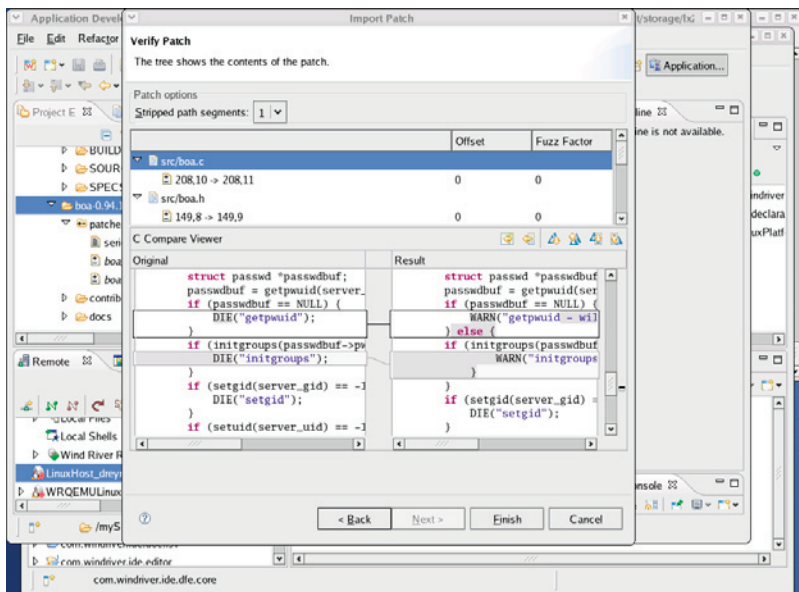


Figure 4: Enhanced Patch Manager

## Wind River Linux Platform Enhancements

**64-bit target debugging and analysis tools support:** Workbench 3.0 provides both kernel and user space debugging for Wind River Linux with a 64-bit kernel on PPC64, x86-64, and MIPS64 architectures.

**Linux File System Configurator:** The new 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 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, and the patch management tool commonly used by the community is quilt, a text-based

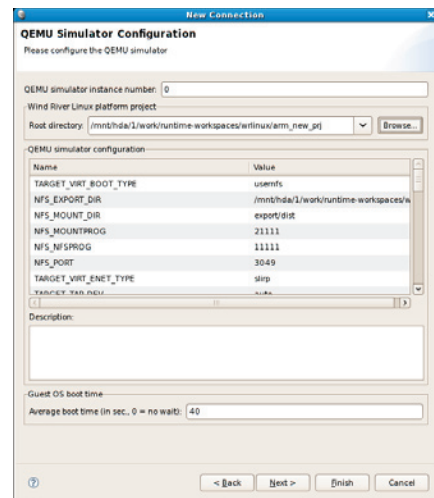


Figure 5: QEMU control from Workbench

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, and 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:

Users can now 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.

**GCC 4.x compiler support:** Workbench 3.0 supports the latest version of the GNU C compiler.

## Wind River Workbench

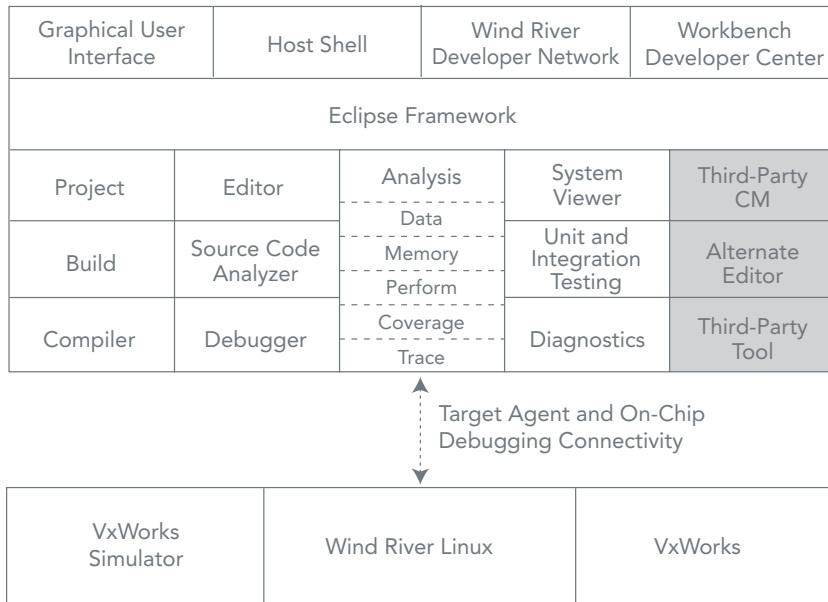


Figure 6: Wind River Workbench components

## Included in Wind River Workbench

### Application Components

- Eclipse
  - Eclipse platform 3.3.1
  - C/C++ Development Tooling (Eclipse CDT project) 4.0.1
  - Target Management/Remote System Explorer (Eclipse DSDP-TM project) 2.0
  - Device Debugging (Eclipse DSDP-DD project) 0.9
- Project System
- Build System
- Index-based global text search-and-replace
- Wind River compilers
  - Wind River Compiler for VxWorks
  - Wind River GNU Compiler
- Debugger
  - Target debugging agents for Wind River Linux
  - Target debugging agent for VxWorks
- Shell environments
- Simulators
  - VxWorks Simulator
  - QEMU open source emulator
- 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 (formerly ProfileScope)
  - Memory Analyzer (formerly MemScope)
  - Data Monitor (formerly StethoScope)

