

Wind River VxWorks Platforms 3.7

Table of Contents

| | |
|---|----|
| Platforms Available in | |
| VxWorks Edition | 2 |
| New in VxWorks Platforms 3.7 | 2 |
| VxWorks Platforms Features | 3 |
| VxWorks Real-Time Operating System | 3 |
| Compatibility | 3 |
| State-of-the-Art Memory Protection | 4 |
| VxBus Framework | 4 |
| Message Channels and TIPC | 4 |
| Memory Management | 5 |
| Error Management | 5 |
| Processor Abstraction Layer | 6 |
| Operating System Scalability and Performance Tuning | 6 |
| File Systems | 6 |
| Wind River Network Stack | 6 |
| Wind River PPP | 9 |
| Wind River USB | 9 |
| VxWorks 6.7 Multiprocessing | 9 |
| Middleware Technology | 16 |
| Security | 16 |
| Management | 18 |
| Distributed Messaging and Services | 20 |
| Bridging and Routing | 21 |
| Graphics and Local User Interface | 21 |
| Connectivity | 21 |
| Wireless | 22 |
| Wind River Workbench Development Suite | 22 |
| Eclipse | 23 |
| Project System | 23 |
| Build System | 23 |
| Command-Line Project and Build System | 23 |
| Index-Based Global Text Search-and-Replace | 23 |

| | |
|---|----|
| Wind River Compiler and Wind River GNU Compiler | 24 |
| Workbench Debugger | 24 |
| VxWorks Simulator | 24 |
| Workbench VxWorks Source Build Configuration | 25 |
| VxWorks 6.x Kernel Configurator | 25 |
| Host Shell | 25 |
| Kernel Shell | 25 |
| Run-Time Analysis Tools | 26 |
| Optional Add-Ons | 26 |
| VxWorks 6.7 Multiprocessing | 26 |
| Wind River Test Management and Wind River Lab Diagnostics | 26 |
| Wind River Workbench, On-Chip Debugging Edition | 27 |
| IPL Cantata++ for Wind River Workbench | 27 |
| Datalight FlashFX Pro Version 3.3 | 27 |
| Technical Specifications | 27 |
| VxWorks 6.7 | 27 |
| Workbench 3.1 | 28 |
| Architectures, Hosts, and Board Support Packages | 28 |
| Supported Target Architectures | 28 |
| Supported Hosts | 28 |
| Supported Board Packages | 28 |
| Partner Ecosystem | 28 |
| Professional Services | 29 |
| Installation and Orientation Service | 29 |
| Education Services | 29 |
| Public Courses | 30 |
| Onsite Education | 30 |
| Support Services | 30 |
| North America, South America, and Asia/Pacific | 30 |
| Japan | 30 |
| Europe, Middle East, Africa | 30 |

The market for secure, intelligent, connected devices is constantly expanding. Embedded devices are becoming more complex to meet market demands. Internet connectivity allows new levels of remote management but also calls for increased levels of security.

More powerful processors are being considered to drive intelligence and higher functionality into devices. Because real-time and performance requirements are nonnegotiable, manufacturers are cautious about incorporating new technologies into proven systems. To succeed, companies must optimize their device software across the entire product life cycle: from design through development, from QA to the remote management of deployed devices. The challenge for many device developers is in balancing the need for increased speed, greater efficiency, and lower cost with an acceptable level of development risk.

Wind River VxWorks platforms meet this challenge with a Device Software Optimization (DSO) solution that combines VxWorks, the industry's leading commercial-grade real-time operating system (RTOS); Wind River Workbench, the premier open device software development suite; and essential security, device management, and connectivity middleware, including drivers and protocols for connected devices on the factory floor, wireless peripherals, and other devices within the network infrastructure. The VxWorks platforms are backed by Wind River's 25 years of device software industry experience, a world-class support organization, a comprehensive partner ecosystem, and a specialized professional services team.

The Wind River VxWorks platforms are optimized develop-and-run solutions for a range of devices, from aerospace and defense (A&D) applications to networking and consumer electronics, robotics and industrial applications, precision medical instruments, and car navigation and telematics systems. The platforms provide a robust foundation for companies that need to leverage their investment in proprietary intellectual property. VxWorks has been deployed successfully in hundreds of millions of devices worldwide.

Platforms Available in VxWorks Edition

- Wind River General Purpose Platform:** This versatile platform is used for devices ranging from aerospace and defense, automotive telematics, and small-footprint consumer devices to industrial devices and networking equipment.
- Wind River Platform for Automotive Devices:** Designed for the development of applications that require high reliability, low power consumption, and a small memory footprint, its targets include both vehicle and security control systems (powertrain, engine, ABS, crash and airbag sensors, window/door entry) and in-vehicle systems (digital dashboard displays, navigation systems, telematics systems, and entertainment systems).
- Wind River Platform for Consumer Devices:** This offers a fast-boot, small-footprint run-time environment ideal for memory-constrained devices. Platform targets include digital video, mobile handheld, digital imaging, and broadband access devices.
- Wind River Platform for Industrial Devices:** This provides industrial device manufacturers with essential multimedia and connectivity run-time technologies, including drivers and protocols for connected devices on the factory floor, wireless peripherals, and other devices within the network infrastructure. Targets include industrial automation, building automation, and medical, transportation, and test and measurement devices.
- Wind River Platform for Network Equipment:** This enables customers to rapidly create, test, deploy, maintain, and manage high-quality network infrastructure devices. The platform offers an extensive suite of security protocols to protect network data. It is ideally suited for wireless infrastructure, enterprise network, core networking,

Development Suite

| |
|----------------------|
| Wind River Workbench |
|----------------------|

Software Partners

| | | |
|-------------------|------------------------|-----------------------------|
| Ada Support | Advanced Flash Support | Advanced Security |
| Browsers | CAN | Common Internet File System |
| Databases | Design Tools | Graphics |
| High Availability | Java | Others |

Additional Middleware*

| | | | |
|----------------------------|----------------------|------------------|-----------------|
| Wireless Ethernet | Mobile IPv4/IPv6 | 802.1Q VLAN | Media Library |
| SSL & SSH | IPsec & IKE | NAT/Firewall | IGMP/MLD |
| RADIUS and Diameter Client | Wireless Security | Crypto Libraries | EAP |
| SNMP v1/v2/v3 | Web Server | CLI/MIBway | Learning Bridge |
| VRRP | Web Svcs-Interop/SEC | DCOM | CAN/OPC |

Base Middleware**

| | | |
|-------------------------|---------------------------|--------------------|
| TIPC | Distributed Shared Memory | USB 1.1, 2.0 |
| dosFs | Flash Support (TrueFFS) | Highly Reliable FS |
| IPv4/IPv6 Network Stack | | PPP |

Operating Systems

| |
|---------------------------------|
| VxWorks/VxWorks Multiprocessing |
|---------------------------------|

Hardware Partners

| |
|--|
| Reference Designs, Semiconductor Architectures |
|--|

Services

| | |
|-------------------------------------|------------------------|
| Education Services and Installation | Platform Customization |
| System Design | Design Services |
| Hardware/Software Integration | |

Figure 1: VxWorks platform components

* Included in VxWorks industry-specific platforms
 ** Included in all VxWorks platforms

network edge, WiMAX/LTE infrastructure, and broadband access devices.

New in VxWorks Platforms 3.7

The latest release of VxWorks platforms includes updated and enhanced run-time components in VxWorks, as well as industry-specific middleware technologies. These components can also be used with VxWorks symmetric multiprocessing (SMP) and asymmetric multiprocessing (AMP) technologies, found in the VxWorks 6.7 Multiprocessing add-on, to exploit the capabilities of the latest multicore processors.

The Wind River Workbench 3.1 development suite includes enhancements to the Workbench core and improved VxWorks and multiprocessing platform support, on-chip debugging, and diagnostics tools. Wind River's device management tools (add-on products to our VxWorks-based platforms) provide a powerful, enterprise-wide infrastructure that enables development, test, and field engineering teams to collect and aggregate data to diagnose and repair faults in running software at every phase in the device life cycle.

This release also includes Wind River Network Stack, using Wind River Advanced Networking Technologies to

provide significant performance, scalability, and feature improvements to the networking capabilities of the platform.

The VxWorks 3.7 platforms introduce the following new features, functionalities, and enhancements:

- VxWorks 6.7
- Support for AMP; AMP technology allows multi-OS systems on multicore hardware
- SMP performance enhancements
- VxWorks Source Build (VSB) projects for footprint and performance optimization
- Support for overlapped virtual memory for real-time processes
- Broad and deep networking functionality
 - Mobile IP features
 - Network Mobility (NEMO)
 - Mobile IPv6 node extensions
 - Support for the network access identifier
 - NAT support for the Session Initiation Protocol (SIP) Application Level Gateway (ALG)
 - Network Stack 6.7 enhancements
 - Enhancements to the DHCP implementation
 - Generation and validation of cryptographically generated addresses (CGAs) and CGA parameters
 - Support for setting the IPv6 address scope and zone for a mode
 - Support for
 - Gratuitous ARP
 - Robust Header Compression (ROHC)
 - Netlink and RTNETLINK
 - Secure Neighbor Discovery (SEND) protocol
 - MIB-II, using the M2 library m2Lib
 - ifLib and if6Lib interface configuration libraries
 - Custom (non-Ethernet) address resolution routines
 - Wind River RADIUS and Diameter 6.7, including
 - Support for Diameter agent (relay and proxy)
 - Diameter client error messages
 - Examples detailing use of RADIUS shell commands
 - New Wind River EAP module
 - Support for SMP and Wi-Fi multimedia enhancement (WME) within Wind River wireless Ethernet drivers

- Support for the dot1xPaeConfigGroup supplicant MIB within Wind River Wireless Security and introduction of new commands to support run-time configuration of multiple port access entity (PAE) instances
- Robust security
 - IKE support for traffic flow confidentiality (TFC) and extended sequence number (ESN) anti-replay support for high-speed IPsec implementations
 - IPsec offloading with SMP systems
 - IPsec support for ESN functionality, TFC, VPN-A, and VPN-B cryptographic suites as described in RFC 4308
 - Support for the pktflags keyword that filters IPsec, NAT, and tunneled packets within Wind River Firewall
 - Wind River SSH changes to sockets used in APIs, directory structure, and some default values
 - Wind River SSL 6.7 updated to version 0.9.8h of OpenSSL
- Network management
 - Ability to use a GET request to set WindMark values
 - API support for SNMPv3 INFORM PDU operations
- Wind River Web Services upgrade of the SOAP component to gSOAP 2.7.10 distribution
- Wind River Media Library
 - Adds general still image API with support for JPEG and PNG image formats
- New Workbench 3.1 features
 - Dynamic printf debugging support
 - VxWorks Source Build configuration tool
 - Debugger support for VxWorks AMP
 - Support for overlapping memory RTPs
 - User interface improvements for run-time analysis tools
 - Device debug alignment with Eclipse
 - Improved search and source code indexing

VxWorks Platforms Features

VxWorks Real-Time Operating System

VxWorks is the industry-leading, commercial-grade device software operating system. Its high determinism, high performance, modular scalability, and small footprint make more than 350 million devices worldwide run faster and more reliably. The next generation of VxWorks adds powerful new features and a focus on multiprocessing, openness, performance, reliability, and interoperability.

With VxWorks 6.7, users can do the following:

- Optimize developer productivity through open standards
- Leverage the power of multicore silicon to deliver higher performance and more capable devices
- Increase reliability through memory management unit (MMU)-based memory protection
- Accelerate time-to-market through enhanced error management
- Seamlessly migrate existing VxWorks-based IP and other existing IP, including open source
- Continue to deliver products that rely on the core attributes of VxWorks, including high performance, high reliability, high determinism, low latency, and modular scalability

Complementary solution elements include the integrated, Eclipse-based Wind River Workbench development suite, extensive middleware components, comprehensive processor and board support package (BSP) support, expert professional services, and the DSO industry's largest partner ecosystem of hardware, software, development tools, middleware, and applications providers.

Compatibility

VxWorks 6.x and VxWorks SMP are designed to ease migration from VxWorks 5.5. To that end, the kernel of VxWorks 6.x supports the VxWorks 5.5 kernel operating environment. Most BSPs, drivers, and kernel applications developed for or ported to VxWorks 5.5 will run in the VxWorks 6.x kernel. By default, the VxWorks 6.x kernel builds like that of VxWorks 5.5. VxWorks 6.7 is also backward-compatible with all previous versions of VxWorks 6. VxWorks SMP is API-compatible with the uniprocessor versions of VxWorks (with a few exceptions).

The VxWorks technical product documentation includes information about the following:

- Migrating VxWorks 5.5 kernel applications to the current kernel and user modes
- Porting BSPs and drivers
- Migrating code to VxWorks SMP
- Changes in networking and security technologies from VxWorks platforms 3.4 to this release

State-of-the-Art Memory Protection

VxWorks 6.x enables manufacturers to increase their device reliability through MMU-based memory protection. VxWorks introduces process-based, user-mode application execution in addition to its traditional kernel-mode execution. The kernel is protected from user-mode applications running in VxWorks real-time processes (RTPs). User-mode applications are also protected from each other.

Features of memory protection include the following:

- MMU-based memory protection provides isolation of the kernel from user-mode applications and of applications from each other, increasing device reliability.
- The standard, process-based programming model simplifies application development.
- VxWorks' preemptive, priority-based global task scheduler ensures real-time deterministic behavior.

- The ability to create private or public objects in the kernel and in RTPs offers flexibility to use objects that are protected from manipulation, or that can easily be shared among kernel and process tasks.
- The extensible system call interface enables application developers to employ custom-developed kernel services from user-mode execution.
- Support for shared libraries among RTPs improves code efficiency and reusability as well as speeding code development and debugging.

VxBus Framework

VxWorks 6.7 includes the VxBus framework, which device drivers plug into and coordinate device-related operations among device drivers, the BSP, and the VxWorks kernel. It provides a formal interface between device drivers and hardware. It also provides a uniform set of APIs that can be used by higher-level features such as file systems, network stacks, power management frameworks, and so on. VxBus provides the following:

- Device discovery mechanisms
- Common infrastructure between devices, middleware, and the OS
- Support for different bus topologies

These features allow device drivers to be decoupled from the BSPs they ship with. In turn, this allows drivers to work with minimal or no modifications on many BSPs. The task of writing BSPs for VxWorks is made simpler with the VxBus framework.

Message Channels and TIPC

With memory protection comes the challenge of segmentation of applications into protected memory spaces. Message channels are a new connection-orientated, bidirectional messaging mechanism introduced in VxWorks 6.0 that allow tasks to communicate across memory boundaries and complement traditional communication mechanisms provided in VxWorks.

Tasks located in different processes or in the kernel can establish connections with each other, independent of their locations. Communication can take place between one task in a process and another in the kernel, or between tasks in separate processes, or between tasks in the same process. In VxWorks 6.1 and later, message channels use the open source, industry-standard TIPC protocol to extend messaging to tasks in multi-processor systems that can run VxWorks, Linux, or any other operating system that supports TIPC. The VxWorks 6.7-based platforms support TIPC version 1.7.

