WIND RIVER

Wind River Linux 4

Table of Contents

As project resources and time frames shrink, developers must leverage cost-effective, leading-edge technologies to keep up with market requirements. Meanwhile, advances in Linux create emerging opportunities to use open source solutions for a variety of projects. Linux is the ideal platform for building embedded devices and is gaining rapid adoption, given the flexibility, innovation, performance, and total cost of ownership advantages of the open source model.

Wind River Linux 4 is a commercial-grade Linux solution for embedded device development. The platform contains a fully tested, validated, and supported Linux run-time image based on Linux 2.6.34 kernel technology. Wind River Linux offers the optimal combination of integrated and validated open source software with advanced features, optimized for specific target device markets.

Wind River Linux 4 is delivered in an optimized platform design to address the unique needs of developers of devices for markets such as aerospace and defense, networking, industrial and medical devices, mobile and multimedia, and consumer electronics.

Figure 1 is an overview of the components of Wind River Linux. The product consists of source code and a build system that generates an optimized run-time image suitable for embedded devices. The components of the product are referenced by the developer to create a defined run-time image.

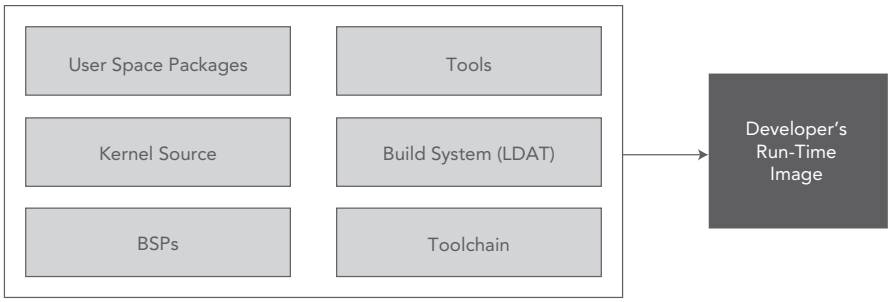

Figure 1: Major components of Wind River Linux

Key Components

Wind River Linux contains the following components:

- **Application packages:** Hundreds of software packages that operate in protected Linux user mode
- **Kernel source:** The 2.6.34 Linux kernel with many fixes and feature enhancements
- **Board support packages:** Hardware enablement components
- **Tools:** Software development tools, including the award-winning Eclipse-based Wind River Workbench development suite
- **Build system:** The Wind River Linux Distribution Assembly Tool, which is used by the developer to compile and assemble these components
- **Toolchain:** The cross-compiler based on the GNU Compiler Collection (GCC)
- **Pre-configured profiles for market-specific device types:**
  - Consumer premise equipment profile
  - Industrial equipment profile
  - Network equipment profile
  - Network equipment profile with Linux Standard Base (LSB) support
  - Small footprint network equipment profile
  - Mobile multimedia device profile

Key Benefits

The following are some of the competitive advantages of Wind River Linux:

- Commercial product quality with extensive testing and quality assurance, reliable service packs, and security patches with standard software product life cycle support
- Extensive hardware and software ecosystem support
- Lower costs, by eliminating the burden of building, supporting, and maintaining your own Linux distribution, allowing you to focus on differentiating applications rather than on maintaining Linux itself
- Reduced complexity of present and future projects, by leveraging the Wind River Linux cross-build system and layers development methodology
- Rich tools and development-environment support based on the Eclipse framework
- Compliance with industry standards such as Carrier Grade Linux to meet market-specific needs
- Delivery of advanced functionality and capabilities with guaranteed real-time performance, advanced networking stacks, and enhanced multi-core support
- Predictable roadmap for long-term product planning
- Complete and detailed documentation of software package license information to help ensure compliance with legal and regulatory requirements

Wind River Linux Distribution Assembly Tool

Wind River Linux Distribution Assembly Tool (LDAT) is a build system to cross-compile and assemble components for run-time images. LDAT is licensed under the GNU General Public License version 2. LDAT commonly addresses the use cases shown in Figure 2.