- Code Coverage Analyzer (formerly CoverageScope)
- Function Tracer (formerly TraceScope) (VxWorks only)

### Optional Add-Ons

- Wind River Workbench, On-Chip Debugging Edition 3.0
  - Wind River ICE
  - Wind River Trace
  - Wind River Probe
- Wind River Device Management

- Wind River Lab Diagnostics 2.2
- Wind River Field Diagnostics 2.2
- IPL Cantata++ 5.2 for Wind River Workbench 3.0 (formerly Wind River Workbench Unit Tester)

## 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.

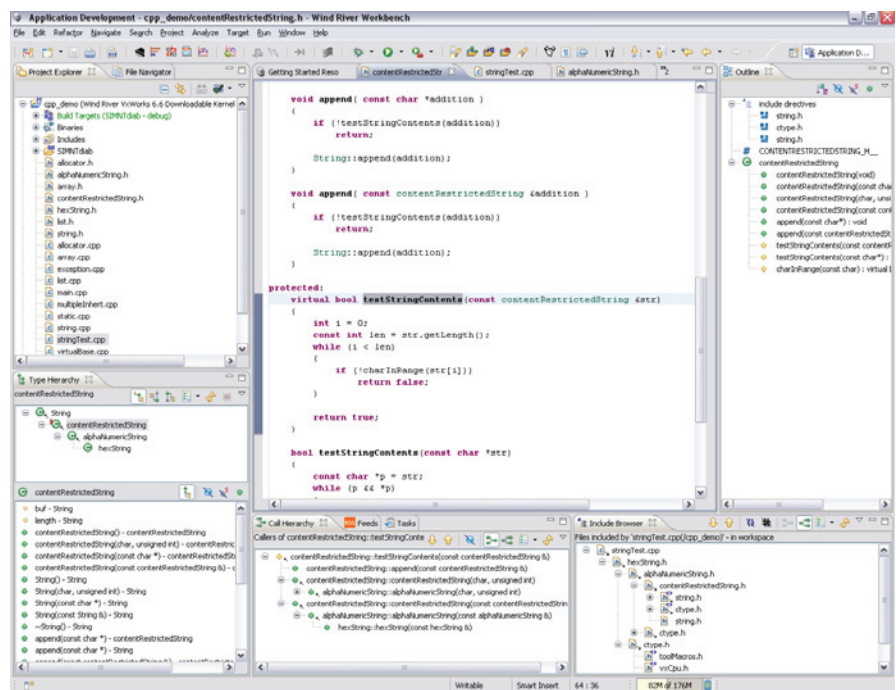


Figure 7: Wind River Workbench



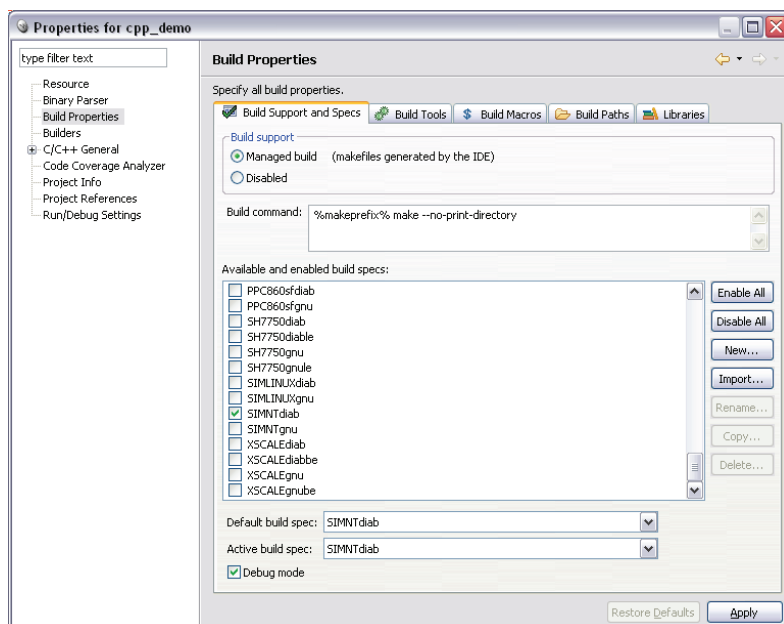


Figure 8: Workbench Build System

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.

### C/C++ Development Tooling (Eclipse CDT Project)

Workbench 3.0 includes useful new features from the Eclipse CDT project, many of which were developed and contributed by Wind River for the benefit of Wind River customers. In earlier versions of Workbench, Wind River provided these features independently, but with the migration of these capabilities to CDT 4.0, this functionality is now available in the open source project. Notable among the features provided by CDT 4.0 are the editor, the source code parser, and the indexer.