Features of TIPC 1.7 include the following:

- Open-standard TIPC
 - High-speed, reliable message-passing service
 - Location transparency with logical address and internal address translation table
 - Lightweight, connectionless, or connection-orientated communication modes
 - No message losses
 - No message duplicates
 - Uninterrupted message sequence order
 - Reliable multicast messaging
- OS independence, with interoperability between VxWorks and Linux

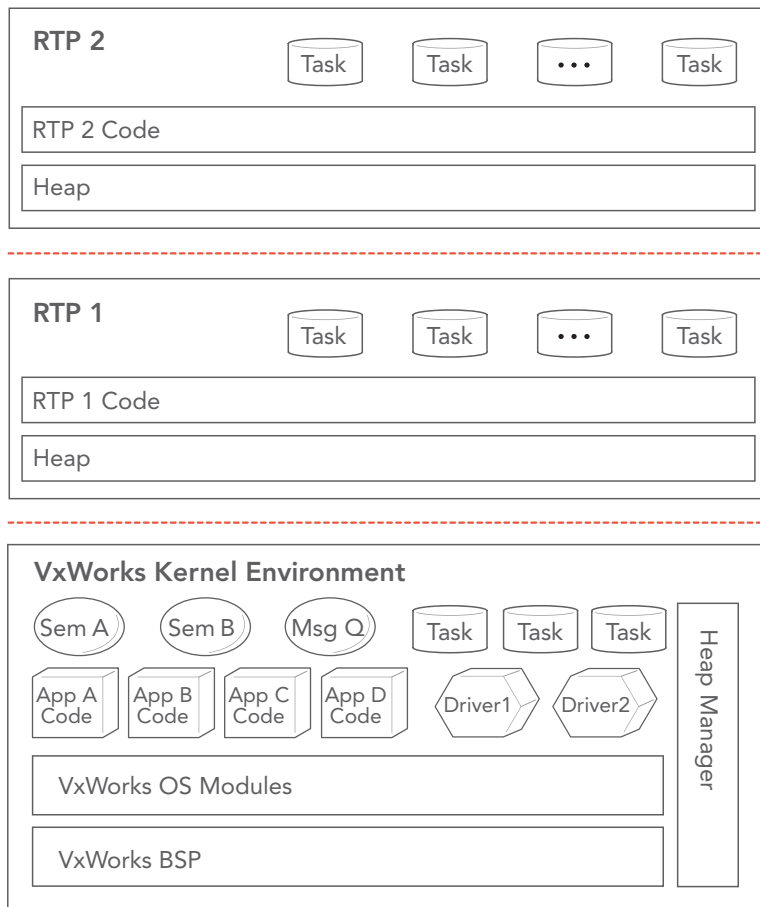


Figure 2: VxWorks kernel environment and two RTPs

- Scalability from single processor to multicore to cluster of nodes
- Extensible framework for failover
- Suitability for systems where short, real-time critical transactions are performed
- Compatibility with traditional Wind River intertask communication methods
- Multicenter network topology support

Memory Management

Although VxWorks 6.x provides MMU-enabled memory protection, it continues to use the nonoverlapped addressing model in all earlier versions of VxWorks. VxWorks 6.7 introduces an overlapped virtual memory model for RTPs, allowing RTPs to share the same range of virtual addresses. The nonoverlapped model is still the default mode. Overlapped mode can be selected through a configuration parameter.

The nonoverlapped memory model promotes backward-compatibility with legacy VxWorks code and features the following benefits:

- High determinism and low latency are maintained by not requiring memory to be mapped in or out and by eliminating unnecessary cache flushes.
- Address pointers are unique, enabling reuse of existing VxWorks drivers and applications, as well as simplifying new driver and application development.

The overlapped virtual memory feature closely resembles the traditional user virtual address space used for processes on operating systems such as UNIX, Linux, and Windows. All RTPs in the system get exclusive use of a common range of virtual addresses. Each RTP has a private view of this common virtual memory area. Shared data regions and shared libraries are mapped at other locations. When using the overlapped virtual memory feature, RTP executables can be fully linked and therefore not require symbol relocations at load time that are otherwise necessary in nonoverlapped mode. Fully linked RTP executables therefore load significantly faster than RTPs running in nonoverlapped mode.

Other VxWorks 6.x memory management enhancements include the following:

User Application: RTP

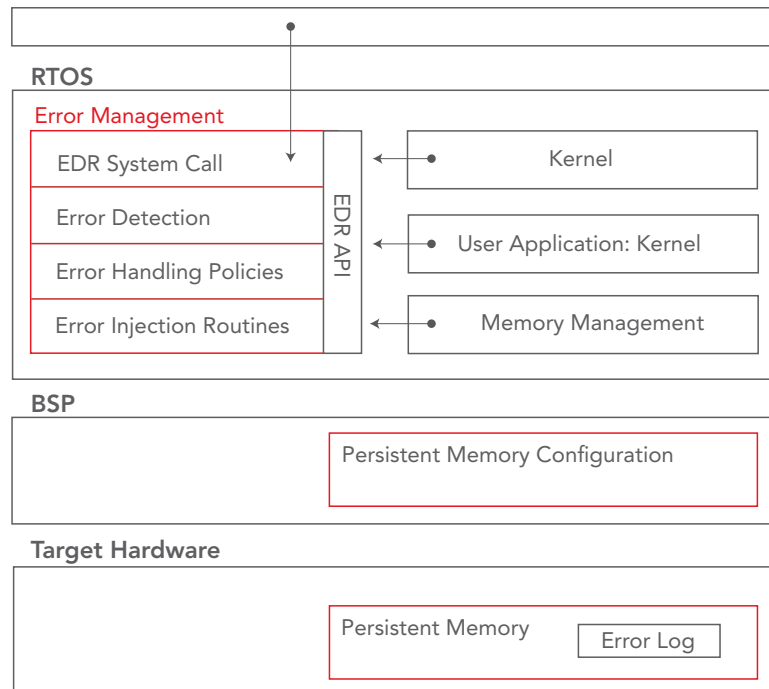


Figure 3: Error management

- **Automatic resource reclamation:** Maximizes memory availability and helps prevent memory leaks, improving device robustness
- **Improved memory allocation using a “best-fit” algorithm:** Reduces memory fragmentation and achieves near deterministic memory allocation/free performance
- **User-mode heap and memory partition support:** Enables RTPs and user-mode execution
- **Developer-replaceable user-mode heap allocator:** Maximizes system design flexibility
- **Heap instrumentation for heaps and memory partitions in both the kernel and RTPs:** Assists in diagnosing common memory problems by detecting and reporting memory errors
- **Tight integration with error management:** Maximizes reliability and minimizes time-to-market
- **Compiler-assisted code instrumentation:** Improves static code analysis of memory errors

Error Management

Wind River provides an error management framework to help customers isolate, diagnose, and correct error conditions encountered during development and testing. With this framework, customers can manage failures, minimizing the need

to reproduce failures in order to diagnose the device. The error management framework includes memory error detection and error reporting technology and provides a foundation for debugging device software out of the box and is extensible to enable customers to design reliable devices.

Error management capabilities are available in both the kernel and in RTPs and include the following:

- ISR and task stack overrun and underrun detection
- Code corruption detection
- Null pointer usage detection
- Heap block overrun and underrun detection
- Heap usage tracking and leakage detection
- RTP error detection
- API support for kernel application errors
- API support for user-defined application errors
- Customizable error-handling policies
- Comprehensive error records with common headers, key OS information, hardware information, and user-defined optional string
- User-defined memory storage of error records
- Integration and coordination with Wind River device management tools

Processor Abstraction Layer

With VxWorks 6.x, the processor abstraction layer (PAL) provides the capability to extend OS support to similar architecture families. An architecture-specific PAL defines the functional interconnects for each architecture family and, to a degree, abstracts the functionality of that hardware feature from the operating system. While this is transparent to the application, the presence of a PAL for an architecture family facilitates the quick adoption of processor variants, improving time-to-market for VxWorks 6.x customers. In VxWorks 6.7, the PAL also supports VxWorks SMP and AMP for the respective supported processors.

Operating System Scalability and Performance Tuning

VxWorks 6.7 introduces a new kind of project named VxWorks Source Build (VSB) projects. VSB projects are set up to build VxWorks from source by applying key options to optimize performance and scale footprint. VSB projects can be set up for any BSP, using the kernel configurator in Wind River Workbench.

Predefined VSB build options are available for users to select or unselect from their VSB projects. A given set of options can be combined into a profile. A maximal performance profile has been defined, which excludes RTP support, Wind River System Viewer instrumentation, and some error checks. This profile allows kernel-only systems to be built, providing up to 20 percent kernel performance improvement over a default VxWorks image project build.

VxWorks Source Build and the kernel component configuration, together with an appropriately written BSP, can allow VxWorks image footprints to be significantly reduced to fit the needs of memory-constrained devices.

File Systems

VxWorks includes a FAT-compatible file system called dosFs. VxWorks versions 6.2 and later also include a new file system framework that enhances the capabilities, performance, and reliability of VxWorks-based file systems.

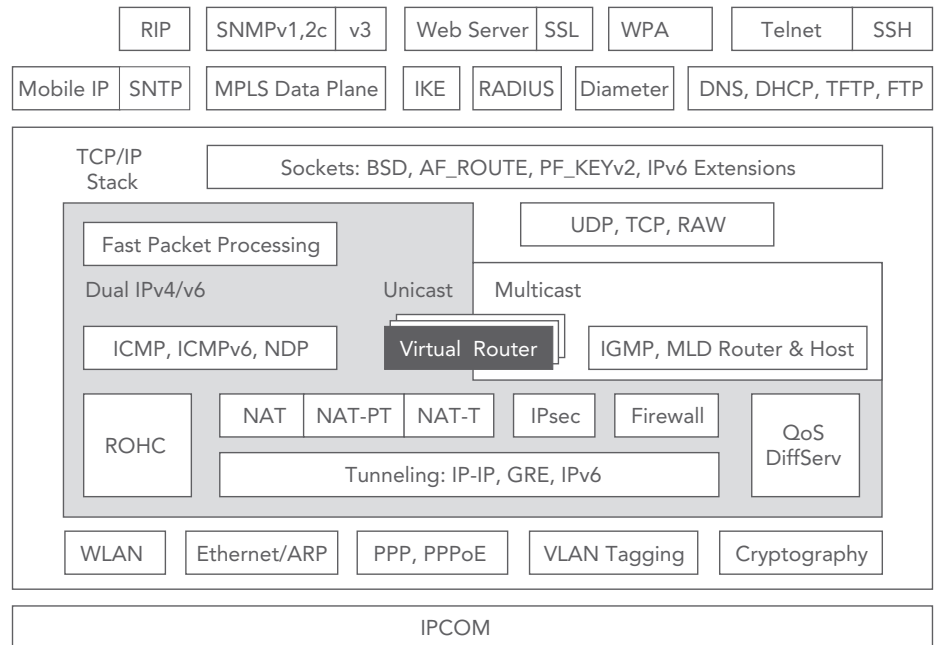


Figure 4: Wind River Network Stack and related components

Features of the file system framework include the following:

- Support for multiple file systems, removable media, and automatic file system detection
- Cache write-through option for dosFs, ensuring data is committed to the file system, enhancing reliability
- Improved CheckDisk for dosFs using the “clean bit,” allowing the file system to skip CheckDisk for improved boot-up performance
- Unicode filename support

The transaction-based highly reliable file system (HRFS) provides complete power-safe reliability (except for the case where a write action is interrupted). In VxWorks 6.7, HRFS is enhanced to support configurable commit policies, which allow finer control and better file system performance without sacrificing reliability.

Wind River Network Stack

Wind River Network Stack is a full-featured IPv4/IPv6 dual stack specifically designed for next-generation device software applications. Wind River Network Stack incorporates Wind River Advanced Networking Technologies, which provides IPv4/IPv6 networking, security, wireless, and mobility

capabilities for compatibility with a wide range of networking environments. It provides rich networking features with high-performance, scalable implementation and a small footprint. It conforms to relevant industry standards and provides a clean code structure for easy integration and faster implementation. In order to ensure high quality, conformance to standards, and interoperability with other network devices, Network Stack has undergone extensive third-party testing and external test lab validation. As a result, Network Stack is one of the first few TCP/IP stacks in the industry to receive the IPv6 Ready Phase II logo.

Figure 4 depicts how Wind River Network Stack is integrated with all the relevant networking technologies.

IPv4/IPv6 Dual Stack

Network Stack offers greater flexibility in configuring the stack to support IPv4 and/or IPv6. The stack may be built for IPv4 only, for IPv6 only, or for both protocol versions.

Multicast is supported via the host and router sides of the Internet Group Multicast Protocol (IGMP) and Multicast Listener Discovery Protocol (MLD).*

*These components are included in the industry-specific platforms only.

802.1Q VLAN*

The network stack will recognize and manage 802.1Q VLAN tagged Ethernet frames. Supported frame types are Untagged, Priority Tagged, and VLAN Tagged. Traffic can be filtered, routed, and prioritized based on statically configured Layer 2 VLAN characteristics.

Fast Packet Forwarding, Cavium OCTEON

For Cavium OCTEON 38xx and 58xx, Wind River Network Stack 6.7 includes an option to dedicate one or more processor cores to fast-forwarding of IPv4 or IPv6 packets. At 500MHz, a single core can forward up to 862Mbps of network traffic, which exceeds the wire speed for Gigabit Ethernet. A single forwarding core can be used to process packets from all Gigabit Ethernet interfaces, or multiple forwarders can be configured to work together.

MPLS Data Plane Support*

Multiprotocol label switching (MPLS) brings the speed of layer 2 switching to layer 3. It allows routers to eliminate the complex lookup process based on the destination IP address and to make forwarding decisions based on the contents of a simple label. In addition to speed, MPLS offers two key advantages: It supports quality of service (QoS) and virtual private networks (VPNs).

The MPLS data plane support provides the following benefits:

- MPLS over Ethernet devices
- Generic MPLS labels
- IPv4 and IPv6 over MPLS
- Label stacking
- Common label operations
- Delivery of PDUs to Network Stack
- Ingress, transit, or egress node

QoS and DiffServ*

Differentiated services (DiffServ) is a standards-based priority mechanism for organizing different QoS categories. Network Stack includes DiffServ for egress traffic as a built-in feature of the dual-mode IPv4/IPv6 stack. It supports the following Internet Engineering Task Force (IETF) standards:

- **RFC 2474:** Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers
- **RFC 2475:** An architecture for differentiated service
- **RFC 2597:** Assured forwarding PHB group
- **RFC 2598:** An expedited forwarding PHB
- **RFC 2698:** A two-rate, three-color marker

Policy-Based Routing

Policy-based routing in Wind River Network Stack makes it possible to base the route lookup decision on more than just the destination address. With policy-based routing enabled, each virtual router (stack) can look up the destination in one or more forwarding information bases (FIBs). The virtual router looks up the packet information in a policy database to determine which FIB to use. If no matching entry is found, the default FIB is used.

Equal Cost Multipath (ECMP) Routing

Implementing ECMP routing enables network load-sharing by using multiple paths to the same destination. Wind River Network Stack includes two different algorithms, both defined in RFC 2991. Multipath issues unicast and multicast next-hop selections that can be used to select which routes to use when two or more routes have an equal cost. ECMP routing also means that if a certain network path becomes unavailable, the flow of packets will be switched rapidly to an alternative path.

Virtual Router Redundancy Protocol (VRRP)*

Using the Virtual Router Redundancy Protocol (VRRP) implementation in Wind River Network Stack allows a higher availability default path without the configuration of dynamic routing or router discovery protocols on every end host. You can configure VRRP to provide redundancy and load-sharing by defining multiple routers as default gateways and sharing traffic between them. VRRP support is currently defined for IPv4 only.