Project Creation

Users start by creating a project using LDAT. They reference configuration information such as the hardware target, kernel type, profiles, and pointers to other custom software. This creates a project. LDAT uses autoconf to generate a configured build directory. There are a large number of options to select the board, kernel configuration, user space configuration, and so on. The profile option may be used to automatically select the kernel and user space configuration based on the selected board support package (BSP) and profile combination. The core layers are selected automatically based on the LDAT configuration directory (toolchain version, kernel version, etc.) unless overridden by user arguments.

Additional layers may be included either at the user's request or automatically by other layers. The configure script searches for layers specified without an absolute or relative path.

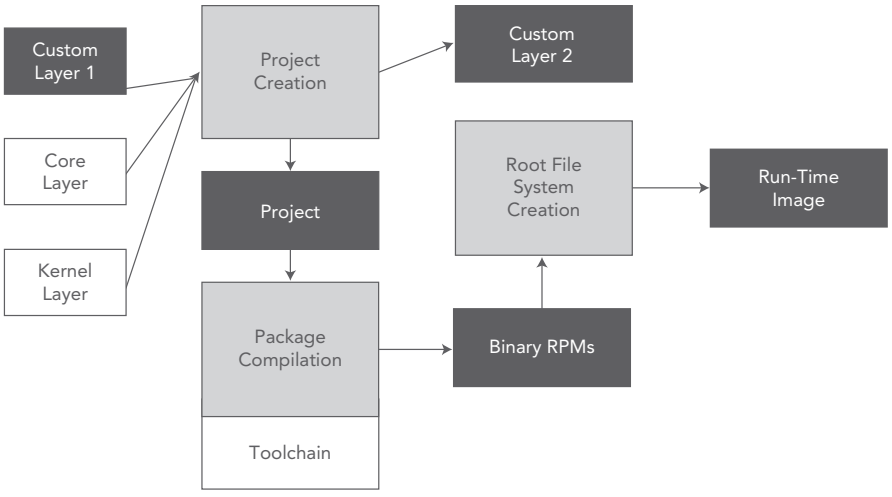

Figure 2: Typical use cases enabled with Wind River Linux

The configured project directory is itself a layer that can provide modified versions of packages or tools, change configuration information, and include other layers. The project directory can resynchronize these external layers. This allows for the user to include software being actively developed externally.

Development

Tools such as Wind River Workbench can help you add packages, make changes, debug, and compile software. LDAT creates file systems in a multitude of configurable ways. An original equipment manufacturer (OEM), a subsequent integrator or developer, or even final users may want to modify the file system composition. To this end, all the user space components, regardless of origin, open source trees, source RPMs, or customer trees, are packaged as binary RPMs and can be added or deleted or updated at will using the target-bound RPM binaries, which also can be included in the file system.

Users compile packages from source packages to binary RPMs. This uses the Wind River toolchain. The LDAT cross-build environment uses a simple front-end makefile fragment and, for SRPM packages, a modified spec file to provide the information required to make the package conform to LDAT and make it cross-buildable. If source code changes to the package are also required, patches are either applied once the package is extracted or during the normal prep stage of the RPM build. Patches (and the spec file) can be provided with the package in the layer and through templates included in the configuration, if configuration-specific modifications are required. Most open source packages can be imported with minimal changes and are immediately buildable with LDAT.

Both package formats support a pristine source model. This means the original upstream open source code is provided plus incremental patches that each implement one major feature. This allows for maximum transparency of source code.

Configuration Checksums

LDAT understands dependencies so that packages are rebuilt when their dependents are changed. To determine whether a package is valid for the current configuration, LDAT uses a checksum value calculated for the current configuration and compares it with the version stored in the RPM. The configuration checksum includes the sources and patches for the package or tool, the LDAT makefile fragment for the package, any additional configuration files, toolchain flags, and so on, along with the configuration checksums of any other package this package depends on. When the checksum doesn't match, the prebuilt version is ignored and the package will be rebuilt (as will any packages that depend on it).

To build packages, the build system calls the cross-toolchains with minimal performance overhead. The ability to use prebuilt versions of most or all host tools and target packages with checksums to ensure they are valid, along with the ability to update the build configuration with layer changes and to update the target file system without reassembling it from scratch, provides quick turnaround for development builds.

Package building also generates a sysroot to ensure that the right development libraries are being used. This can be exported to other users.

File System Creation

Users create a tuned root file system and run-time image. Once all of the packages are built, the target file system is assembled from the RPMs. If the file system has previously been assembled in this build directory, it can either be rebuilt from scratch or updated in place by uninstalling and reinstalling RPMs that have been changed, added, or removed.