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 that were once available only to Workbench users are now available as part of the CDT. These capabilities include active vs. inactive code analysis, visualization of source and header file include hierarchies, call hierarchy browsing, and class hierarchy browsing. Several existing source code analysis features of CDT have been significantly improved as well.

### Target Management/Remote System Explorer (Eclipse DSDP-TM Project)

Target Management provides Eclipse with 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.

### Device Debugging (Eclipse DSDP-DD Project)

Device Debugging provides Eclipse with the infrastructure to support device software development from hardware bring-up to application software development. Device Debugging provides Workbench users with enhanced capabilities for multiple-core debugging;

standard views for viewing registers, variables, expressions, and memory; and control over how debugger views are updated from the target device.

## Project System

The Workbench Project System allows developers to organize and manage the primary components in a device software development project, including source files and target systems. By design, Workbench enables users to manage multiple projects simultaneously.

Features of the Project System include the following:

- Accelerate setup of common project types with wizards and templates, including downloadable kernel modules (DKMs) and RTPs for VxWorks 6.x.
- Share projects among team members, enabling parallel development. Each team member can work in one or more workspace with local copies of source files, builds, and debugging.
- 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:

- The managed application build process, based on make, is simplified through a series of wizards and self-guided dialogs. Through the Build System, you can specify how you want to generate makefiles specific to a project.
- Automatically generate dependencies (compiler-dependent).
- Generate include search paths through source code analysis.
- GUI configuration of build tools and build specifications can be designated at the project, build target, folder, or file level, with property sheets for advanced customization.
- The compiler configuration GUI allows custom compiler target and optimization settings to be specified at the project, build target, folder, or file level. The GUI supports both Wind River Compiler and Wind River GNU Compiler.
- Easily switch between build specifications.
- Multiple targets can be supported per project, including specific build settings and flexible content definition (including exclusion patterns, etc.).
- The command-line build capability can be used for scripted and nightly builds.
- Default settings can be set, which shortens the time to an initial build.
- There is VxWorks shared library build support.
- Flexible content definition, file-level build, settings, and multiple build targets provide expert-level control and customizability.
- Wizards create new projects, such as shared libraries, VxWorks file system, downloadable modules, bootable ROMs, and RTPs.
- Options for user-defined builds using existing makefiles.
- 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.

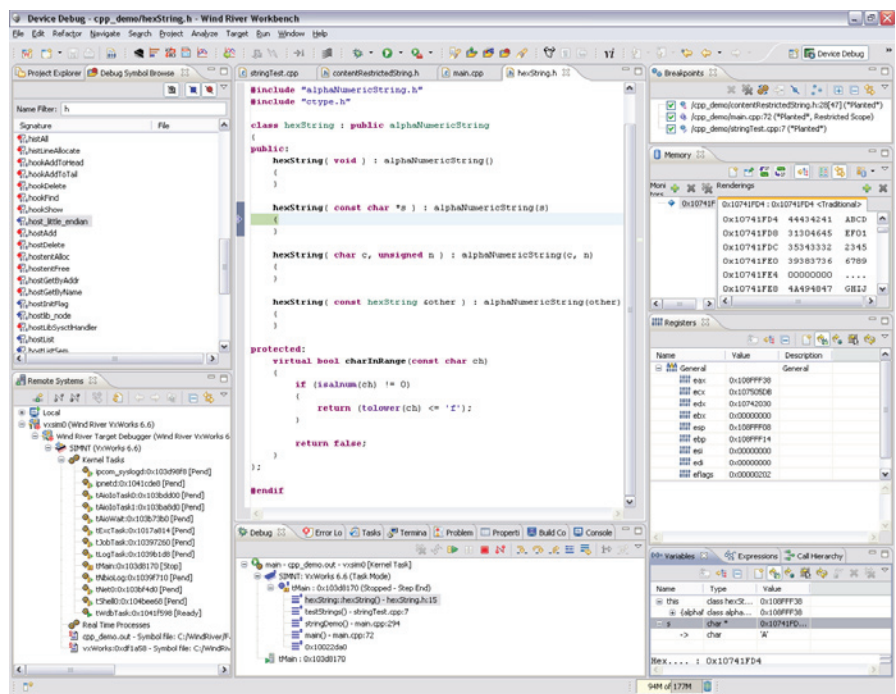


Figure 9: Workbench Debugger

### Wind River Compilers

Wind River provides two compilers for use in Wind River Workbench when developing with VxWorks 6.x: Wind River Compiler and Wind River GNU Compiler. Both compilers are included and supported as part of Wind River Workbench for VxWorks 6.x.

#### Wind River Compiler

Wind River 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 Compiler produces reliable, tight, and fast-executing code.

Features of Wind River 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 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.

For more detailed information, see the Wind River 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
- id, the programmable static linker
- as, the portable assembler
- Binary utilities

## 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. Multicontext 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—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.

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 avoid viewing relevant data for only one processor, process, or thread at a time, or 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 multithreading 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 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 RTP. 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.

Additional 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
- 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, formerly known as VxSim, is a complete prototyping and simulation tool for VxWorks 6.x applications. 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 connec-

tion, allowing complete configuration and debugging control through standard interfaces.