GRE and IP/IP Tunneling*

- Generic routing encapsulation (GRE) to tunnel IPv4 or IPv6 packets over IPv4

- Generic tunnel interface (GIF) to tunnel IPv4 and/or IPv6 packets in a point-to-point tunnel between two IPv4 hosts or IPv6 hosts

Robust Header Compression (ROHC)

ROHC compresses packet headers transmitted over a wireless link between a mobile terminal and a base station, thus reducing the overhead required for packet headers. The mobile terminal compresses the packet headers before transmitting them over the wireless link, and the base station decompresses them on receipt. Similar processing occurs in the reverse direction. Wind River Network Stack currently supports the uncompressed profile (profile 0x0000) and the IP compression profile (profile 0x0004). ROHC is required for WiMAX and other 4G and long-term evolution (LTE) networks. ROHC is integrated with MIP4 and MIP6.

DHCP

A number of enhancements have been made to DHCP. These include the following:

- DHCP Client now provides notice of lease expiration and success or failure of negotiation to acquire an IP address.
- DHCP Client now sends out a DHCPRELEASE message when DHCP autoconfiguration has been disabled.
- DHCP Client now identifies the responding interface in its callback routine. If a system is running a DHCP negotiation over two or more interfaces simultaneously and an offer is received on one of these interfaces, the callback routine now identifies the interface on which the offer is received.
- DHCP Client now specifies options in discovery messages sent to the DHCP server. Options include 12 (host name), 60 (class ID), and 124 (vendor class ID).
- DHCP Client can now run a negotiation without actually configuring the interface with the acquired addresses. This feature means a client can acquire dynamic addresses from the DHCP server without deleting its static IP addresses.
- DHCP Client can now act as a plain BOOTP client. In this scenario a system does not add a DHCP message type option to boot requests and does not manage the lease.

*These components are included in the industry-specific platforms only.

Custom Address Resolution Routine

Wind River Network Stack now supports custom (non-Ethernet) address resolution routines. This functionality lets you register these address resolution routines from an END driver with the network stack multiplexer, or MUX. The network stack then calls these routines before sending a packet to the END driver.

Packet Pool Configuration System Variable

Wind River Network Stack now supports a system variable for packet pool configuration. Instead of manually configuring the packet pool for the number and size of the packets expected, users now specify a maximum amount of memory, which the network stack is allowed to use to create the packet pool layout. The layout of the packet pool then changes dynamically to adapt to the current workload.

IFLib and Route APIs

Wind River Network Stack now includes support for the ifLib and if6Lib interface configuration libraries. These libraries enable interface configuration and routing routines that were available in the Network Stack before release 6.5.

IPv6 Scoped Address Architecture

You can now set the IPv6 address scope and zone for a node. You can either assign the default zone for a scope, such as a node, link, subnet, admin, site, or organization, or you can assign a particular interface to a zone.

Gratuitous ARP

Wind River Network Stack now supports gratuitous ARP. When gratuitous ARP is enabled, the system sends an ARP request for every new address added to a network interface. The ARP request is sent to verify that no other node on the link uses the address.

Cryptographically Generated Addresses

You can now generate and validate cryptographically generated addresses (CGAs) and generate CGA parameters. CGAs are IPv6 addresses for which the interface identifier is generated by

computing a cryptographic one-way hash function from a public key and auxiliary parameters. Systems can verify the binding between the public key and the address by recomputing the hash value and by comparing that hash value with the interface identifier.

Secure Neighbor Discovery

Wind River Network Stack now supports the Secure Neighbor Discovery (SEND) protocol. This protocol is used in IPv6 networks to counter threats to the Neighbor Discovery Protocol (NDP) that can occur in environments where physical security on the link is not ensured (such as over wireless).

Netlink

Wind River Network Stack now supports netlink and RTNETLINK. Netlink is a network service that enables communication between the kernel and user-space processes. Netlink enables a bidirectional communication link between the kernel and user-space processes so that either a user-space process or the kernel module can initiate communication.

RTNETLINK is a protocol extension to netlink that implements management and information retrieval for links, addresses, rules, routes, and neighbors.

MIB-II Support

Wind River Network Stack now supports MIB-II (RFC 1213), using the M2 library m2Lib. Applications developed with this library can allow an SNMP agent to access MIB information, bypass the SNMP agent to access MIB information directly from a device, and access hardware statistics counters.

Routing Engine

Wind River Network Stack contains a high-performance routing engine that uses highly optimized Radix trees that allow both static and dynamic routes. A standard BSD routing socket interface enables the use of standard routing daemons and also allows for dedicated routing devices to cooperate with the TCP/IP stack.

Virtual Routing

Wind River Network Stack supports full virtualization with multiple independent routing tables used in virtual routers. This means that one network stack can act as multiple routers, enabling a massive reduction in router hardware. The virtual routing support includes a number of BSD socket extensions to manage the additional routing tables.

IPv4 and IPv6 Autoconfiguration

Wind River Network Stack supports both IPv4 and IPv6 autoconfiguration.

Wind River Mobile IPv4/IPv6*

Mobile IP provides an efficient, scalable mechanism that allows users to roam seamlessly among wireless networks. Using Mobile IP, applications such as Internet telephony, media streaming, and virtual private networking can be supported without service interruption when users move across network boundaries.

Wind River Mobile IP products are compliant with WiMAX Forum standards for mobility in WiMAX networks, as well as the 3GPP2/3GPP standards for next-generation cellular networks, including LTE. In addition, these products implement fast handover ("make-before-break") for the support of Voice over Internet Protocol (VoIP) and other real-time applications.

Wind River Network Stack includes support for the following:

- Mobile IPv4 mobile node
- Mobile IPv4 home agent
- Mobile IPv4 foreign agent
- Mobile IPv6 mobile node
- RADIUS Key Exchange and accounting
- Mobility addressing extensions
- MOBIKE (IKEv2)
- NEMO, which extends the Mobile IPv6 node to be able to register network prefixes and provide forwarding functionality and roaming for network entities using said prefixes
- Extended Mobile IPv6 node to attach the mobile node identifier to its registration requests
- Support for the network access identifier, which provides the user identity of the connecting node to other mobility resources that the mobile node contacts

*These components are included in the industry-specific platforms only.

Wind River PPP

Wind River PPP is a source-code product written in C with an object-oriented design. With Wind River PPP, you can implement a dynamically configured PPP stack for diverse remote access applications. These applications can have a variety of network interface types, network stack types, link speeds, control protocols, framing techniques, and so on.

Features of Wind River PPP include the following:

- Complete RFC 2516 implementation (server mode)
- MD5 integrity protection against denial-of-service attacks
- Highly portable function API

Wind River USB

Wind River USB enables developers to quickly incorporate standard universal serial bus (USB) connectivity in VxWorks-based embedded devices and their attached peripherals.

Wind River USB 2.4 supports version 2.0 of the USB specification, providing support for the universal host controller interface (UHCI), the open host controller interface (OHCI), and the enhanced host controller interface (EHCI).

Starting in VxWorks 6.6, Wind River USB has been integrated under the VxBus device driver framework. With this release of VxWorks, the USB host controller drivers are fully integrated with VxBus.

VxWorks 6.7 Multiprocessing

VxWorks 6.7 Multiprocessing is an add-on product to all VxWorks 6.7-based

platforms. It is an evolution of the VxWorks RTOS to provide both SMP and AMP capabilities, not a new operating system. VxWorks SMP leverages multicore processors to achieve true concurrent execution of applications, allowing applications to gain in performance through parallelism. VxWorks AMP technologies allow users to construct systems running multiple operating systems, handling different tasks and communicating within themselves. Users may purchase the VxWorks 6.7 Multiprocessing add-on either with the base VxWorks 6.7 platform or any time after. Once installed, users can develop on both VxWorks uniprocessor (UP) and SMP and use the AMP technologies according to their project needs.

VxWorks 6.7 SMP

Multiprocessing systems include two or more processors in a single system. SMP is a variant of multiprocessing technology in which one instance of an operating system controls all processors and in which memory is shared. SMP differs from AMP in that an AMP system has a separate instance of an operating system executing on each processor (and each instance may or may not be the same type of operating system).

The VxWorks SMP configuration allows a single instance of the operating system to use multiple processors in a single system. At the same time, it maintains the same key RTOS characteristics of performance, small footprint, high reliability, and determinism as the uniprocessor configuration of VxWorks.

Terminology

The terms *CPU* and *processor* are often used interchangeably in computer literature. However, it is useful to distinguish between the two for hardware that supports SMP. In this document, particularly in the context of VxWorks SMP, the terms are used as follows:

- **CPU:** A single processing entity capable of executing program instructions and processing data (also referred to as a *core*, as in *multicore*)
- **Processor:** A silicon unit that contains one CPU
- **Multiprocessor:** A single hardware system with two or more processors
- **Uniprocessor (UP):** A silicon unit that contains a single CPU
- **Multicore processor:** A silicon unit that contains two or more CPUs

For example, a dual-core processor has two CPUs. A quad-core processor has four CPUs.

UP code may not always execute properly on an SMP system, and code that has been adapted to execute properly on an SMP system may still not make optimal use of SMP. The following terms are therefore used to clarify the state of code in relation to SMP:

- **SMP-ready:** Runs correctly on an SMP operating system, although it may not make use of more than one CPU (i.e., does not take full advantage of concurrent execution for better performance)
- **SMP-optimized:** Runs correctly on an SMP operating system, uses more than one CPU, and takes sufficient advantage of multitasking and concurrent execution to provide performance gains over a uniprocessor implementation

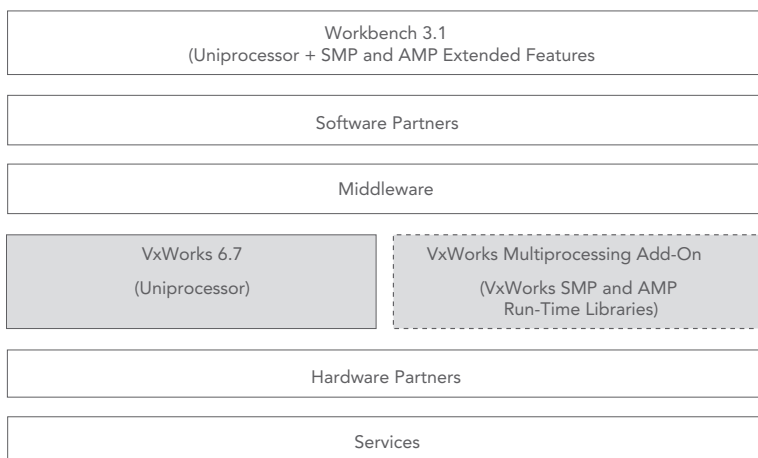


Figure 5: VxWorks Multiprocessing installation with VxWorks platforms

The VxWorks SMP-enabled platforms allow you to do the following:

- Deliver higher-performance multicore-powered products with reduced risk and development investment
- Speed time-to-market by using the commercially available and supported run-time platforms and developer tools for multiprocessing
- Increase productivity by using the same development environment and process available for uniprocessor development by using the SMP extension to existing development tools
- Enable a flexible design approach using Wind River's expertise and broad technology support for multiprocessing

VxWorks SMP Operating System

Features

VxWorks SMP is designed for symmetric target hardware. That is, each CPU has equivalent access to all memory and all devices. VxWorks SMP can therefore run on targets with multiple single-core processors or with multicore processors, as long as they provide a uniform memory access (UMA) architecture with hardware-managed cache coherency.

Multitasking

SMP changes the conventional UP paradigm of priority-based preemptive multitasking programming because it allows true concurrent execution of tasks and handling of interrupts. This is possible because multiple tasks can run on multiple CPUs, while being controlled by a single instance of an operating system.

Scheduling

VxWorks SMP provides a priority-based preemptive scheduler, such as VxWorks UP. In both VxWorks UP and VxWorks SMP, tasks are scheduled—and RTPs are not. However, the VxWorks SMP scheduler is different from the UP scheduler in that it also manages the concurrent execution of tasks on different CPUs.

Mutual Exclusion

Because SMP systems allow for truly concurrent execution, the UP mechanisms for disabling (masking) interrupts

and for suspending task preemption in order to protect critical regions are inappropriate for—and not available in—an SMP operating system. Enforcing interrupt masking or suspending task preemption across all CPUs would defeat the advantages of truly concurrent execution and drag multiprocessing performance down toward the level of a UP system.

VxWorks SMP therefore provides specialized mechanisms for mutual exclusion between tasks and interrupts executing and being received (respectively) simultaneously on different CPUs. In place of UP task and interrupt locking routines—such as `taskLock()` and `intLock()`—VxWorks SMP provides spinlocks, atomic memory operations, and CPU-specific mutual exclusion facilities.

CPU Affinity

By default, any task can run on any of the CPUs in the system (which generally provides the best load balancing), and interrupts are routed to CPU 0 (the bootstrap CPU). There are instances, however, in which it is useful to assign specific tasks or interrupts to a specific CPU. VxWorks SMP provides this capability, which is referred to as CPU affinity.

VxWorks SMP Hardware

The hardware required for use with VxWorks SMP must consist of symmetric multiprocessors—either multicore processors or hardware systems with multiple single CPUs. The processors must be identical; all memory must be shared between the CPUs (none may be local to a CPU); and all devices must be equally accessible from all CPUs—that is, targets for VxWorks SMP must adhere to the UMA architecture.

Regardless of the number of CPUs (typically two, four, or eight) in an SMP system, the important characteristics are the same:

- Each CPU accesses the very same physical memory subsystem; there is no memory local to a CPU. This means it is irrelevant which CPU executes code.

- Each CPU has its own memory management unit that allows concurrent execution of tasks with different virtual memory contexts. For example, CPU 0 can execute a task in RTP 1 while CPU 1 executes a task in RTP 2.
- Each CPU has access to all devices. Interrupts from these devices can be routed to any one of the CPUs through a programmable interrupt controller. This means that it is irrelevant which CPU executes ISRs when handling interrupts.
- Tasks and ISRs can be synchronized across CPUs and mutual exclusion enforced by using spinlocks.
- Bus snooping logic ensures the data caches between CPUs are always coherent. This means that the operating system does not normally need to perform special data cache operations in order to maintain coherent caches. However, this implies that only memory access attributes that allow bus snooping are used in the system.

Comparison of VxWorks SMP and AMP

The features of VxWorks SMP may be highlighted by comparison with the way VxWorks is used in AMP, using the same target hardware in both cases.

In an SMP configuration, the entire physical memory space is shared between the CPUs. This memory space is used to store a single VxWorks SMP image (text, data, bss, heap). It is also used to store any RTPs that are created during the lifetime of the system. Because both CPUs can potentially read from, write to, and execute any memory location, any kernel task or user (RTP) task can be executed by either CPU.

In an AMP configuration there is one copy of the VxWorks image in memory for each CPU. Each operating system image can only be accessed by the CPU to which it belongs. It is therefore impossible for CPU 1 to execute kernel tasks residing in VxWorks CPU 0's memory, or the reverse. The same situation applies for RTPs. An RTP can only be accessed and executed by the instance of VxWorks from which it was started.