Footprint optimization tools are included to reduce resources.

Layer Export

Users can create a custom layer for reuse by other development groups or other projects. After the system is built, the make export-layer command can be used to generate a layer that incorporates any local changes to the configuration, packages, or tools. The make install-prebuilt target can be used to place prebuilt copies of the host tools, target packages, and kernel into either a new or existing layer for use by subsequent builds.

Pseudo

Pseudo is a tool that ships with Wind River Linux, providing limited emulation of root access, preserving user ownership, modes, and device nodes correctly for the eventual target file system. The Pseudo tool allows package bills and file system creation without root privileges.

QEMU

Wind River Linux ships with QEMU, a hardware simulator. QEMU simulates ARM, PowerPC, x86, and MIPS hardware. This is integrated tightly with LDAT so that launching a test version in simulated hardware is trivial.

Layers

LDAT uses a hierarchical "layer" structure where each layer may provide anything from a simple configuration setting or a single package to something more complicated, such as completely replacing the kernel or augmenting toolchains, host tools, and so on.

The layers provided with Wind River Linux include the "core" layer (called wrll- wrlinux), which supplies the user space packages and core configuration information, the kernel layer for the kernel sources, a host-tools layer for tools that run on the build host, and a toolchain layer that contains the toolchains for cross-development. Add-ons for ecosystem products are typically provided as layers that may augment or modify the default configuration. User-defined layers may provide local changes and prebuilt binaries to speed development.

Server Install

A possible output of LDAT is a bootable DVD image to boot an x86 machine that will then launch an interactive installer. The user can then use this tool to install the generated run-time image on that machine's hard drive.

Compiler on the Target

The entire toolchain can be used on x86 targets to allow for native-mode compilation of applications.

Toolchain

The Wind River Linux toolchain is a customized distribution of Sourcery G++ version 4.4. This toolchain offers a number of additional features and modifications from the upstream projects, making exact version numbers noncomparable, but the core utilities are gcc 4.4, binutils 2.19, and gdb 7.0. Additional support for specific embedded processors and extended functionality are included in the toolchain.

The toolchain is maintained by Code Sourcery, providing support from some of the major contributors to the toolchain components. This expertise provides confidence that bugs will be fixed quickly and correctly. New functionality and bug fixes developed for this toolchain are contributed to the upstream projects whenever possible, as part of Wind River's commitment to the open source community.

The Wind River Linux toolchain supports ARM, x86, MIPS, and PowerPC targets.

C Library

To enhance support for embedded systems in the C library, Wind River has collaborated with other key vendors to found eGlibc, an embedded-oriented distribution of the GNU C library, and continues to sponsor the eGlibc project. The eGlibc project provides superior support for non-x86 architectures such as ARM, PowerPC, and MIPS and has a number of features to make it easier to build with a smaller footprint for target systems. Wind River also has uClibc support, suitable for small footprint targets.

Multilib Support

The Wind River Linux build system supports simultaneous use of two different CPU types (within the same general family) on a single target; for instance, a target could be configured with both 32-bit and 64-bit binaries, allowing per-package choices of space/speed trade-offs.

Prebuilt Multilibs

The Wind River Linux toolchain provides prebuilt library components, including the C library, for a large variety of multilibs. Each prebuilt library is fully validated as part of the validation process. Prebuilt libraries reduce compilation time while providing extra security and certainty. For each multilib, the toolchain contains configuration files to set up the compiler and other tools to produce compatible code for that multilib. Additional CPU-specific optimizations are available for a broad range of CPUs and configurations.

Toolchain Wrappers

Wind River Linux uses toolchain wrapper scripts that simplify cross-development. The project configuration process sets up toolchain wrappers for each CPU type used in the project. These wrappers combine target sysroot configuration and toolchain compilation options to provide seamless building for a target CPU. Users can then use the wrapper program as a substitute for GCC rather than trying to embed needed compiler options in package build processes. The preconfigured CPU templates provide the right combination of options to get good results from each CPU.