VxWorks Simulator is a native application that has been ported from the VxWorks 6.x operating system to accurately implement the sophisticated features of VxWorks 6.x, including RTPs, memory protection, file systems, and UNIX-style networking (TCP/IP, rlogin, etc.). The simulator also provides network simulation capabilities that let you create complete simulations of complex networks consisting of multiple IPv4, IPv6, or other protocols, subnets, and routing systems. The vxsimnetd tool allows large networks of VxWorks targets to be built on the host for packet drop simulation.

The simulator runs on your chosen host workstation, decreasing the need for evaluation hardware early in the development cycle. On multiple-processor host systems, the simulator supports VxWorks running in SMP mode. It also provides easy access to the host operating system API, so you can use the host facilities and peripherals in your simulation. For instance, a PCI card used in your final system can be installed on the host machine, then accessed by the simulator.

#### 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, previously known as WindSh, 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
- Interactive exploration of the operating system by calling any platform routines and APIs
- Interactive exploration of VxWorks 6.x RTPs
- Debug 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.

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
- 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)

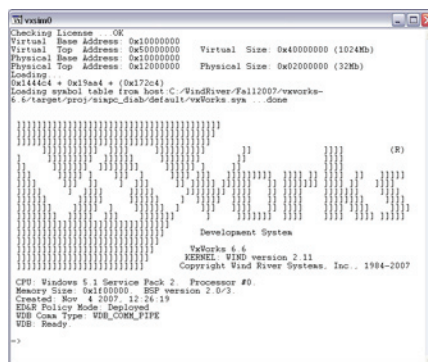


Figure 10: VxWorks Kernel Shell

## 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
- 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

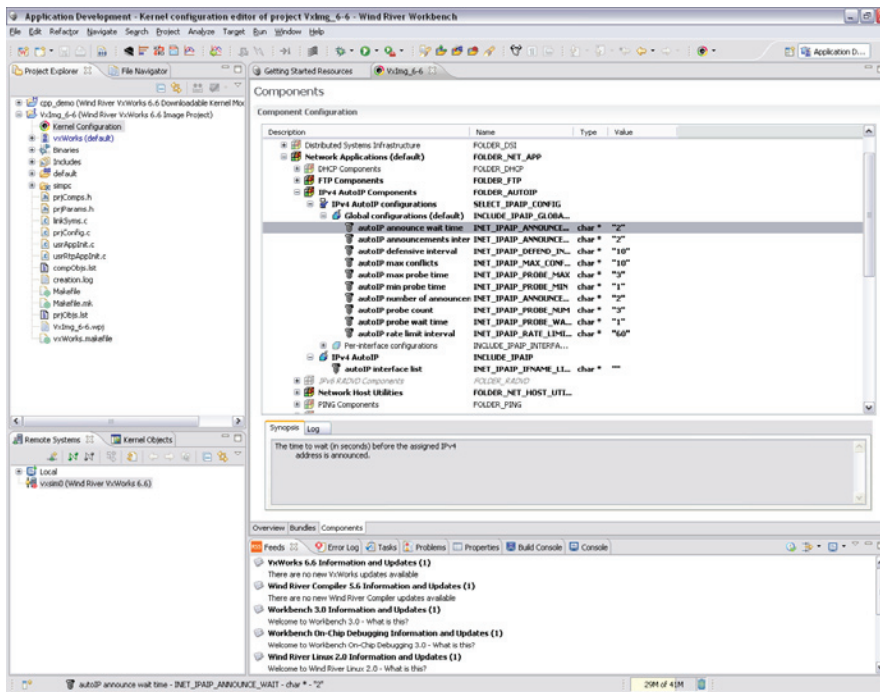


Figure 11: VxWorks Kernel Configurator

- 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
- 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