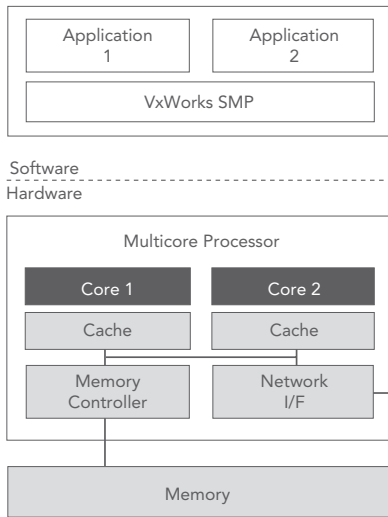


Figure 6

Comparison of the relationship between CPUs and memory use in SMP and AMP systems

In an AMP system, some memory is shared, but typically the sharing is restricted to reading and writing data—for example, for passing messages between two instances of VxWorks. Hardware resources are mostly divided between instances of the operating system so that coordination between CPUs is only required when accessing shared memory.

SMP and Mutual Exclusion

The use of mutual exclusion facilities is one of the critical differences between UP and SMP programming. While some facilities are the same for VxWorks UP and VxWorks SMP, others are necessarily different. In addition, reliance on implicit synchronization techniques—such as relying on task priority instead of explicit locking—do not work in an SMP system.

Unlike UP systems, SMP systems allow for truly concurrent execution, in which multiple tasks may execute and multiple interrupts may be received and serviced, all at the same time. In most cases, the same mechanisms—semaphores, message queues, and so on—can be used in both UP and SMP systems for mutual exclusion and coordination of tasks.

The most basic differences for SMP programming have to do with the mechanisms available for mutual exclusion between tasks and interrupts

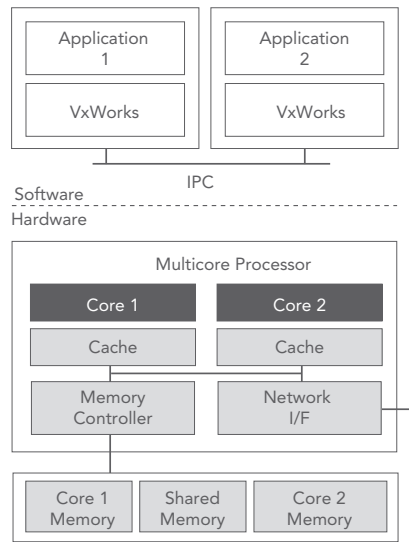


Figure 7

executing and being received (respectively) on different CPUs. In place of UP task and interrupt locking routines—such as `taskLock()` and `intLock()`—VxWorks SMP provides the following facilities:

- Spinlocks for tasks and ISRs
- CPU-specific mutual exclusion for tasks and ISRs
- Atomic memory operations
- Memory barriers

As with the UP mechanisms used for protecting critical regions, spinlocks and CPU-specific mutual exclusion facilities should only be used when they are guaranteed to be in effect for very short periods of time. The appropriate use of these facilities is critical to making an application SMP-ready.

Note that both spinlocks and semaphores provide full memory barriers (in addition to the memory barrier macros themselves).

RTP Applications

As in VxWorks UP systems, RTP (user-mode) applications have a more limited set of mutual exclusion and synchronization mechanisms available to them than kernel code or kernel applications. In VxWorks SMP, they can make use of semaphores and atomic operations, but not spinlocks, memory barriers, or CPU-specific mutual exclusion mechanisms.

Spinlocks for Mutual Exclusion and Synchronization

Spinlocks provide a facility for short-term mutual exclusion and synchronization in an SMP system. Spinlocks must be explicitly taken and released. While semaphores can also be used for mutual exclusion and synchronization, spinlocks are designed for use in situations comparable to those in which `taskLock()` and `intLock()` are used in VxWorks UP. Semaphores should be used in an SMP system for the same purposes as in a UP system. (Note that both spinlocks and semaphores provide full memory barriers.)

One of the unique characteristics of VxWorks spinlocks is that they are implemented with algorithms that ensure that they are fair, meaning that they are deterministic in the time between the request and take, and they operate in as close to FIFO order as possible.

Note: Spinlocks are not available to RTP (user-mode) applications.

Spinlocks as Full Memory Barriers

VxWorks spinlocks operate as full memory barriers between acquisition and release. A full memory barrier forces both read and write memory access operations to be performed in strict order. The process of updating data structures is therefore fully completed between the time a spinlock is acquired and released.

Types of Spinlocks

- **ISR-callable spinlocks:** Used to address contention between ISRs—or between a task and other tasks and ISRs. They disable (mask) interrupts on the local CPU. When called by tasks, they suspend task preemption on the local CPU as well.
- **Task-only spinlocks:** Used to address contention between tasks (and not ISRs). They suspend task preemption on the local CPU. The *local CPU* is the one on which the spinlock call is performed.

Spinlock Behavior and Usage Guidelines

Unlike the behavior associated with semaphores, a task that attempts to take a spinlock that is already held by another task does not get suspended. Instead it continues executing, simply spinning in a tight loop waiting for the spinlock to be freed.

The terms *spinning* and *busy waiting*—which are both used to describe this activity—provide insight into both the advantages and disadvantages of spinlocks. Because a task (or ISR) continues execution while attempting to take a spinlock, the overhead of rescheduling and context switching can be avoided (which is not the case with a semaphore). On the other hand, spinning does no useful work and ties up one or more of the CPUs. Spinlocks should therefore only be used when they are likely to be efficient—that is, when they are going to be held for very short periods of time (as with `taskLock()` and `intLock()` in a uniprocessor system). If a spinlock is held for a long period of time, the drawbacks are similar to `intLock()` and `taskLock()` being held for a long time in VxWorks UP: increased interrupt and task latency.

Acquisition of a spinlock on one CPU does not affect the processing of interrupts or scheduling of tasks on other CPUs. Tasks cannot be deleted while they hold a spinlock.

Task-Only Spinlocks

Spinlocks that are used to address contention between tasks alone (and not ISRs) are called *task-only spinlocks*. These spinlocks disable task preemption on the local CPU while the caller holds the lock (which could otherwise lead to a livelock situation). This prevents the caller from being preempted by other tasks and allows it to execute the critical section that the lock is protecting. Interrupts are not disabled and task preemption on other CPUs is not affected. For VxWorks UP, task-only spinlocks are implemented with the same behavior as the task locking routines `taskLock()` and `taskUnlock()`.

CPU-Specific Mutual Exclusion

VxWorks SMP provides facilities for CPU-specific mutual exclusion, which are for mutual exclusion operations whose scope is entirely restricted to the CPU on which the call is made (the *local CPU*). These facilities are designed to facilitate porting uniprocessor code to an SMP system.

CPU-Specific Mutual Exclusion for Interrupts

CPU-specific mutual exclusion for interrupts allows for disabling (masking) interrupts on the CPU on which the calling task or ISR is running. For example if task A, running on CPU 0, performs a local CPU interrupt lock operation, no interrupts can be processed by CPU 0 until the lock is released by task A. Execution of interrupts on other CPUs in the SMP system is not affected. In order to be an effective means of mutual exclusion, all tasks and ISRs that should participate in the mutual exclusion scenario should have CPU affinity set for the local.

Note that some routines should not be used if the calling task or ISR has locked interrupts on the local CPU—similar to the case of holding spinlocks.

CPU-Specific Mutual Exclusion for Tasks

CPU-specific mutual exclusion for tasks allows for suspending task preemption on the CPU on which the calling task is running. That is, it provides for local CPU task locking and effectively prevents any other task from running on the local CPU. For example, task A running on CPU 0 can perform a local CPU task lock operation so that no other task can run on CPU 0 until it releases the lock or makes a blocking call. The calling task is also prevented from migrating to another CPU until the lock is released. Execution on other CPUs in the SMP system is not affected. In order to be an effective means of mutual exclusion, all tasks that should participate in the mutual exclusion scenario should have CPU affinity set for the local CPU.

Memory Barriers

A memory barrier is a class of instructions that cause a CPU to enforce ordering of memory operations. CPUs use optimizations that can cause out-of-order execution, which causes unpredictable behavior in concurrent programs and device drivers in SMP systems. If code depends on the order in which another CPU writes to memory, a memory barrier should be used. Note that both semaphores and spinlocks provide full memory barriers on their own.

Atomic Memory Operations

Atomic operations make use of CPU support for atomically accessing memory. They combine a set of (architecture-specific) operations into what is effectively a single operation that cannot be interrupted by any other operation on the memory location in question. Atomic operations thereby provide mutual exclusion for a simple set of operations (such as incrementing and decrementing variables). Atomic operations can be useful as a simpler alternative to spinlocks, such as for updating a single data element. For example, you can update the *next* pointer in a singly linked list from NULL to non-NULL (without interrupts locked) using an atomic operation, which allows you to create lock-less algorithms. Because the atomic operations are performed on a memory location supplied by the caller, users must ensure the location has memory access attributes and an alignment that allows atomic memory access—otherwise an access exception will occur. Restrictions, if any, are specific to the CPU architecture.

Task CPU Affinity

VxWorks SMP provides the ability to assign tasks to a specific CPU, after which the scheduler ensures the tasks are only executed on that CPU. This assignment is referred to as *task CPU affinity*.

While the default SMP operation in which any task can run on any CPU often provides the best overall load balancing, there are cases in which assigning a specific set of tasks to a specific CPU can

be useful. For example, if a CPU is dedicated to signal processing and does no other work, the cache remains filled with the code and data required for that activity. This saves the cost of moving to another CPU—which is incurred even within a single piece of silicon, as the L1 cache is bound to a single CPU, and the L1 must be refilled with new text and data if the task migrates to a different CPU.

Another example is a case in which profiling an application reveals that some of its tasks are frequently contending for the same spinlock and a fair amount of execution time is wasted waiting for a spinlock to become available. Overall performance could be improved by setting task CPU affinity such that all tasks involved in spinlock contention run on the same CPU. This would free up more time on other CPUs for other tasks.

RTP Tasks and CPU Affinity

By default, RTP tasks inherit the CPU affinity setting of the task that created the RTP. If the parent task has no specific CPU affinity (i.e., it can execute on any available CPU and may migrate from one CPU to the other during its lifetime), the RTP's tasks have no specific CPU affinity either. If the parent task has its affinity set to a given CPU, by default the RTP tasks inherit this affinity and execute only on the same CPU as the parent task.

Interrupt CPU Affinity

SMP hardware requires programmable interrupt controller devices. VxWorks SMP makes use of this hardware to allow assignment interrupts to a specific CPU. By default, interrupts are routed to the bootstrap CPU (CPU 0).

Interrupt CPU affinity can be useful for load balancing (e.g., if there is too much total interrupt traffic for one CPU to handle). It can also be used as an aid in migrating code from VxWorks UP.

Run-time assignment of interrupts to a specific CPU occurs at boot time, when the system reads interrupt configuration information from the BSP. The interrupt controller then receives a command directing that a given interrupt be routed to a specific CPU.

Developer Tools for Debugging SMP Code

A multiprocessing platform is of little value unless the software applications that will run on it make effective use of multiprocessing. The Wind River Workbench development environment is based on the Eclipse framework and integrates a broad range of capabilities. The Workbench development environment and associated tools offer extensive capabilities to meet the challenges of multiprocessing device software development.

Development Challenges and Solutions

Multiprocessing-Enabled Tools

A multiprocessing situation requires the ability to physically control multiple processors or processor cores. The tools enabled for this task are included in the Wind River Workbench development environment, which is included in the platform, and the Wind River on-chip debugging tools, which are an optional add-on product. The multiprocessing capabilities of the various components are detailed here.

Wind River Workbench includes a target manager to coordinate multiple processor systems. In order to be useful, debuggers and other developer tools must be able to connect to a processor of interest. In many multiple processor systems, there may be processors that do not sit on the same network as the developer's desktop computer. In cases where a processor of interest during debug is not on the same network as the developer's desktop but can be reached through an intermediate processor that does share the network, the proxy included in the Workbench Debugger can make the connection—and even translate the protocol if needed.

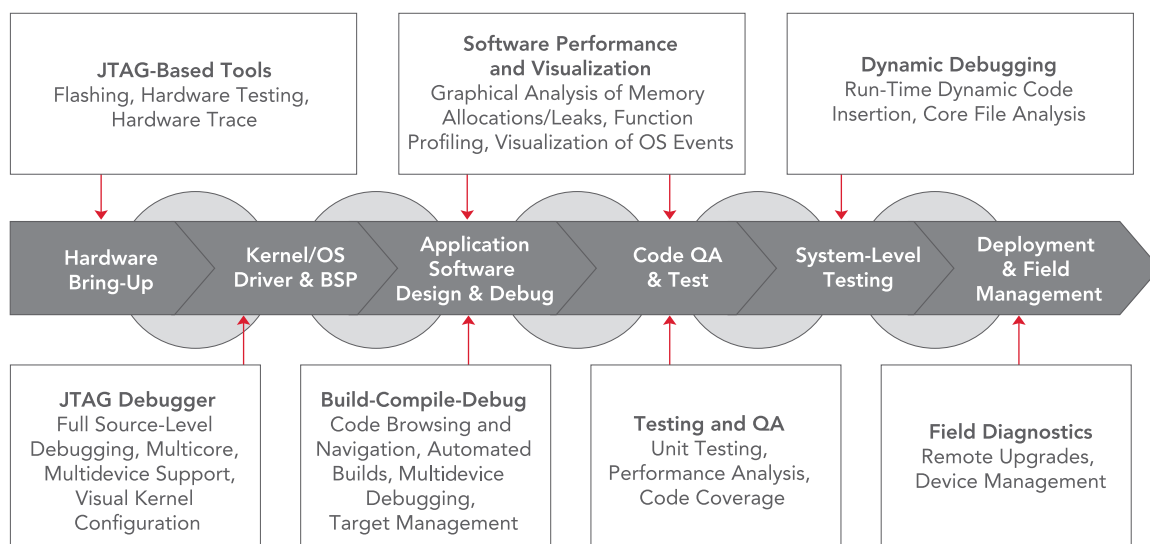


Figure 8: Wind River Workbench development environment integrates a broad range of capabilities

When using on-chip debugging, a single Wind River ICE can be used to control multiple processors or processor cores. Many other on-chip debugging solutions require a separate debugging hardware unit for every single processor or core, significantly increasing cost and complexity. Wind River's support for multi-processor on-chip debugging also includes synchronized run-control across multiple processors when they are supported by the capabilities of the underlying hardware.

Identifying Overused and Underused Processors

New multiprocessing-specific enhancements have been included in Wind River System Viewer. For example, System Viewer can display the utilization or number of pending tasks for each processor core over time. For SMP systems, the data for every processor core will be integrated in a single, time-correlated display. In addition, the VxWorks commands can be used to determine the amount of CPU time used by each task, the amount of time spent at interrupt level and in the kernel, and the amount of time the CPU was idle.

Identifying and Diagnosing Race Conditions

Wind River System Viewer supports the visualization of user-defined events. Developers can identify event sequences that represent possible race conditions by logging appropriate events to a file then processing the data in that file with a script or program.

Debugging Thread Interaction Across Multiple Processors

The Workbench Debugger works simultaneously on multiple contexts. It assigns each thread its own debug context that includes a source/assembly view, a register window, and a stack back-trace view. Each thread can also be controlled independently using thread-qualified breakpoints that further extend the developers' ability to debug multiple processor interaction.

Debugging Multiple Processors Without Breakpoints

In AMP systems, it may not be possible to debug certain problems using breakpoints. Stopping one of several processors can cause the system to malfunction or change the system's behavior so much that it hides the problem that's being sought. Wind River Lab Diagnostics enables dynamic interaction with a running application—adding diagnostic code, collecting data, and modifying program execution. Providing these capabilities without requiring the processor to be stopped or the application code to be rebuilt makes Lab Diagnostics a much less intrusive solution for an AMP system than a traditional debugger.