Toolchain Building

The Wind River Linux toolchain is distributed and fully supported as a collection of prebuilt binaries. However, source for every toolchain component is provided, along with configuration data and build scripts to rebuild the toolchain completely from source. Each toolchain component is distributed as a combination of a specific upstream source release or snapshot plus any additional patches provided by Code Sourcery or Wind River. The toolchain build process automates the task of bootstrapping a new cross-compile toolchain, reducing hundreds of commands and configuration steps to a simple "make toolchain."

Toolchain Export

When a toolchain has been configured for a project on a Linux host, it is possible to export that toolchain as an archive that can be used as a toolchain on another host. This can help with distributed development environments.

Run-Time Features

The following are features of the run-time image that runs on the target.

Kernel

The Linux 2.6.34 kernel forms the basis of the Wind River Linux kernel. Wind River adds many features and bug fixes to this kernel, and this specific kernel source configuration is tested and supported. The Wind River kernel adds to the kernel.org 2.6.34.x baseline by importing and validating changes from the following categories:

- **Mainline:** The feature set of the Linux kernel from kernel.org, with extended or validated features in particular configurations and applications
- **External:** Features imported from another external source and merged into the Wind River kernel
- **Internal:** Features that are in layers or merged into the kernel, developed by Wind River
- **Fixed:** Bug fixes for drivers; features in mainline or external projects

These kernel features are tested individually, merged, and then tested as a complete system. This includes stress and use case testing and ensures that the features are stable individually, integrate with Wind River tools, and form a solid base for deployment tuning.

The Wind River kernel is presented to the developer through a fully patched, history-clean Git repository. This stores the selected features, board support, and configurations extensively tested by Wind River. Presenting the Wind River kernel in this manner allows the end user

to leverage community best practices to seamlessly manage the development, build, and debug cycles.

From this Git tree, the build system generates a flat tree that contains the specific features required for the target kernel's use. Storing the source code in Git enables users to more easily understand what changes have been made to the kernel and why. Wind River uses a combination of tags and branches to assist in delineating between the various added features.

The workflow of the Wind River kernel follows the recognized community best practices. In particular, the kernel as shipped with the product should be considered an "upstream source" and viewed as a series of historical/documented modifications (commits) to the kernel. These modifications to the kernel represent the development and stabilization done by the Wind River kernel development teams.

Contact Wind River for more information about kernel development workflow with Wind River Linux.

Browsing

The Wind River Linux kernel development methodology simplifies the following use cases for browsing and understanding kernel code:

- Showing changes, e.g., "What changes were made to foo.c?"
- Showing foo.diff
- Showing groups of changes, e.g., "Show me only the LTTng patches."
- Comparing branches, e.g., "What's different between the ixm27 and imx31 BSPs?"
- Completing annotation for all changes, e.g., "Where did feature X come from and why is it there?"
- Showing standard commit IDs, e.g., "I see a kernel change on another tree; is this included in my tree?"

Kernel Features

The kernel forms the basis of many features in the Wind River Linux run-time. The following kernel features are available:

- **kexec:** A system call that provides the ability to shut down the current kernel and start another without rebooting hardware
- **kdump:** Kernel crash dump
- **ramconsole:** Early boot message logging into RAM small/embedded configurations and modifications for footprint reduction and analysis

Kernel Styles

The code base of the Wind River Linux kernel supports many features that are available for specific applications but not necessarily suitable for all. Wind River provides predefined kernel styles that are specific to these applications. The kernel styles shipped with the product are described here:

- **standard:** The standard kernel represents a common feature and technology base for all other kernel types. It is Wind River's goal to include all possible features in this single kernel type, making it suitable for many applications and a jumping-off point for more specific kernel implementations or the application of customizations such as embedded templates, or optional features. When it isn't possible for a feature to coexist at either compile time or run-time, it is merged into a specific kernel type.
- **cgl:** This kernel is targeted at networking equipment. This kernel includes upgrades such as shelf management, security, fault tolerance, threaded interrupt request lines (IRQs), and crash analysis that are not available in the other kernel types. This kernel is a suitable jumping-off point for high availability solutions.
- **preempt_rt:** The preempt_rt or "real-time preemption" kernel implements real-time capabilities for specific hardware.