## 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

## 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.

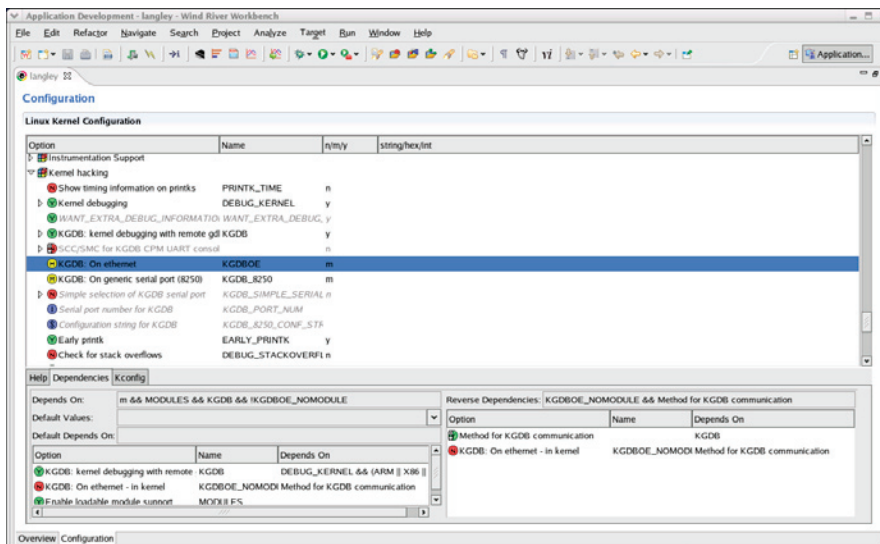


Figure 12: Linux Kernel Configuration Tool

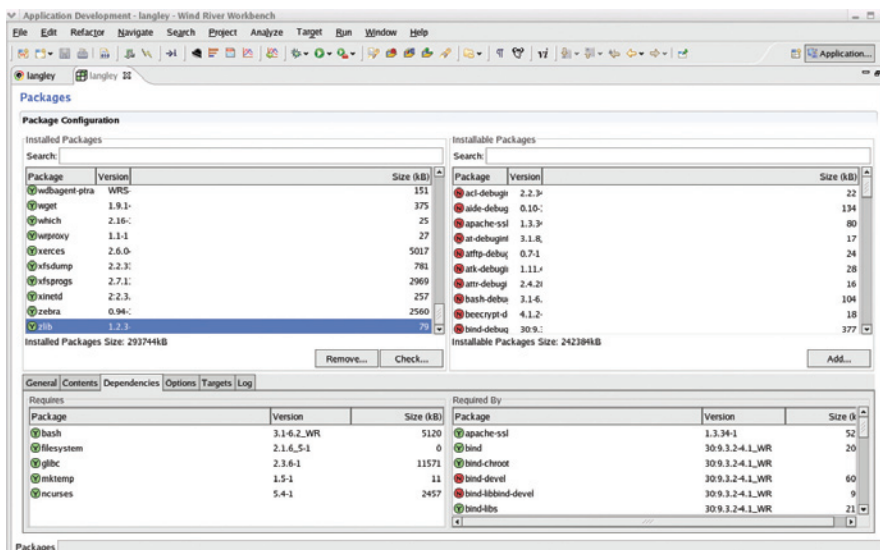


Figure 13: Linux User Space Configuration Tool

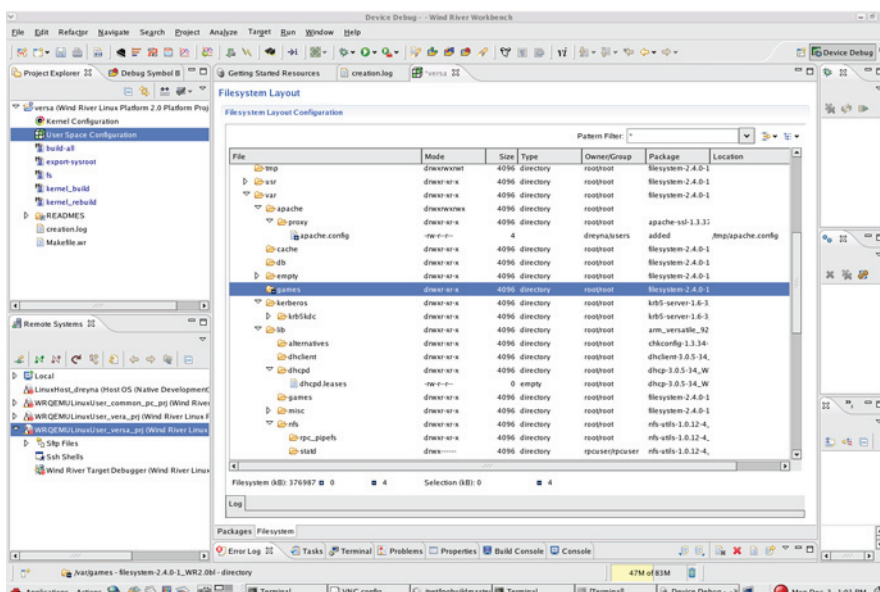


Figure 14: Linux File System Configurator

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 new 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 sym-links). 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.

## Run-Time Analysis Tools

Workbench contains six run-time analysis tools, five of which were previously known as the ScopeTools suite. 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 multicore 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.