Tracking Data Values Systemwide

Understanding the operation of a multiple processor system often depends on understanding the changes in the state of the system created by the interaction of multiple processors and having the ability to track data values over time. Wind River Data Monitor (formerly StethoScope) is an easy way to track key data values in an SMP system, or the value of data shared between multiple processors in an AMP system. Workbench target agent and on-chip debugging support for processors with hardware breakpoint capability provide another way to track data values, by setting breakpoints that trigger on data accesses at a specified address.

Identifying the Best Functions to Speed Up

To increase performance by applying parallelism, begin with the parts of your program that consume the most CPU cycles. The Wind River development environment for Linux and VxWorks provides several ways to collect this information. The VxWorks Spy() capability, and the top command available in a Linux environment, provide one alternative. Wind River Performance Profiler (formerly ProfileScope) and System Viewer make it easy to identify which functions in an application are consuming the most CPU cycles. In an SMP

system, additional threading will help to increase system performance. In an AMP system, these are the functions that may benefit from decomposition or reassignment to a processor with a lighter load.

Uncovering Shared Resource Contention

The rich data collection capabilities of Wind River System Viewer can be used to log information that can later be analyzed to expose a range of resource contention-related issues.

Tracking Interprocessor Communication and Synchronization

In multiprocessor systems, the interaction of processors is at least as important as what happens on any single processor. Spinlocks are commonly used to provide synchronization in multiprocessing systems. Spinlocks are supported with VxWorks SMP, and spinlock activity is trackable in System Viewer visualization. System Viewer instrumentation of TIPC messages also provides insight into the interaction between processors.

Displaying Static or Dynamic Routing of Interrupts to Processors

The assignment of hardware resources, including interrupts, to specific processors is an important multiprocessing design decision. The ability to query the state of VxWorks kernel objects allows developers to see how interrupts are assigned to processors. This will apply to static and dynamic assignment.

VxWorks AMP Technology Overview

VxWorks 6.7 introduces brand-new technologies aimed at facilitating the development of AMP technologies on multicore hardware. The ultimate goal for the Wind River AMP technologies is to facilitate the construction of multi-OS systems involving more than one type of OS. These technologies have been designed to be portable across different operating systems and can be used with minor porting effort in other operating systems such as Wind River Linux. As of the release of VxWorks 6.7, the AMP technologies support the construction of AMP systems comprising two or more VxWorks 6.7 instances.

Terminology: Nodes and CPUs

In this document, the term node applies to an instance of an operating system on one or more cores and does not necessarily correspond to a CPU itself. A node is a single processor computer in its own right, yet only part of a larger system comprising more than one node.

The simplest example of this is when an operating system runs on one core, that core with its OS is called a *node* in an AMP system. If VxWorks SMP is running on four CPUs (CPU 0 through CPU 3), the four CPUs constitute a single node. In this case, if CPU 4 is used for AMP, it counts as node 2. (For information on SMP, see *VxWorks Kernel Programmer's Guide: VxWorks SMP*.) In a multicore context in which all CPUs have a single VxWorks image without SMP, node and CPU are equivalent.

Internode Communications Using MIPC

MIPC is a message-passing protocol designed for internode communications in AMP systems. MIPC uses shared memory as a fast, efficient communications medium between OSes running on different cores. Messages are passed between nodes without copying (i.e., zero-copy). The network topology is statically defined at configuration time. Programmatically, a familiar socketlike interface allows TCP/IP- or TIPC-based software to be ported easily and then run at hardware speeds.

Shared Memory and Hardware Partitioning

Shared memory between nodes is the most commonly found medium by which nodes in an AMP system may share data and pass messages. In a multicore system, the total physical memory in the system must be split into dedicated memory regions for each node and its OS and shared memory regions used for internode communications. Devices available in the system must also be cleanly divided between the different nodes that control them. This system

level hardware configuration is different for each system and must be carefully done to avoid conflicts.

Booting AMP Systems: the wrload Utility

AMP systems need to boot nodes in a controlled order. This is controlled by the wrload utility. Boot images in the form of fully linked ELF files can be loaded by wrload in a configurable sequence. One CPU with VxWorks—the primary CPU—boots first and then uses the wrload utility to load and boot VxWorks on the other CPUs—the secondary CPUs. The paths of boot images are passed to boot images, along with other boot options—for example, the VxWorks boot line. wrload can be invoked either programmatically, such as from a script, or via the VxWorks kernel shell. Multiple invocations of wrload are used to define the boot order of nodes in an AMP system.

Shared Devices, Virtual Consoles

AMP systems using multicore CPUs often have fewer physical peripheral devices than the number of nodes present in the system. A case in point is the console serial port, of which there is usually only one or a couple. It is thus necessary to set up the sharing of one physical console port across more than one node. This allows resource-constrained systems with a limited number of physical devices to be shared among more than one node in an AMP system. Nodes that do not have access to a physical serial port use a virtual serial-port driver that uses MIPC, called an MIPC serial device (MSD), to communicate to a node that does. The messages are output to the physical console and can be prefixed with a configurable tag to identify the node they originated from. Alternately, if the terminal supports it, a different color can be assigned to the messages for easy identification.

The VxWorks tip utility provides full duplex communication between a host and the nodes on a multicore target system. The VxWorks tip utility resembles the tip utility for UNIX and the Linux cu utility. Using VxWorks tip, commands can be sent to individual nodes in a multicore system.

AMP Debugging

AMP system debugging relies on a combination of local debug agent in each node and a “proxy” agent in one of the nodes (usually Core 0 of a multicore processor) to route debug commands and information back to the debugger running on the host machine. This allows the debugger to connect to use one host-target network link to debug either individual nodes or the whole system. Wind River ICE can also be used to debug supported processors running an AMP system. The debug operations themselves, such as step, continue, step-over, and so on, are the same ones used in a single-OS environment.

Legacy AMP Systems Using VxMP

VxWorks 6.7 continues to support VxMP as a means of constructing AMP systems. VxMP and the new AMP technologies can coexist in the same system. However, the newer technologies are both more efficient for multicore systems and more robust. In contrast, VxMP was designed to work over a shared memory backplane, such as on VME busses, and so forth, which are less efficient than shared memory on multicore systems. A single node that suffers failure does not need the whole system to be rebooted, as was the case with VxMP. It is encouraged for users to adopt the newer AMP technologies as soon as convenient.

Programming AMP Systems

Programming AMP systems is fundamentally no different than programming a set of single-OS systems. Each node is a single-OS programming environment in its own right. Except where specifically noted, all software facilities, features, and restrictions for the single-OS or SMP configurations of VxWorks are applicable to nodes of an AMP system.

Users should note, however, that there is still true concurrency of different actions happening in an AMP system, though the concurrent execution is happening in other OSes. Therefore, AMP programs should not assume any specific order of execution or timing characteristics of actions happening on other nodes. Access to data items shared between

nodes should either use lock-free algorithms or block on receipt of a message as a synchronization mechanism before accessing a shared data item.

Middleware Technology

Security

Wind River IPsec and IKE

Wind River IPsec is a scalable implementation of IPsec, as specified by the IETF. It provides authentication, data integrity, encryption, and replay protection of any network traffic on the IP layer. It is implemented as a tightly integrated software module for Wind River Network Stack, for both IPv4 and IPv6 operations. Wind River IPsec is interoperable with other IPsec implementations and conforms to the IPsec RFCs, as specified by the IETF.

Features of Wind River IPsec 6.7 include the following:

- Tunnel and transport mode in any security association (SA) combination
- Support for AH and ESP modes
- IP in IP tunneling
- Bypass/apply/discard IP packet filtering with both input and output selectors
- Support for IPsec monitoring MIB
- Key and SA management with PF_Key Management API v2 with openbsd extension
- Support for all required authentication transforms and encryption algorithms
- Validated with Common Cryptography Interface (CCI)
- Extended Sequence Number (ESN) Addendum to IPsec
- Interoperability tested with popular IPsec and IKE products
- Integrated and validated with optional security coprocessors, demonstrating significant performance improvement over software processing
- Virtual Private Network Consortium (VPNC) certification for basic and AES interoperability
- VPN-A and VPN-B suites
- Traffic flow confidentiality (TFC)
- IPsec offloading with SMP systems

Wind River IKE is a scalable implementation of IKE versions 1 and 2, as specified by the IETF, and it provides for secure key exchange for IPsec.

Features of Wind River IKE 6.7 include the following:

- Support for IKE v1 and IKE v2
- NAT traversal of ESP packets over UDP
- Integration with Wind River Network Stack
- Authentication based on X.509 certificates and preshared secrets (passwords)
- Passive and active establishment of IPsec connections
- Secure, interoperable communication with other IPsec endpoints
- Plug-and-play integration with Wind River IPsec
- Flexible and powerful policy-based configuration
- Support for tunnel and transport IPsec connections
- VPNC certification for IKE v2 interoperability
- Support for TFC
- Support for ESN anti-replay support for high-speed IPsec implementations

Wind River Wireless Security

Wind River Wireless Security is a suite of security protocols that includes supplicant and authenticator for the 802.1X protocol. The Wireless Security authenticator is integrated with the Wind River RADIUS and Diameter, Wind River Learning Bridge, and Wind River Wireless Ethernet Driver, providing all the core functionality for typical authenticator products, such as wireless access points. Both supplicant and authenticator can be used in the same product, allowing greater flexibility and a range of application support. Wind River Wireless Security works together with Wind River EAP that supports multiple EAP types. Integration with Wind River SNMP is included to interface with the 802.1X MIB.

In Wind River Wireless Security 2.x, EAP was integrated with the supplicant. In Wind River Wireless Security 6.7, the supplicant operates with the IPCOM EAP library, where EAP is a separate module.

Features of Wind River Wireless Security 6.7 include the following:

- 802.1X
- Wi-Fi Protected Access (WPA and WPA2)
- 802.11i
- Temporal Key Integrity Protocol (TKIP)
- AES
- Preshared keys

- EAP types and EAP type combinations added with use of new independent EAP module (IPEAP) and allows EAP to be used by other Wind River modules
- Full integration and testing with Wind River Wireless Ethernet Driver (station and access point modes), easily portable to other wireless driver solutions
- Support for both authenticator and supplicant modes
- Support for wide range of encryption and hashing algorithms
- Support for dot1xPaeConfigGroup supplicant MIB
- New commands to support run-time configuration of multiple port access entity (PAE) instances

Wind River EAP

Extensible Authentication Protocol (EAP) allows a client and a server to negotiate an authentication method and establish client, and optionally server, authentication. The protocol defines a packet structure that contains commands and attributes used by the client and server during an authentication session. A number of EAP types and EAP type combinations are available to implement an authentication method.

EAP is a separate module based on the IPCOM EAP (IPEAP). EAP was previously integrated with the Wind River Wireless Security supplicant. The Wind River EAP associated with Wind River Wireless Security 2.4 is no longer supported. EAP is used by the Wind River Wireless Security supplicant to negotiate an authentication method.

The following are the EAP types supported in Wind River EAP 6.7:

- EAP-AKA
- EAP-GTC
- EAP-LEAP
- EAP-MD5
- EAP-MS-CHAPv2
- EAP-PEAPv0
- EAP-PEAPv1
- EAP-SIM
- EAP-TLS
- EAP-TTLS

Wind River Firewall and Wind River NAT

Wind River NAT is a full-featured implementation of the industry-standard Network Address Translation Protocol (NATP) for use in routers, firewalls, DSL

and cable modems, and residential gateways. A device running Wind River NAT can connect an entire department or a small office to the Internet using a single global IP address. Address mapping effectively conceals the size and topology of the private network from the outside, providing a basic level of security.

The chief advantage of NAT is that addresses on the private network are hidden from the public Internet, providing a measure of security. A second advantage, realized with certain varieties of NAT, is that scarce IP addresses are conserved, reducing network administration costs.

Wind River NAT supports the two most widely used NAT modes. Basic NAT performs one-to-one mapping of private IP addresses to a preallocated block of external IP addresses. The more commonly used NATP maps port numbers as well as IP addresses. NATP allows multiple private addresses (up to 64,000 address/port combinations) to be multiplexed on a single public address, offering the full benefit of address conservation and security.

NAT provides basic security by blocking all incoming connection requests that don't map to recognized address translations.

Wind River NAT supports the following features:

- Basic NAT
- Network Address Port Translation (NAPT)
- Bidirectional NAT
- Network Address Translation-Protocol Translation (NAT-PT)
- Demilitarized zone (DMZ) host
- Application-level gateways (ALGs)
- Port triggering
- Support for the Session Initiation Protocol (SIP) Application-Level Gateway (ALG), allowing simple SIP UDP sessions to pass through a NAT device

Wind River Firewall supplies a powerful filtering engine that allows device manufacturers to optimize their software to provide advanced features that protect valuable data. This engine is ideally suited to a wide range of products, including SOHO routers, broadband access devices, and small to medium-size enterprise devices.

Features of Wind River Firewall 6.7 include the following:

- IP filtering with stateful inspection for IPv4 or IPv6 packets
- MAC (media access control) filtering
- Logging at the network (L3) and data link (L2) layers
- HTTP content filtering for URLs (both specific and by keyword) and proxy traffic
- Java applets, ActiveX controls, and cookies
- Nonvolatile (NV) storage of firewall rules
- Input and output filters
- Stateful inspection
- Rate limiting
- Filter on network interface
- Support for the pktflags keyword that's used to filter IPsec, NAT, and tunneled packets.
- Rule grouping
- Support for stateful firewalling

Wind River Cryptography

Wind River Cryptography is a collection of cryptographic modules that can be used in developing secure applications. It consists of widely used algorithms based on OpenSSL 0.9.8h, optimized for embedded devices. It is also used by other components requiring access to crypto functions.

Wind River Cryptography includes an implementation of X.509 digital certificates. Digital certificates can be used by a wide variety of other components, such as Wind River IPsec and IKE, Wind River Wireless Security, Wind River Web Server, and Wind River Web Services. Wind River Cryptography uses the Cryptography Libraries API, which is defined in the *Wind River Cryptography Libraries for VxWorks 6 Programmer's Guide 6.7*. The Cryptography Libraries also include an implementation of hardware support for the Freescale SEC security processor, included in many Freescale PowerQuicc processors.

The following are included algorithms:

- AES
- Blowfish
- CAST
- DES
- RC2
- Arcfour
- RC5
- AES key wrap
- SHA-1

- MD2
- MD4
- MD5
- RIPEMD-160
- HMAC
- AES CMAC
- Diffie-Hellman
- DSA
- RSA

Wind River Security Libraries

Starting with VxWorks 6.6, cryptographic services are provided by the Wind River Cryptography component. The Wind River Security Libraries are also included for backward-compatibility with prior versions of VxWorks. This includes a library of cryptographic algorithms, the common crypto interface (CCI), used by other components requiring access to crypto functions, and the crypto provider interface (CPI), which supplies a mechanism for developers to add other crypto libraries or hardware-based crypto functions. The Wind River Security Libraries are being deprecated and new development should use the Wind River Cryptography Libraries, which provide enhanced functionality.

Wind River SSL and SSH

Wind River SSL is a client server technology used to secure any higher layer protocol that uses sockets. A typical application is to secure HTTP connections (HTTPS) for e-commerce.