File Systems

The following are kernel file system features in Wind River Linux:

- Boot technologies, i.e., ramdisk, execute in place (XIP), kernel libc support for boot environments (klibc), initial ramfs support (initramfs), fastboot (asynchronous boot/init)
- Flash file systems, i.e., yaffs, yaffs2, jffs, Advanced XIP file system (axfs)
- Logical Volume Manager (LVM and LVM2)
- RAID
- Network file systems, i.e., NFS, SMB, CIFS
- Disk file systems, i.e., ext2, ext3, ext4
- Other file systems, i.e., stackable unification file system (unionfs), file system for large device scalability (logfs), compressed read-only file system (squashfs), compressed ROM file system (cramfs)
- Persistent file systems, i.e., pramfs, pmemfs
- Revoke, i.e., revokeat() system call for inode-based revocation

Input and Output

The following are input and output (I/O) features of the kernel:

- **I/O splice:** A system call that copies data between a file descriptor and a pipe, or between a pipe and user space, without actually copying the data
- **User space I/O:** Drivers that allow programs easy access to kernel interrupts and memory locations; used for user mode drivers
- **eventfd:** An event wait/notify mechanism used by user space applications and the kernel to notify user space applications of events, enhanced to efficiently deliver interrupts to KVM guests via an ioeventfd
- **epoll:** A variant of poll(2) that can be used either as an edge- or a level-triggered interface; scales well to large numbers of watched fds

Security

Wind River Linux includes the following features to improve system security:

- **BSDJail (bsdjail):** A facility for creating chroot jails hardened against attack vectors that defeat common chroot protections
- **Simplified Mandatory Access Control Kernel (SMACK):** A small, fast, name-based mandatory access control mechanism focusing on the Bell-La Padula security model
- **Integrity Measurement Architecture:** Support for run-time validation of executables and critical system files optionally using a hardware Trusted Platform Module
- **Security Enhanced Linux (SELinux):** A full-featured Linux Security Module providing a reference monitor capable of providing multilayer security and multi-category security, implementing both mandatory access control (MAC) and role-based access control (RBAC)
- **cgroups and controllers:** Support for grouping sets of processes together, for use with process control subsystems such as Cpusets, CFS, memory controls, and device isolation; includes net traffic controller, memrlimit controller, dm-ioband bio_tracking, and group scheduling controllers
- **OCF (Open Cryptographic Framework):** Hardware-accelerated cryptographic support for specific BSPs

Debugging and Profiling

The following are debugging features of Wind River Linux:

- **oprofile:** Kernel.org oprofile enhanced with tracing through the syscall boundary
- **ftrace:** Lightweight function tracing, with dynamic ftrace and early-ftrace for boot-time measurement enhancements; also provides trace-cmd for direct access to options and tracers
- **ptrace:** Process trace, single-step, multithreaded trace support
- **kprobes:** Kernel address trapping
- **KGDB:** Kernel debug support over serial, Ethernet, and console
- **lockdep:** Lock dependency checking and analysis
- **KDB (all architectures):** Kernel debugger with kernel mode switching, allowing for debugging the kernel you are running on, thus not requiring an additional machine
- **PowerTOP:** Location of programs that are misbehaving while your computer is idle
- **perf:** Rich abstractions over hardware performance monitoring capabilities; per-task, per-CPU, and per-workload counters and counter groups and sampling capabilities; extensions for support of MIPS and PowerPC architectures
- **kmemcheck:** Kernel memory checking and leak detection
- **wrnote:** ELF image annotation for core dump debug
- **On-chip debugging:** Support for Wind River on-chip debugging devices
- **Linux Trace Toolkit (LTTng):** Extensible, lightweight kernel instrumentation for tracing program execution and debugging parallel and real-time behavior
- **latencyTOP:** Latency visualization support
- **Boot-time reduction:** Enhancements for measuring and streamlining boot time
- **Footprint reduction:** Kernel configuration and modifications to limit the run-time kernel footprint

Real-Time and Deterministic Scheduling Behavior

The following are specific real-time and deterministic scheduling behavior features:

- preempt_rt, which reduces the scheduling latency of the kernel by replacing almost every spinlock used by the kernel with preemptible mutexes, making all but the most critical kernel code involuntarily preemptible
- Voluntary kernel preemption (desktop), which reduces kernel latency by adding more explicit preemption points to kernel code
- No forced preemption (none), the traditional Linux preemption model
- Robust priority inheritance mutex, inheritance support for user space mutexes
- High resolution timers (HRT)
- Dynamic tick support (NOHZ), which only triggers timer interrupts on an as-needed basis, when the system is busy and when it is idle
- CPU isolation for dedicated processing or use by Wind River Hypervisor
- Optional/experimental support for BFS and EDF scheduling on x86 targets

Wind River Workbench and Development Tools

The Eclipse-based Wind River Workbench development suite offers deep capability throughout the development process in a single integrated environment, with complete platform integration and tools for debugging, code analysis, advanced visualization, root-cause analysis, and test.