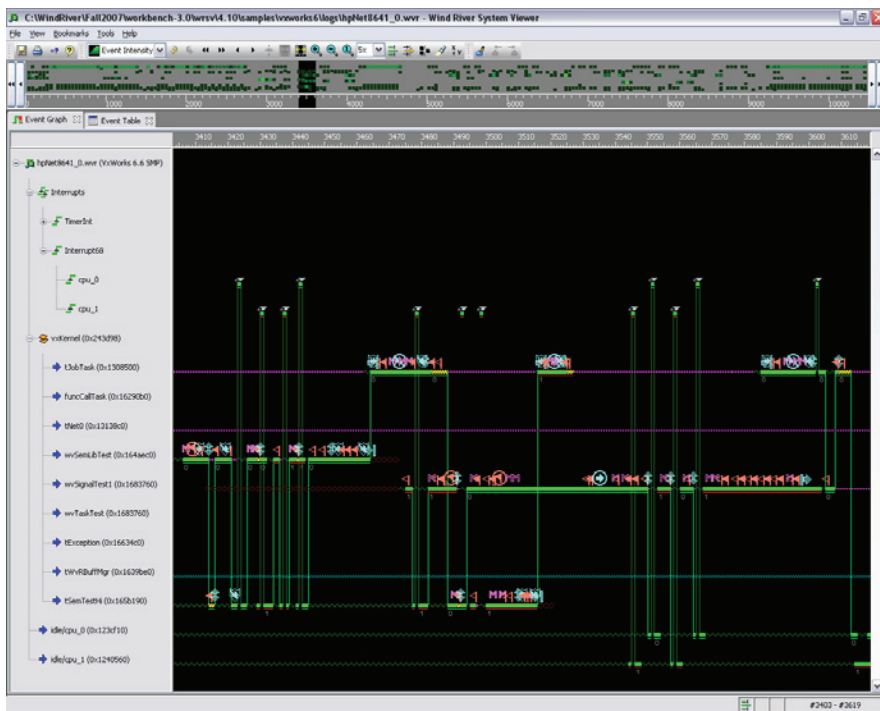


Figure 15: System Viewer

- 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

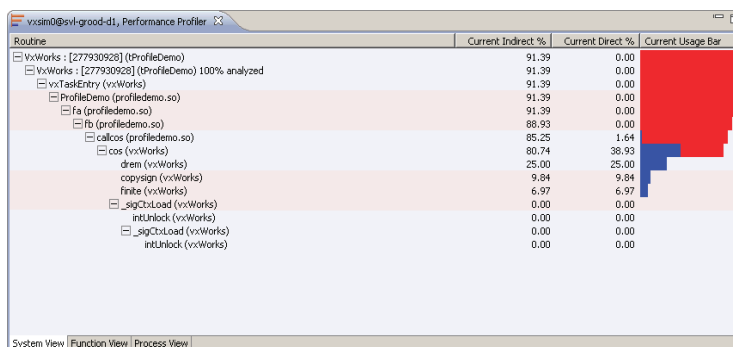


Figure 16: Performance Profiler

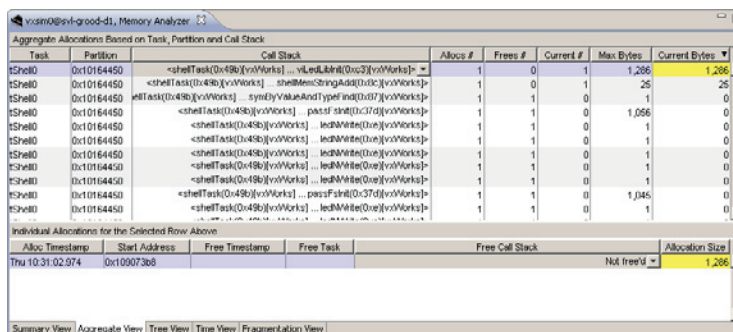


Figure 17: Memory Analyzer

- Performance problems, such as priority setting, resource contention, and mutual exclusion
- Timing problems arising from the interaction of interrupts and tasks

## Performance Profiler (Formerly ProfileScope)

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.

## Memory Analyzer (Formerly MemScope)

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 non-characterized memory leaks. Wind River Memory Analyzer provides greater