Security is provided by the following:

- Privacy, using data encryption
- Authentication, using digital certificates
- Message integrity, using message digests

Features of Wind River SSL 6.7 include the following:

- SSLv2, SSLv3, TLSv1, and DTLSv1 support
- HMAC-SHA-1 and HMAC-MD5
- DES, 3DES, and AES
- RSA public-key cryptography
- Implementation of OpenSSL APIs to allow for easy porting of existing applications
- Advanced Encryption Standard (AES) Ciphersuites for TLS (RFC 3268)
- Datagram Transport Layer Security (RFC 4347)
- Update to version 0.9.8h of OpenSSL

The Wind River SSH (Secure Shell) protocol creates a secure terminal connection between an SSH client and an SSH server. This means embedded systems can communicate at the application level over a connection that is encrypted and provides data integrity and replay protection. This effectively eliminates eavesdropping, connection hijacking, IP spoofing, and other network-level attacks.

In addition, embedded SSH provides several secure tunneling capabilities that may be used to create VPNs. A variety of authentication methods is also supported.

Features of Wind River SSH 6.7 include the following:

- SSH server mode
- SFTP client support
- SSH versions 1.5 and 2.0
- Port forwarding
- Terminal connections and SFTP connections
- Integration with RADIUS Client
- Use of VxWorks sockets in place of IPCOM sockets in its API
- Directory structure changes
- Changes to some default values

Wind River RADIUS and Diameter

Wind River RADIUS Client is a full-featured implementation of the industry-standard remote authentication dial-in user protocol. Wind River RADIUS Client supports a complete set of functions for authentication, accounting, and security, and it has been verified against several commercial RADIUS servers, ensuring compatibility with a wide range of applications.

Wind River RADIUS Client 6.7 allows the network to determine whether a user is allowed access (authentication). Authentication is also used to determine that a message has not been fabricated or altered in transit. Authorization determines which network resources a user may access, and the accounting functions provide a record of usage.

The Diameter authentication, authorization, and accounting (AAA) protocol provides support for peering AAA transactions across the Internet. The Diameter base protocol provides the

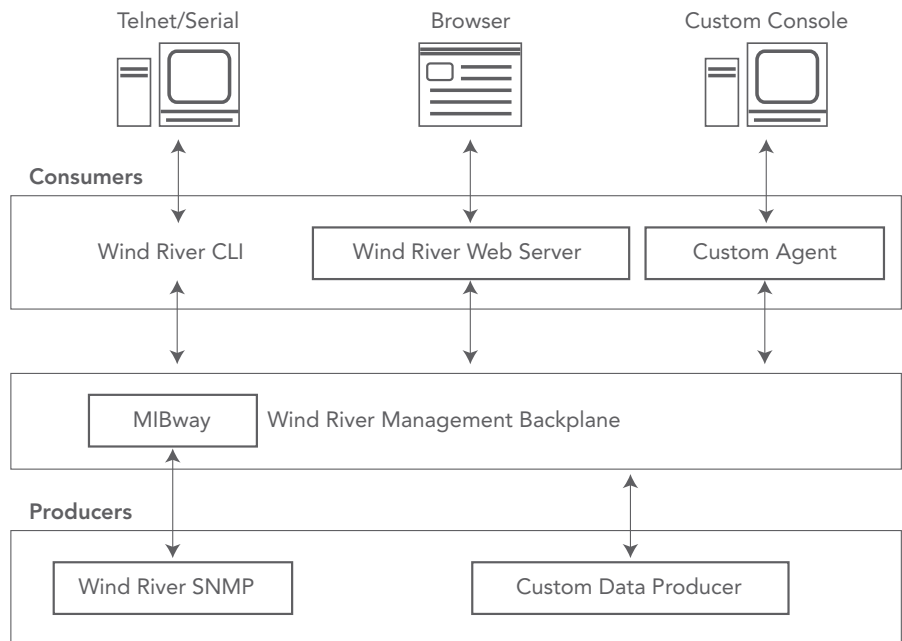


Figure 9: Wind River network management architecture

minimum requirements needed for AAA protocol, Mobile Internet Protocol version 4 (MIPv4), and remote network access applications. Diameter runs on one or more network access points, sending authentication requests to a shared Diameter server.

The Diameter protocol has several advantages over previous AAA protocols in that it offers improvements in the areas of reliability, security, scalability, and flexibility. Wind River Diameter Client 6.7 supports Diameter Proxy Agent.

The Diameter base protocol provides the following functionality:

- Connection and session management
- User authentication and capabilities negotiation
- Reliable delivery of attribute value pairs (AVPs)
- Agent support for proxy, redirect, and relay servers
- Extensibility, through the addition of new commands and AVPs
- Basic accounting services
- Dynamic peer discovery

Management

Wind River provides a scalable, unified, small-footprint management framework that enables creation of Web-based, CLI-based, or custom management interfaces to manage networked

elements. It consists of a management backplane, which acts as a conduit for data-handling between management interfaces (consumers) and manageable elements (producers). The scalable framework can have any type of consumers and any type of producers.

Wind River Management Backplane interfaces with a CLI agent, Wind River CLI; an embedded web server, Wind River Web Server; and an SNMP implementation, Wind River SNMP (Simple Network Management Protocol). In addition, the framework comes with a full-featured, Windows-based developer tool (GUI), Wind River Management Integration Tool (WMIT). This tool eases the development of management interfaces by bringing all the framework components together.

In addition to WMIT, Management Configuration Editor (MCE) is a simplified Eclipse plug-in to help with development of CLI- and Web-based management interfaces. MCE is integrated with Wind River Workbench and may be run on any host that Workbench supports. Developers may choose WMIT, MCE, or a combination of the two to develop a desired management interface.

Wind River SNMP

SNMP is designed to facilitate management and configuration of networked devices. Wind River SNMP is a highly portable, memory-efficient, standards-compliant implementation of SNMP specifically designed for original equipment manufacturers (OEMs) and system integrators who require full compliance with SNMP standards in a fast, small SNMP agent. This complete solution for integrated SNMP design and implementation includes a full management information base (MIB) development platform. It is composed of SNMP v1/v2c/v3 and AgentX.

Features of Wind River SNMP 10.2 include the following:

- Bilingual SNMP agent supporting SNMPv1/v2c protocols
- Asynchronous support
- SNMP v3 security
- SNMP notifications
- "Target" and "Notify" MIBs
- SNMP proxy
- SNMP v1/v2/v3 coexistence
- AgentX module
- MIB compiler
- Compact, interoperable, standards-based configuration
- Integration with Wind River Network Stack
- Portable design and implementation
- A new API to support SNMPv3 INFORM PDU operations

Wind River Web Server

Wind River Web Server is a scalable, secure, small-footprint commercial embedded web server. In addition to the functionality of a standard web server, it provides all the functionality needed to quickly create a browser-based management interface. Specifically designed for embedded systems, Wind River Web Server is HTTP 1.0 and 1.1 compliant and supplies SSL hooks to provide a secure transaction mechanism. The Wind River Management Integration Tool is a GUI tool that accelerates the process of creating and configuring a custom web-based management interface or a standalone web server by generating more than 70 percent of the required embedded code for such an application. The web-based management interfaces with a scalable backplane that interacts

with SNMP through Wind River MIBway or with any custom manageable data.

Features of Wind River Web Server 4.7 include the following:

- HTTP 1.0/1.1-compliant web server
- Bidirectional CGI layer
- LiveControl
- Visual integration tool
- GZIP/PKZIP compression
- Support for HTML, DHTML, CSS, JavaScript, and XML
- Drop-in support for SNMP objects
- Bundled SMTP email alerts
- User time-out sessions
- HTML/query string processing
- Bidirectional gateway for tying management objects to HTML pages, email alerts, and JavaScript libraries
- File-based uploads (RFC 1867)
- Server side includes (SSI)
- Secure authentication (basic base64 encoding and digest authentication)
- SSL hooks verified with OpenSSL
- Wind River SNMP inheritance through MIBway
- In-depth tutorial and sample code that steps through API usage, development tools, and best practices
- Ability to use a GET request to set WindMark values

Wind River CLI

Wind River CLI includes a full-featured command shell and development tools that enable users to build either the standard "craft" style interface or a custom interface to manage a networked device. CLI commands can be executed on a device through either an RS-232 or a Telnet connection. The Wind River Management Integration Tool is a GUI tool that accelerates the process of creating and configuring a custom web-based management interface or a standalone web server by generating more than 70 percent of the final embedded CLI code for such an application. CLI interfaces with a scalable backplane, which interacts with SNMP through MIBway or directly with any custom manageable data.

Features of Wind River CLI 4.7 include the following:

- Complete CLI management solution
- High-speed generation of command tree, handler functions, and individual commands

- Drop-in, prebuilt commands
- Telnet server
- Secure communication over SSH
- With MIBway, instant reflection of SNMP MIBs in the command line
- Same architecture as Wind River Web Server
- Command completion
- Context-sensitive help
- Command history
- Intermediate mode handling
- Parameter handling, verification, and grouping
- Negate commands (e.g., to restore defaults)
- Support for simultaneous Telnet sessions and serial ports
- Common command libraries
- Security parameters defined by object, command, or session
- Fully reentrant ANSI C code
- In-depth tutorial and sample code that steps through API usage, developed tools, and best practices
- Ability to use a GET request to set WindMark values

Wind River MIBway

Wind River MIBway enables developers to automatically leverage all SNMP objects for reuse in command-line and web-based management interfaces, with zero additional engineering effort required. Wind River MIBway provides an SNMP inheritance library to access data objects already instrumented for the Wind River SNMP agent. This makes it possible to leverage the thousands of developer hours already invested in writing MIB variables and code in a schema that is both flexible and scalable for future device management requirements.

Benefits of Wind River MIBway 4.7 include the following:

- Instant leverage of all existing SNMP code
- Reduction in development cost by cutting down months of coding and testing to days
- Creation of powerful, feature-rich web and command-line interfaces
- Separation of application logic from interface design
- Embedded code that handles SNMP-specific queries from Wind River Management Backplane
- Single-click integration with Wind River SNMP

- In-depth tutorial and sample code that steps through API usage, development tools, and best practices
- Ability to use a GET request to set WindMark values

Wind River Management Integration Tool

The Wind River Management Integration Tool is a Windows-based GUI that enables users to build web-based or CLI-based management applications by incorporating features to set project options, configuration options, build options, and resource constraints settings, and automate code generation. It provides an MIB compiler to facilitate Wind River SNMP users as well as a simple HTML editor as a miscellaneous tool.

Management Configuration Editor

Management Configuration Editor (MCE) is an Eclipse plug-in integrated with Wind River Workbench to help develop management interfaces. This tool is provided in addition to Wind River Management Integration Tool, so developers may use MCE on any host that supports Wind River Workbench (whereas Wind River Management Integration Tool is supported only on Windows). MCE functionality is limited when compared to that of Wind River Management Integration Tool. Wind River Management Integration Tool and MCE will coexist until all necessary functionality is available in MCE and customers have completed migration to MCE.

Wind River HTTP Client

Wind River HTTP Client is a library of client-side functions required to interact with an HTTP server. Using the functions provided, developers can write simple applications for transferring files (using GET and POST methods) with URL syntax.

Features of Wind River HTTP Client include the following:

- Conformance with HTTP 1.1 specification
- Support for GET and POST methods
- Support for cookies
- Support for direct and proxy server requests
- Secure HTTP communication with HTTPS

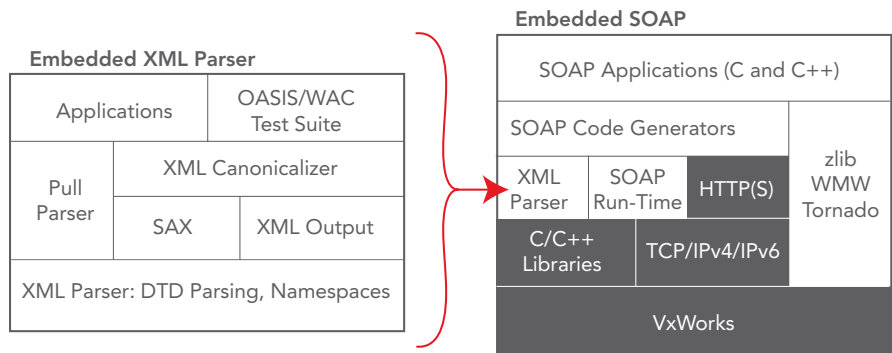


Figure 10: Wind River Web Services

Distributed Messaging and Services

Wind River Web Services

Wind River Web Services enable distributed applications running on a variety of platforms to communicate and interoperate seamlessly in a manner conforming to Web Services standards. They allow developers to create interoperable Web Services applications (clients and servers) by providing the fundamental building blocks: XML, SOAP, WSDL, and GUI-based compiler tools for Web Services intermediate code generation and compilation. Wind River Web Services are standards-compliant and specifically tuned for embedded systems applications.

Features of Wind River Web Services 1.6 include the following:

- WS-Security
- WS-I Basic Profile 1.0 conformance
- Clear and structured code and documentation for ease of configuration and maintenance
- Standards-conformant: XML 1.0, SOAP 1.2

- Communication through use of SOAP messages with any system supporting Web Services, including Microsoft .NET or Apache Axis
- Graphical WSDL compiler tool integrated into Wind River Workbench

Wind River DCOM

Wind River DCOM is the technology that implements Component Object Model (COM) and distributed COM (DCOM) on VxWorks. The name Wind River DCOM refers both to this technology and to the product. Wind River DCOM, an implementation of Microsoft's DCOM, scaled for device software development, provides the foundation for management protocols such as OLE for Process Control (OPC). Wind River DCOM enables application distribution between VxWorks-based devices and non-real-time devices, such as desktop computers. It allows developers to seamlessly integrate device software applications with Windows applications. Wind River DCOM provides a compact,

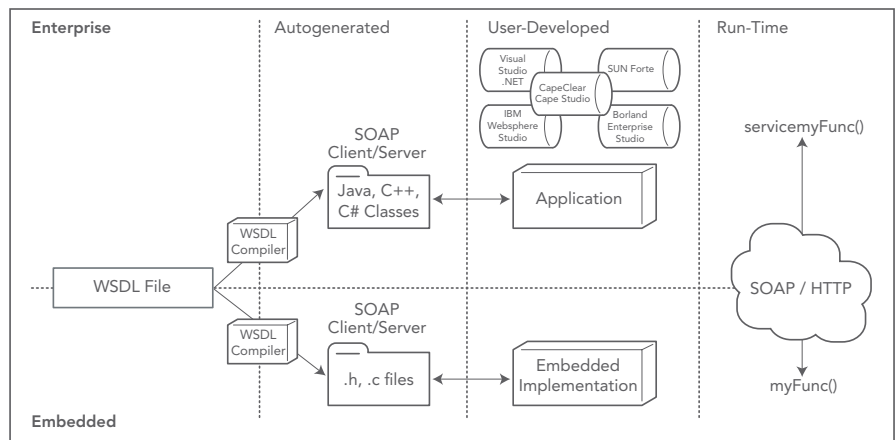


Figure 11: Web Services software development flow

performance-focused solution targeted specifically for the strict requirements of devices.

Features of Wind River DCOM 2.3.7 include the following:

- Small footprint
- Source-compatible with Win32 SDK COM and DCOM API
- Object template library that provides convenience methods for DCOM application developers
- IDL compiler (compiling IDL files into Wind River DCOM application classes)
- GUI wizard for creating DCOM interfaces

Bridging and Routing

Wind River Learning Bridge

Wind River Learning Bridge is a basic implementation of a transparent layer 2 Ethernet learning bridge that learns the network topology by analyzing the source address of incoming frames from all attached networks. The Learning Bridge attaches above the MUX layer as a SNARF network service type, and it includes two mirror END drivers used to bridge traffic destined for a stack located on the same machine as the bridge.