Analysis Tools

Workbench and Wind River Linux make a number of analysis tools available to the developer. Some are enhanced versions of open source tools related to profiling and memory usage, and some are specifically developed by Wind River:

- **Performance analysis:** Wind River Workbench Performance Profiler analyzes how a CPU is spending its cycles by providing a detailed function-by-function analysis that shows how individual routines within the processes consume those cycles. This feature is based on the open source tool oprofile, with additional visualization and integration in Workbench.
- **Memory analysis:** Wind River Workbench Memory Analyzer is a dynamic memory analysis tool that helps prevent and fix such problems as memory leaks, excessive number of memory allocations, and excessive memory allocation sizes. Memory Analyzer uses the open source tool mpatrol, with additional visualization in Workbench.
- **Boot-time analysis:** This uses the ftrace tool to provide lightweight function tracing and includes dynamic ftrace and early-ftrace for boot-time analysis.
- **Code execution coverage:** The code coverage analyzer feature of Wind River Workbench determines the percentage of source code executed by your software test case and points to the sections of code that have not been fully tested.
- **Valgrind:** By running an application in a virtual machine, valgrind tracks memory management problems and threading bugs.
- **LTTng:** This tool provides tracing capabilities for both kernel and user space.

Package Management

Wind River provides several tools to examine the file system's package list, examine package-level dependencies, perform safe package addition or removal based on those dependencies, and perform file-level examination and control of the file system contents:

- **Package lists and snapshots:** The user space file system is built up from discrete packages, from open source, user source, and virtual packages from custom content. Workbench provides a way to control that package list, to explore different package combinations, and to preserve safe combinations.
- **Dependency tracking:** Workbench allows the user to visualize the (deep) forward and reverse dependencies and to add or remove packages, knowing that the dependencies are reported and managed.
- **Direct package updates to target:** Workbench facilitates RPM management on the target as on a regular Linux host. Packages can be developed and compiled and then pushed to the running target for fast turnaround debugging using incremental updates.

Development Tools

Wind River provides several tools to examine and directly control the file system content below the package level. There are also tools to import new open source packages, import new patches, and directly examine a package's patch tree:

- **File system layout:** The user can directly see the final file system content and directly remove or add files at a fine-grain level below the coarser package-level dependencies, allowing direct control of the file system footprint. Wind River also provides tools to discover and visualize which files are touched during a target run.
- **Disk space allocation tool:** The user can understand what and how much space directories are consuming.
- **Package import tools:** Workbench has a feature to handle most of the initial package importation and cross-compilation setup, to help speed up the adoption of new open source packages into a user's project.
- **Patch import and export tools:** Workbench enables the user to view the patch tree directly and patch files for both source RPM and regular packages. It also provides tools to help import and resolve new patches and to export user changes as new portable patches into a layer directory.

Export Layer

The export layer feature will automatically export changes made in a project into a new portable Wind River Linux layer. This includes package list changes, file system trimming, kernel configure changes, and new local packages additions:

- Import source trees from within a Wind River Linux layer for compile/edit/debug.
- Interact with source code held in a software configuration management (SCM) system.
- Export Workbench projects into Wind River Linux packages.