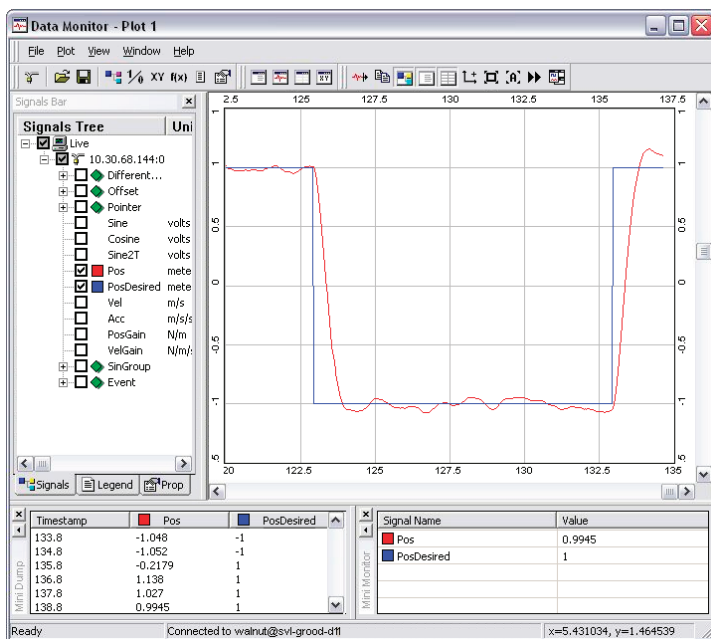


Figure 18: Data Monitor

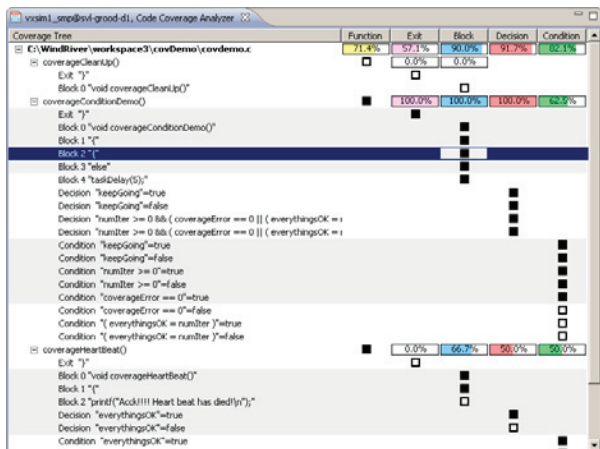


Figure 19: Code Coverage Analyzer

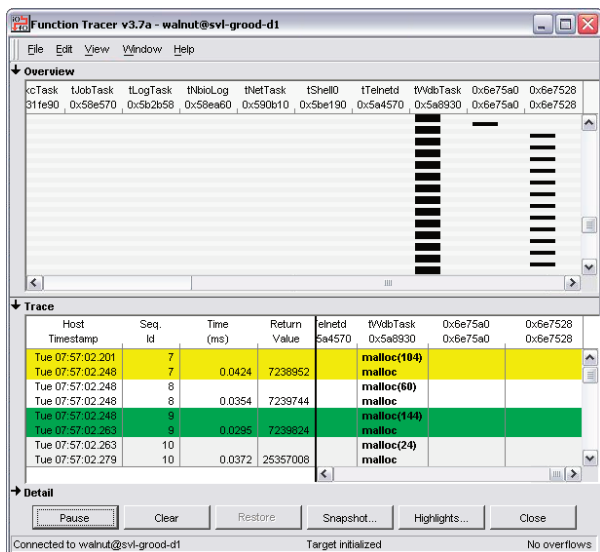


Figure 20: Function Tracer

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.

### Data Monitor (Formerly StethoScope)

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 (Formerly CoverageScope)

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.

### Function Tracer (Formerly TraceScope)

Trace your code execution in real-time by monitoring function call sequences as your code executes. Included with the function call displays are the parameters, as well as the returned values, to allow you to identify when function behavior and execution timing change. Function Tracer is only available on hardware platforms running VxWorks.

## Optional Add-Ons

### Wind River Workbench, On-Chip Debugging Edition

The Workbench development environment can be enabled for on-chip debugging. Workbench On-Chip Debugging, along with Wind River ICE, Wind River Trace, 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.

On-chip debugging provides developers with complete system-level control of their environment at all times, enabling more efficient and effective hardware bring-up, firmware development, and device driver and BSP generation.

On-chip debugging 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:

- On-chip debugging target connection manager
- On-chip debugging command shell
- On-chip debugging console
- Flash programming
- Hardware and memory diagnostics
- Configuration options

- JTAG editor
- Extensions to Register view, including the following:
  - Bit-level register details
  - Additional peripheral register support for most processors
  - Emulator initialization values and enablement
- On-chip debugging user's perspective within Workbench
- Wind River Trace (may require additional hardware, to be purchased separately)
- Firmware update
- Cache Memory view
- Statistical performance analyzer (PFA)
- On-chip debugging reset and download/launch
- Linux, VxWorks 5.5, VxWorks 6.x, and ThreadX OS awareness via JTAG
  - All hosts
  - All supported targets with BSP available
  - Debugging of SMP systems

### Wind River ICE

Wind River ICE is a network-based emulator 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.

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

In addition, Wind River ICE has the ability to support externally trace-buffered SoCs, such as AMCC's PPC405 and 440 families and Freescale's MCF5xxx family. This is supported through addition of our Wind River Trace module, an adapter module that plugs into the end of the Wind River ICE cable. For more information, see the Wind River ICE product note.

### Wind River Trace

Wind River Trace allows developers better visibility into the hardware/software interaction within their device. It provides a GUI within Workbench to show trace configuration parameters and display trace

data. Wind River Trace also includes a hardware adapter for Wind River ICE, enabling it to capture and buffer more than 900,000 lines of trace data from the target. For more information, see the Wind River Trace product note.

### 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 (Formerly Unit Tester)

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 Device Management

Wind River Device Management consists of two interoperable products that create a powerful, enterprise-wide infrastructure to enable development and software quality assurance (SQA) engineers to streamline the SQA process, and field engineering teams to streamline the support process. Benefits include faster time-to-market, higher-quality products, and the ability to rapidly and remotely

diagnose and repair software defects in deployed devices. This leads to lower support costs, increased system uptime, and improved customer satisfaction. Both products are available for VxWorks 5.5 and VxWorks 6.x, as well as for Wind River Linux 1.x and 2.0.

### **Wind River Lab Diagnostics**

Wind River Lab Diagnostics is an enterprise-class server application that enables engineers to comprehensively test applications and resolve issues encountered in test labs. It allows companies to greatly enhance productivity by streamlining system integration, software verification, and product validation, spreading the test load across distributed teams, enabling fact-based diagnostics of multiple devices, and eliminating time-consuming instrumentation-build-test cycles. Lab Diagnostics also enables manufacturers to “design in” supportability, which eases and speeds issue resolution in deployed devices. This standalone product is interoperable with Wind River Workbench and is sold as an add-on to Wind River platforms. For more information, see the Wind River Lab Diagnostics product note.