Wind River Learning Bridge 1.3.3 includes a station cache: a basic database that stores the relationship between MAC addresses and the ports from which it sees frames associated with those MAC addresses. Forwarding decisions are based on this cache. Both source and destination MAC addresses are used to build the cache database, and a cache-aging algorithm removes inactive entries. Learning Bridge is used by the Wind River Wireless Security Authenticator.

Graphics and Local User Interface

Wind River Media Library

The scalable Wind River Media Library facilitates and speeds GUI development locally within a device. The library consists of a software development component and a driver development component that together provide a hardware abstraction layer for graphics, video, audio, and input devices.

The software development component is used for developing hardware-independent applications for a variety of

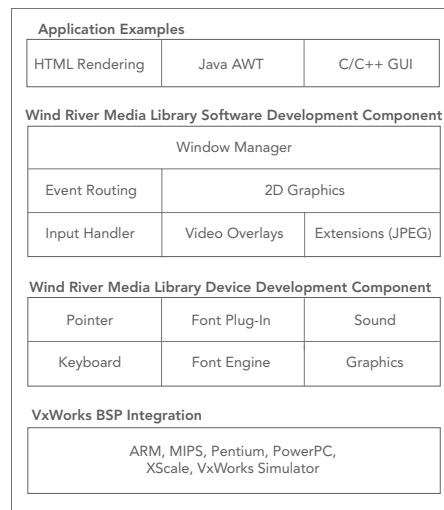


Figure 12: Wind River Media Library

platforms. It includes a comprehensive API for 2D graphics, window management, region management, text display, color management, video overlay support, alpha blending, and JPEG image support. The 2D API allows hardware-accelerated features to be used in a hardware-independent manner. The integrated window manager and input event routing facilities enable multiple applications to share one screen. For example, native C/C++ applications and Java-based applications can execute simultaneously while sharing the input and output device. The event service handles input events and routes them to specific applications, and it controls cursors for pointer devices, such as touch screens, mice, and remote controls.

The device development component is used for implementing drivers. It interfaces directly with the application's target hardware devices, including graphics chips, video controllers, LCD displays, audio chips, keyboards, and touch screens, and it provides a rich set of reference drivers for common hardware configurations. This component includes generic frame buffer drivers for 1-, 4-, 8-, 16-, and 32-bit color modes, which allow developers to bootstrap new drivers quickly and support any graphics device easily. The graphics driver framework provides both a native Media Library interface and an XFree86 driver interface, enabling development on the most current graphics hardware in the

market. Media Library also includes an integrated FreeType font engine, which supports TrueType fonts and enables faster GUI application development. Finally, the driver development component is extensible, so it can accommodate the specific hardware functionalities of a device.

New features of Wind River Media Library 5.2 include the following:

- Built-in support for the JPEG and PNG image formats is included. Additional image formats can be added by attaching the format to the imaging library.
- Previously, Media Library supported only JPEG images and used an API that was specific to JPEG images. Media Library now supports additional image formats through a general-purpose image API. Using this API, images can now be manipulated in a consistent fashion, regardless of the image format.

Additional features of Wind River Media Library include the following:

- FreeType font engine v2.3.2
- Screen rotation support
- VxBus support
- Additional XFree86 resolutions
- Virtual input support
- Shared data library (optional) for kernel applications
- X.org drivers support
- Multiple display functionality
- Video overlay enhancements
- Drawing surface extensions, including support for OpenGL
- Menu and button widgets
- Timers
- Touch screen calibration support
- RTP support for local bus graphics devices
- XFree86 driver support

Connectivity

Wind River CAN

Wind River CAN 1.5.5, an implementation of the Controller Area Network (CAN) protocol for VxWorks embedded targets, provides developers with a standardized interface to one or more CAN devices. The product supports many of the popular CAN boards in use today, such as ESD PC104 200, ESD PCI 200, MCF5282 FlexCAN, PPC5xx TouCAN, and MSMCAN PC104. In addition, the Wind River CAN API is

independent of the target architecture and the I/O mechanism used to access the CAN controller. This standardized interface greatly simplifies programming CAN hardware and allows applications to be ported quickly to new target architectures with minimal or no changes to user application source code. Additional protocols, such as CANopen (available from Wind River partner IXXAT) and DeviceNet, are integrated on top of this standard interface.

Wind River OPC

Wind River OPC, our implementation of the OLE for Process Control (OPC) specification for the VxWorks RTOS, is a nonproprietary technical specification that defines a set of standard interfaces based on Microsoft's OLE/COM technology. The OPC standard protocol enables interoperability between automation/control applications, field devices, and business/office applications by allowing clients and servers from different vendors to speak the same standard language. OPC streamlines development of device interfaces, increased connectivity, and interoperability between custom applications and allows applications to access subsystem data easily. The Wind River OPC implementation is designed specifically to support devices with high-performance and small footprint requirements.

Wind River OPC 3.1.6 support includes the following:

- OPC Data Access Custom Interface Standard 2.05a specification
- OPC Alarms and Events 1.1 specification
- OPC Data eXchange 1.0 specification
- Interactive sample OPC client
- Optimization for real-time devices
- Integration with development tools

Wireless

Wind River Wireless Ethernet Driver

Wind River Wireless Ethernet Driver provides access point and station-side support for the industry-standard 802.11a, b, and g protocols. Included are drivers for Atheros AR500x and Broadcom BCM43xx chipsets. 802.11 draft-n and Wi-Fi Multimedia Enhancement (WME) are supported in the Broadcom driver. A fully abstracted hardware

interface layer provides ease of portability to other wireless chips. The driver supports multiple hardware interfaces on the same target, allowing for more advanced applications. The driver can be used in a wide variety of target hardware platforms. A standard IOCTL application interface is provided for user configuration and control. Wireless security is provided through preintegration with the Wind River Wireless Security protocol, supporting the 802.1X, WPA, WPA2, and 802.11i standards, and Wind River Wireless Ethernet Driver facilitates management of spectrum regulation differences between different countries through 802.11d support.

The security implementation includes preshared keys and TKIP (Temporal Key Integrity Protocol). A range of encryption and hashing algorithms is available to give developers flexibility in trading off security level versus performance.

Features of Wind River Wireless Ethernet Driver 3.1 include the following:

- Supports multiple modes: 802.11a, b, g
- Supports country-specific spectrum regulations through 802.11d
- Supports WME, enables packet routing prioritization based on the type of service (TOS) of each packet
- Supports Wi-Fi Protection Access (WPA/WPA2) and 802.11i for security

- Supports SMP for the SMP-supported BSPs
- Supports multiple service set identifiers
- Directly supports Atheros AR500x chipsets
- Directly supports Broadcom BCM43xx chipsets
- Ports easily to other chipsets
- Supports both access point and station side
- Supports multiple hardware instances on the same target
- Features user configuration and control through use of IOCTL controls
- Features industry-standard application interface format: ioctl (descriptor, function, value)
- Extends easily to support new features and customer requirements
- Maps directly into the 802.11MIB
- Features extensive debugging/logging support and show routines
- Configures through command line or Wind River Workbench
- Integrates with Wind River Wireless Security protocol: 802.1X

Wind River Workbench Development Suite

Wind River Workbench is a collection of Eclipse-based tools that accelerates time-to-market for developers building devices with VxWorks. Workbench offers the only end-to-end, open standards-based collection of tools for device

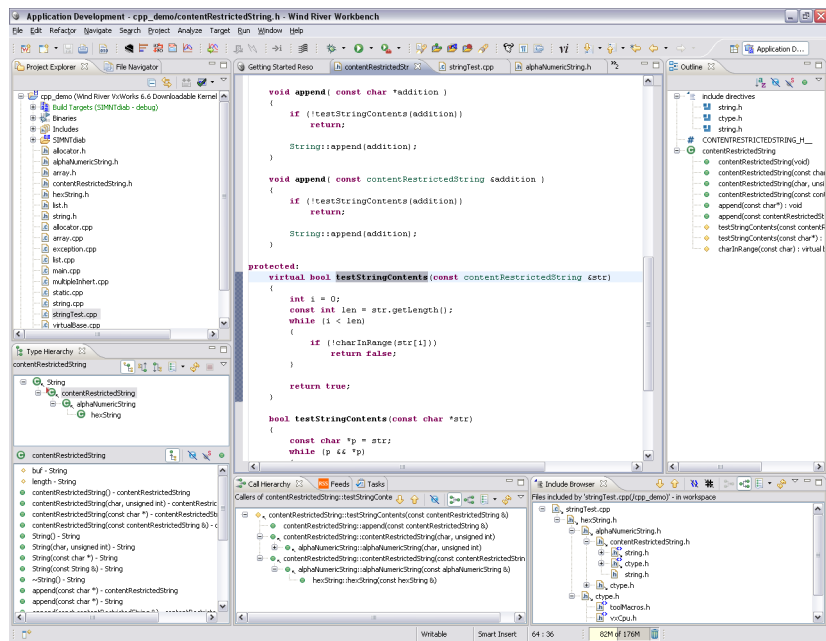


Figure 13: Wind River Workbench

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, and Linux
 - Target processor support for ARM, ColdFire, Intel Architecture/Pentium, MIPS, PowerPC, and XScale processors
 - 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

This version of the VxWorks platform features Workbench 3.1, which includes significant new capabilities in support of increased development team productivity:

- General Workbench enhancements
 - Dynamic printf debugging support
 - User interface improvements for run-time analysis tools
 - Device debug framework (DSF) alignment with Eclipse
 - Improved search and source code indexing
 - Reverse debugging (for simulation targets)
 - Eclipse 3.4 framework
 - New host OS support
- VxWorks platform enhancements
 - VxWorks Source Build configuration tool
 - Debugger support for VxWorks AMP
 - Support for overlapping memory RTPs
 - VxWorks Simulator networking enhancements
 - VxWorks 5.5 support for Intel architectures

- On-chip debugging enhancements
 - Workflow and usability improvements
 - Wind River ICE 2 for multicore processor debugging
 - Wind River Trace 2 for nonintrusive configurable trace
 - VxWorks SMP and AMP awareness
 - Additional processor support for ARM, MIPS, and PowerPC
- Device management enhancements
 - Integration with new Wind River Test Management system
 - Downloadable agent that adds diagnostic support without reconfiguring the kernel
 - Expanded processor support

Eclipse

Because of its openness, capability, and strong community support, Eclipse was chosen as the framework for the Wind River Workbench development suite. The Eclipse 3.4 framework supplies the necessary infrastructure to graphically and functionally integrate the components of Workbench. 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 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, www.eclipse.org.

Wind River Workbench was awarded the “Best Commercial Eclipse-Based Developer Tool” by the Eclipse Foundation at EclipseCon 2008. It was one of three finalists selected from more than 55 nominations by a panel of judges from the Eclipse community. Wind River

Workbench was selected as the winner for its usability of features, completeness of solution, and the manner in which the product uniquely solves problems faced by developers of embedded device software.

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. VxWorks projects of different types can be created for configuring and building the VxWorks source, kernel images, and board support packages as well as application projects based on real-time processes or user-developed kernel modules. By design, Workbench enables users to manage multiple projects simultaneously.

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-settings, fine-grained control at the level of an individual file, and everything in between.

Command-Line Project and Build System

In addition to the Workbench Build System, a full-featured command-line build system is provided for developers who prefer this type of environment. Using GNU make, Tcl libraries, and VxWorks-specific tools, developers can configure VxWorks source and build VxWorks images as well as develop applications and libraries using command-line build facilities. These command-line tools are often integrated into a customized build system.

Indexed-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.

Wind River Compiler and Wind River GNU Compiler

Wind River Compiler is the default C/C++ compiler for most processor architectures configured for building the VxWorks 6.x kernel, libraries, BSPs, and applications in Wind River Workbench. This compiler's optimization capabilities are based on and extend the industry-hardened Diab Compiler technology and it produces robust, tight, fast-executing code.

Wind River Compiler includes the following:

- Superior optimization technology to generate fast, compact, high-quality code
- 100 percent compatibility with the latest ANSI C specifications (ISO/IEC 9899:1990 ("C89") and ISO/IEC 9899:1999 ("C99") standards) and with the latest ANSI C++ specifications ISO/IEC 14882:2003(E) C++ standard
- Standards conformance (EABI, ELF, and DWARF) for maximum tool interoperability
- Complete control of code and data memory allocation
- Position independent code (PIC) and position independent data (PID) support
- Whole program optimization
- Proven performance with VxWorks

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.

Wind River GNU Compiler is the default compiler for Intel architectures and is based on the Free Software Foundation (FSF) distribution of the GNU compiler. Wind River has modified a general distribution version of the compiler specifically for use with VxWorks 6.x. The primary areas of modification deal with support for RTPs and shared libraries.

Wind River GNU Compiler includes the following:

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

Both compilers are included and supported as part of Wind River VxWorks platforms.

Workbench Debugger

Workbench Debugger addresses the needs of developers involved with hardware bring-up, firmware/driver/BSP development, kernel development, and application development. It provides more capability than the GNU debugger (GDB) or other basic source-level debuggers. The Workbench Debugger 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. These capabilities can be extended further with Wind River's on-chip debugging solutions. In combination, these tools provide the necessary functionality for hardware bring-up, device driver/BSP debugging, kernel debugging, and application software debugging.

The Workbench Debugger supports VxWorks debugging for single-core processors as well as multicore processors running VxWorks in either SMP or AMP mode. 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

The Workbench Debugger provides extensive browsing and inspection capabilities on objects in the target platform. This awareness of VxWorks 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

New debugging features in Workbench 3.1 include the following:

- Support for adding dynamic `printf` commands using the Workbench Editor, the host shell, and the VxWorks kernel shell, allowing insertion of `printf` commands into code without recompiling and relinking the application
- Support for debugging AMP systems over an MIPC connection, allowing debug connections to processor cores that do not have an external device to use to establish a debugging connection
- Support for debugging VxWorks RTPs with overlapped memory

VxWorks Simulator

VxWorks Simulator 6.7, formerly known as VxSim, is a complete prototyping and simulation tool for VxWorks 6.x applications. VxWorks Simulator supports simulation of SMP systems, up to a maximum of 32 CPUs on any supported host. 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.

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 SMP, 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 simulator runs on your chosen host workstation, decreasing the need for evaluation hardware early in the development cycle. 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.

Workbench VxWorks Source Build Configuration

Wind River Workbench 3.1 provides a new project type that allows you to configure and build custom VxWorks libraries, which you can then use as the basis for VxWorks image projects. This new project type supports the creation and management of VxWorks Source Build (VSB) projects. These projects are based on a specific BSP and allow the customer to adjust 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 purchase VxWorks source code, VSB projects can only be set up if the sources are installed.

VxWorks 6.x Kernel Configurator

VxWorks 6.x Kernel Configurator is a graphical utility that simplifies and accelerates the task of selecting the operating system components that must 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.

When creating a new bootable kernel image, Workbench analyzes available kernel components as well as BSP and compiler selections. The configurator displays a summary of key configuration data, such as the 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 provided with the configurator include components needed for POSIX compliance, real-time process development, or error management.

Selecting kernel components individually gives you greater flexibility and control over your VxWorks image. VxWorks 6.x Kernel Configurator analyzes component dependencies and highlights conflicts when components are required but not selected or if components are incompatible with one another. An autoscale feature analyzes the entire VxWorks image and removes unused kernel components that may increase the size of a bootable image unnecessarily.

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

Host Shell

The Host Shell, formerly known as WindSh, provides a command-line interface that allows you to download application modules and invoke both VxWorks 6.x and application module subroutines. This facility has many uses:

- Interactive exploration of the operating system by calling any VxWorks routine and API
- Debugging and monitoring processes
- Interactive exploration of VxWorks 6.x RTPs
- Interactive development by calling any application (RTP) routines
- VxWorks 6.x application (RTP) and kernel testing
- Error management support through output of error dumping; the ability to turn on/off error management on a per-task or per-RTP basis
- Message channels (IPC) support through text dump of the message traffic
- Debugging through dynamic printf insertion
- Debugging of multiple instances of VxWorks in an AMP multicore configuration

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 VxWorks 6.x tools, only the target agent is required on the target system. Therefore, the Host Shell can remain available at all times—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 helps with debugging drivers.

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. The Kernel Shell provides similar capabilities to those provided by the Host Shell; it is often used when control or visibility into system status is needed outside a development environment. The kernel works with VxWorks SMP. The task information display includes the CPU/core on which tasks run.

A tip utility, similar to the tip utility available with UNIX, has been added in VxWorks 6.7 to enable the use of the Kernel Shell and Host Shell across multiple instances of VxWorks in a multicore AMP configuration. This capability makes it possible to display output from one or more cores on the same console of a separate host machine and to log on to individual cores remotely and execute shell commands on them. The tip utility uses an MIPC-based serial device for communication between cores.

The Kernel Shell command interpreter provides new capabilities in VxWorks 6.7. Now the Kernel Shell can dynamically spawn tasks in RTPs, as well as in the kernel, which is useful for debugging purposes. The Kernel Shell can also dynamically call routines in RTPs without the overhead of creating and spawning a task each time a routine is dynamically called.

Run-Time Analysis Tools

Workbench contains 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.

System Viewer

System Viewer provides detailed analysis and graphical visualization of VxWorks 6.x system events, revealing the complex interactions of tasks, interrupts, and system objects of an application executing on a target. Context changes are clearly shown, as are system events such as semaphores, message queues, signals, tasks, timers, and user events. System Viewer allows device software developers to detect anomalous behavior quickly then understand the cause and effect by reviewing the complete history of events leading up to the problem, including error management events from VxWorks 6.x.

Performance Profiler (Formerly ProfileScope)

Profiling is critical for real-time systems. Once you understand performance bottlenecks, it becomes easier to optimize application code. 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.

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 noncharacterized memory leaks. Memory Analyzer is an instant memory analyzer that 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.

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. Data Monitor presents this live analysis of your program without stopping or slowing your code.

Code Coverage Analyzer (Formerly CoverageScope)

Code Coverage Analyzer enables analysis of 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.

Optional Add-Ons

VxWorks 6.7 Multiprocessing

VxWorks 6.7 offers support for SMP and AMP as an optional add-on product. The capabilities of the add-on are described in detail previously in the section titled “VxWorks 6.7 Multiprocessing.”

Wind River Test Management and Wind River Lab Diagnostics

Two interoperable device management products 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.

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. With Test Management, you can do the following:

- 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 Lab Diagnostics is a distributed application that enables engineers to isolate, diagnose, and resolve issues encountered in devices at run-time. It allows companies to greatly enhance productivity by streamlining system integration, software verification, and product validation. Lab Diagnostics allows software diagnostics instrumentation (e.g., printf, logs) to be injected

dynamically to running devices without recompiling the device software or stopping and reloading the device. Patches can also be applied “hitlessly” at run-time. Lab Diagnostics dramatically reduces time-consuming instrumentation-build-test cycles, enabling developers to wring out tough problems faster at lower costs.

Designed for workgroups, Lab Diagnostics provides a server-based environment that enables collaboration between developers and testers. This standalone product is interoperable with both command-line and Wind River Workbench environments and is sold as an add-on to Wind River platforms. Its companion product, Wind River Field Diagnostics, enables support engineers to use the same run-time diagnostics and patching technology to better support products in the field, improving uptime and mean-time-to-repair, while reducing costs.

Wind River Workbench, On-Chip Debugging Edition

The Workbench development environment provided with Wind River platforms can be enabled for on-chip debugging. Wind River’s on-chip debugging capability, along with Wind River’s line of JTAG-based hardware debuggers, enable 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. Wind River Workbench, On-Chip Debugging Edition 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 an essential 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 Workbench capabilities offered through the on-chip debugging connection include the following:

- Support for a wide range of processors, including PowerPC, MIPS, ARM, and ColdFire (contact your Wind River representative for details on supported processors)
- On-chip debugging target connection manager
- On-chip debugging command shell
- On-chip debugging console
- Flash programming
- Hardware and memory diagnostics
- Compact flash options
- JTAG editor
- Extensions to Register view, including bit-level register details and additional peripheral register support for most processors
- On-chip debugging user’s perspective within Workbench
- External real-time trace with Wind River Trace and Wind River Trace 2 (may require additional hardware, to be purchased separately)
- Firmware update
- Cache Memory view
- Statistical performance analyzer (PFA)
- On-chip debugging reset and download/launch
- VxWorks 5.5, VxWorks 6.x, Linux, and ThreadX OS awareness via JTAG

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

IPL Cantata++ for Wind River Workbench

IPL Cantata++ for Wind River Workbench, formerly Workbench Unit Tester, available for VxWorks 5.5 and VxWorks 6.x-based platforms, is a set of tools that allows developers greater efficiency in completing unit testing, integration testing, and code coverage analysis on the tests. The integration of Cantata++ with Workbench places these capabilities within easy reach. Cantata++ increases software quality, decreases time-to-market, and reduces support costs through better, faster, more automated testing in the development life cycle.

Datalight FlashFX Pro Version 3.3

This component provides NAND flash memory support.

Technical Specifications

VxWorks 6.7

- VxWorks 5.5, 6.0, 6.1, 6.2, 6.3, 6.4, 6.5, and 6.6 compatibility
- Kernel scalability and performance tuning using VxWorks Source Build
- State-of-the-art memory protection
- Memory management
- Error management
- Message channels IPC, including support for multiprocessor and multi-OS messaging using TIPC
- Improved POSIX compliance, including full support for JTRS SCA AEP 2.2.2 and to POSIX IEEE Std. 1003.13-2003 PSE52
- Dual-mode IPv4/IPv6 network stack
- Power management framework, with CPU power management
- TrueFFS (flash file system)
- dosFs (FAT-compatible file system)
- Highly reliable file system (HRFS) with configurable commit points
- High-speed interconnect framework with PCI and local bus support
- VxMP 2.3.4
- Wind River TIPC 1.7
- Wind River Network Stack 6.7
- Wind River PPP 6.7
- Wind River USB 2.4
- Wind River IPsec and IKE 6.7
- Wind River Crypto 6.7
- Wind River Security Libraries 1.4
- Wind River SSL 6.7
- Wind River SSH 6.7
- Wind River RADIUS and Diameter 6.7
- Wind River Firewall 6.7
- Wind River NAT 6.7

- Wind River Wireless Ethernet Driver 3.1
- Wind River Wireless Security 6.7
- Wind River EAP 6.7
- Wind River Mobile IP 6.7
- Wind River OPC 3.1.6
- Wind River DCOM 2.3.7
- Wind River CAN 1.5.5
- Wind River CLI, Web, MIBway 4.7.1
- Wind River SNMP 10.3
- Wind River Learning Bridge 1.3.3
- Wind River Media Library 5.2
- Wind River Web Services 1.6

Workbench 3.1

- Based on Eclipse platform 3.4
- Project System
- Build System
- Index-based global text search-and-replace
- Wind River compilers
 - Wind River Compiler for VxWorks 5.7
 - Wind River GNU Compiler 4.1.2
- Debugger
- Target debug agent for VxWorks
- VxWorks Simulator
- Host Shell and Kernel Shell environments
- VxWorks Kernel Configurator
- Run-time analysis tools
 - System Viewer
 - Performance Profiler (formerly ProfileScope)
 - Memory Analyzer (formerly MemScope)
 - Data Monitor (formerly StethoScope)
 - Code Coverage Analyzer (formerly CoverageScope)

Architectures, Hosts, and Board Support Packages

Supported Target Architectures

- ARM
 - ARM7
 - ARM9
 - ARM11
 - ARM MPCore
- ColdFire (requires Service Pack 1 for VxWorks 3.3 platforms)
 - ColdFire v3
 - ColdFire v4e
- Intel
 - Pentium family (Pentium, Pentium Pro, Pentium II, Pentium III, Pentium 4, Pentium M)
 - Xeon LV
 - Core
 - Core 2 Duo
 - Atom

- Intel/Marvell XScale
 - IXP4xx
 - IXP2xxx
- MIPS
 - MIPS 4Kx
 - MIPS 5Kx
 - MIPS tx49xx
 - MIPS24Kx
 - MIPS34Kx
 - MIPS74Kx
 - BCM SB1 (1250, 1125, 1122, 1121)
 - BCM SB1a (1480, 1455, 1280, 1255, 1155)
 - Cavium OCTEON CN3XXX
 - Cavium OCTEON CN5XXX
 - RM9000GL
 - VR55xx
 - Raza XLR
- PowerPC
 - PowerPC 40x
 - PowerPC 44x
 - PowerPC 51xx
 - PowerPC 60x
 - PowerPC 7xx
 - PowerPC 8xx
 - PowerPC 52xx
 - PowerPC 74xx
 - PowerPC 82xx
 - PowerPC 83xx
 - PowerPC 85xx, 8572
 - PowerPC 86xx, 8641d
 - PowerPC 970

Supported Hosts

- Windows XP Professional, Service Pack 2
- Windows Vista (Business and Enterprise)
- Red Hat Enterprise Linux Workstation 4, Update 5 or 6, x86 (32-bit)
- Red Hat Enterprise Linux Desktop with Workstation option 5, Update 2*
- Red Hat Fedora Core 9, x86-64
- openSUSE (SUSE Linux) 10.2*
- openSUSE (SUSE Linux) 11, x86-64
- Ubuntu Desktop 8.04, Update 4, x86-64
- Solaris 9 (with GTK), Update 09/05, SPARC 32-bit
- Solaris 10 (with GTK), Update 09/05, SPARC 32-bit

**Both 32-bit and 64-bit versions are supported. For 64-bit version, only x86-64 is supported. IA-64 (original Intel Itanium architecture) is not supported.*

Supported Board Support Packages

The VxWorks platforms support a wide variety of board support packages on the target architectures listed previously. For a complete list of available BSPs, visit the Board Support Packages section of the Wind River website at www.windriver.com/products/bsp_web/index.html.

Partner Ecosystem

Wind River's world-class partner ecosystem assures tight integration between our core technologies and those of the premier hardware and software companies we've chosen to complement our solutions. Our partners help extend the capabilities of Wind River's development and run-time platforms by offering out-of-the-box integration and support for key technologies in the fast-moving industrial market. Our customer support team is trained to troubleshoot partner technologies in use with Wind River products, making ours the most comprehensive and best supported partner ecosystem in the DSO industry.

Our hardware partners include the following:

- Advantech
- AMCC
- ARM
- Broadcom
- Cavium
- Continuous Computing
- Curtiss-Wright Controls Embedded Computing
- Emerson Network Power
- Freescale
- GE Fanuc
- IBM
- Intel
- Kontron
- MIPS Technologies
- Radisys
- RMI
- Sun Microsystems
- Texas Instruments
- Xilinx

Our software partners include the following:

| Technology | Partner |
|---|---|
| Ada development | AdaCore |
| Bluetooth | Parrot, iAnywhere |
| CORBA | Objective Interface Systems (OIS) |
| Databases | Gracernote, Oracle |
| ATM | DCL |
| Design tools | IBM Rational, IPL |
| DTV technology | Zentek |
| Fieldbus solutions | HMS, IXXAT, Softing, Woodhead (Applicom, SST) |
| File system fail-safe | Datalight |
| File system: NAND Flash | Datalight |
| File system: CIFS | Visuality Systems |
| 2D graphics and GUI | PSA, Tilcon |
| 3D graphics and OpenGL | ALT Software, Seaweed Systems |
| GPS | Trimble |
| HTML-based UI and technology (browsers) | Access, Espial, Opera Software, ProSyst |
| IEEE 1394 | VividLogic |
| Industrial Ethernet, CANopen stack | IXXAT |
| Input methods | Agfa, Bitstream, Zi |
| IP routing, MPLS | Data Connection Ltd., IP Infusion |
| Java VM | aicas, Aonix, Aplix, Esmertec, Skelmir |
| Layer 2/3 switching | IP Infusion |
| Layer 2/3/4 with Ethernet | LVL7 |
| MOST | SMSC |
| Navigation | Destinator, 3DVU, 3SOFT |
| NDDS stack | RTI |
| OS coresidency | KUKA Controls |
| Security (consumer) | Nagravision |
| Speech recognition | Asahi Kasei, Conversay, Fonix, Nuance |
| UPnP | Allegro |
| Voice Signaling | DCL, Flextronics |
| QoS | Data Connection Ltd., IP Infusion |

Professional Services

Wind River Professional Services, a CMMI Level 3-rated organization, enables you to reduce risk and focus on development activities that add value and differentiate your design. As part of our comprehensive 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.

Backed by our commercial-grade project methodology, Wind River Professional Services include the following:

- Requirements discovery and definition
- BSP and driver optimization
- Software system and middleware integration
- Application and infrastructure development
- Hardware and FPGA design for prototyping or market-ready systems

Typical projects range from two to four man-weeks for driver and BSP implementation, to one man-month to one man-year for hardware design or extensions to an existing software solution, to multi-man-year programs that bring customer concepts to reality through design, creation, and system test and verification.

Professional Services has extensive experience with platform design, including safety critical systems and

navigation/infotainment systems. Professional Services has implemented both hardware and software solutions for the embedded device market and continues to work with standards organizations to establish the next-generation platforms.

Installation and Orientation Service

Proper installation and orientation of the VxWorks 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 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.

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 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 include the following:

- 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 Service Capability and Performance (SCP)–certified organization, provides support for all Wind River VxWorks platforms. Your subscription to VxWorks platforms includes full maintenance and support, delivered through Wind River's Online Support (OLS) website and our worldwide technical support team. While under subscription, customers receive both maintenance updates and major upgrades.

Visit Wind River Online Support at www.windriver.com/support for fast access to product manuals, downloadable software, and other problem-solving resources. 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.

Additional support features, including proactive email alerts covering particular technologies, platforms, or product patches and technical tips for common problems, are available for all customers on subscription. OLS visitors can also access a community of developers to discuss their issues and experiences.

Support on modified or unsupported configurations is best-effort-based. Wind River Customer Support will try to reproduce the problem on a supported configuration. If the problem can be validated, Wind River will provide a fix that will be tested on a supported configuration. Wind River Professional Services can provide support for boards or host operating system versions that are not supported by the standard product, as well as for customized versions of the source code or additional nonstandard packages.

Customers with a valid support or subscription agreement are eligible for all updates and major upgrades to their platform free of charge. If customers cannot update to a new version but need critical parts of the update applied to an older version of the product, Wind River Professional Services can be engaged to backport the required functionality on a case-by-case basis.

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.

For more details on our support processes including escalations and defect resolution, consult Wind River's Customer Support User's Guide (CSUG), available at www.windriver.com/support/resources/csug.pdf.

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, 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 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

© 2009 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. Rev. 01/2009