### **Wind River Field Diagnostics**

Wind River Field Diagnostics is a scalable, field diagnostics system that enables support engineers to securely collect and manage deployed device data to diagnose and correct software faults. Field Diagnostics is a secure, enterprise-wide infrastructure that includes a site-installed application for onsite device data collection and diagnostics, as well as an enterprise application to manage data aggregation, analysis, and archiving from worldwide deployments. It links device manufacturers with device users through a secure data exchange infrastructure. With Field Diagnostics, device manufacturers can improve uptime, streamline support and maintenance operation, reduce support costs, and increase service revenue. This standalone product is interoperable with Wind River Workbench and is sold as an add-on to Wind River platforms. For more information, see the Wind River Field Diagnostics product note.

## **Technical Specifications**

### **Workbench 3.0 Host OS Support**

- Windows XP Professional, Service Pack 2, x86
- Windows Vista (Business and Enterprise), x86
- Red Hat Enterprise Linux 4 Workstation, Update 5
- Red Hat Enterprise Linux Desktop with Workstation option 5, x86 or x86-64
- Red Hat Fedora Core 7, x86
- SUSE Desktop Linux 10, Service Pack 1, x86
- SUSE Linux/openSUSE 10.2, x86
- Solaris 9, update 9/05 (GTK only)
- Solaris 10

### **Workbench 3.0 Target OS Support**

- VxWorks 6.3, 6.4, 6.5, 6.6 platforms
- VxWorks 5.5
- VxWorks 653
- Wind River Linux 1.4, 1.5, 2.0 based platforms
- Native Linux development on supported Linux hosts
- ThreadX 4.0 and 5.0 (Workbench, On-Chip Debugging Edition only)
- Customizable target OS awareness capability for Workbench, On-Chip Debugging Edition; enables support for other target operating systems to be added

### **Workbench 3.0 Target Processor Support**

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

### **Workbench Target Architecture Support for VxWorks 5.5**

- Available only for Windows hosts: PowerPC architecture

### **Workbench Target Architecture Support for VxWorks 6.x**

- ARM
- ColdFire
- Intel
- Intel XScale
- MIPS
- PowerPC
- Renesas SuperH

### **Workbench Target Architecture Support for Linux**

- ARM
- Intel
- Intel XScale
- MIPS
- PowerPC

### **Workbench Target Architecture Support for VxWorks 653**

- PowerPC

### **Target Architecture Support for Workbench, On-Chip Debugging Edition**

- ARM
- ColdFire
- Intel XScale
- MIPS
- PowerPC

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 comprehensive Device Software Optimization (DSO) 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.

### Education Services

Education is fundamentally connected not only to individual performance but also to the success of a project or entire company. Lack of product knowledge can translate into longer development schedules, poor quality, and higher costs. The ability to learn—and to convert that learning into improved performance—creates extraordinary value for individuals, teams, and organizations. To help your team achieve that result, Wind River offers flexible approaches to delivering product education that best fits your time, budget, and skills development requirements.

#### Personalized Learning Program

Wind River offers a unique solution to minimize the short-term productivity drop associated with the process of adopting new device software technology, and to optimize the long-term return on investment in a new device software platform. The Wind River Personalized Learning Program delivers the right education required by individual learners to accomplish their jobs. The program identifies work-related skill gaps, generates development plans, materials, and learning events to address these skill gaps, and quantifies the impact of the development activities for each individual user.

This programmatic, focused, and project-friendly approach to skills development results in a significant increase in the personal productivity of your team, improved efficiency in the processes they employ, and faster adoption of the technology you have purchased. Personalized Learning Programs deliver improved business performance; customers have reported a return on investment ranging from 18 to 80 percent over a traditional training approach.

Consult your local Wind River sales representative for more information on Personalized Learning Programs.

#### Public Courses

Wind River's public courses are scheduled for your geographical convenience. They are conducted over one to five

days, using a mixed lecture and interactive lab classroom format that leverages the experience of Wind River instructors and other course participants. Courses provide a fast, cost-effective way for students to become more productive in Wind River technology.

Benefits of public courses include the following:

- A conceptual introduction that orients students to the subject matter
- A selective examination of the details, focusing on the most commonly used areas, or on areas with which users tend to be the least familiar
- Personal guidance and hands-on application of individual tools and course concepts
- The chance to grasp device software concepts, as well as the fundamental issues involved in real-time design
- The knowledge needed to develop device drivers, perform hardware porting, or develop applications
- Answers to specific questions about topics addressed in the course

Consult your local Wind River sales representative for course schedules and fees.

#### Onsite Education

If you have a large project team or a number of new users, you may benefit from custom onsite education. Instructors will consult with you and, based on the workshop series curriculum, determine which topics should be included and emphasized. This type of education offers an opportunity for one-on-one discussions with our instructors about your specific project needs, technical requirements, and challenges—all in the comfort of your own office.

Advantages of onsite education are as follows:

- Your entire team gains a common knowledge base.
- Onsite education helps ensure that knowledge and skills will transfer from the classroom to your workplace.
- Use of your location saves employees travel expenses and time away from the office.

Consult your local Wind River sales representative for further information about onsite education.



## Support Services

Wind River Customer Support, a Support Center Practices (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, please 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, the 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 is the global leader in Device Software Optimization (DSO). 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)

© 2007 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. 12/2007