Hardware Support

The following are highlights of hardware features:

- Multi-architecture support: MIPS, MIPS64, ARM, x86, x86-64, PowerPC
- SMP/AMP/multi-core
- CPU isolation (cpuisol)
- CPU hotplug
- IEEE floating point conformance for PowerPC processors supporting signal processing extensions (SPE)
- Talitos Freescale security engine, hardware acceleration for PowerQuicc E processors
- Device drivers for peripherals such as audio, Ethernet, GPIO, SDIO, SCSI, MTD, serial, framebuffer, VGA (graphics), keyboard, USB (gadget, host, OTG), touch screen, PATA/SATA, sound, PMEM, wireless (Wi-Fi), Bluetooth, MTP
- EDAC (error detection and correction), a set of Linux kernel modules for handling hardware-related errors
- Integration of upstream kernel architecture trees, such as linux-omap and linux-davinci staging

Virtualization

Wind River Linux 4 offers a choice of open source and commercial virtualization solutions:

- Wind River Hypervisor, which enables x86 and e500/e500mc-based guest kernels
- KVM (host/guest), on selected x86-64 platforms
- virtio support
- Containers/cgroups, i.e., mainline + blkio, dm-ioband, net_traffic controllers for container-based resource management

Power Management

Wind River Linux 4 offers ways to analyze and reduce power consumption:

- PowerTOP for x86 provides the capability to analyze power consumption on x86 processors.
- Kernel additions to Advanced Configuration and Power Interface (ACPI) provides the capability to control power consumption in systems that support ACPI.
- Tickless kernel allows lower power consumption by eliminating clock interrupts.
- Low-power hypervisor allows virtualization to be used in low-power configurations.

Networking Features

Wind River Linux 4 includes features that directly support networking applications.

System Black Box

Taking cues from the aviation industry, the persistent memory framework (PMEM) of Wind River Linux provides a system black box acting much like the combined flight data recorder and cockpit voice recorder. Scheduler decision history, logs of all exceptions, panic and console logs, kernel log messages, system reset and reboot logging, Linux Trace Toolkit (a set of kernel patches and supporting user space tools to control tracing) logs, even end-user defined events can all be logged to dedicated nonvolatile memory, external memory, peripherals, or even protected segments of normal system RAM. This enables faster recovery and better system uptimes by allowing all necessary debug information to be preserved by the PMEM driver in these protected regions of memory for later analysis while allowing the system to reboot and re-enter service immediately.

Transparent Inter-process Communication Protocol

As a major contributor and one of the maintainers of the TIPC project, Wind River actively develops this cross-platform, high-speed communications technology aimed specifically at clustered computing environments. TIPC is a communications protocol that provides developers with an extremely flexible means of creating distributed, cooperative applications that may migrate as required throughout the cluster seamlessly. Wind River continues to invest in TIPC, and Wind River Linux remains up-to-date with developments in the TIPC project.

Security

Originally developed by the National Security Agency (NSA), SELinux is the gold standard of flexible and trusted computing environments. SELinux is both a Linux Security Module (LSM)—a piece of the kernel that arbitrates access to all system resources based on the system policy—and a collection of supporting user space tools for developing, applying, enforcing, auditing, and debugging the security policy used by the LSM. Wind River Linux includes three levels of security out of the box for SELinux-enabled configurations based on the SELinux Reference Policy Project and configured specifically for Wind River Linux.

Wind River Linux also includes advanced, preemptive security technologies such as multilevel run-time stack and buffer overflow protection and a suite of tools that together provide a complete intrusion detection and prevention system.

The PaX patch set adds dynamically configurable, least-privilege protections for memory pages and segments as well as many system hardening features such as address space layout randomization (ALSR). Complemented by GRSecurity, further kernel patches that build upon PaX and implement a trusted execution model, role-based access control, detailed system accounting logs, and fine-grained privilege separation, the effect is a second highly secure reference monitor providing a complete security system.

Linux Standard Base 4.0

Linux Standard Base (LSB) is a set of standards for Linux distributions and applications aimed at providing cross-platform compatibility between LSB-compliant Linux distributions and applications. It is supported on x86, Power PC, and MIPS architectures. Wind River Linux supports all required user space functionality to be LSB certifiable with the Linux Foundation.

Reliability, Availability, Manageability

The first certified Carrier Grade Linux 4 distribution, Wind River Linux continues to address the needs of network equipment providers and has been designed to meet the upcoming Carrier Grade Linux 5.0 specification. Wind River Linux 4 continues to add support for faster, more reliable systems and advanced cluster management technologies essential to meeting high availability requirements.

Wind River Linux also meets the SCOPE Alliance Linux profile and addresses key SCOPE Alliance gaps. These include persistent shared memory with the system black box, coherent user and kernel tracing framework with LTTng, run-time analysis tools, and common command-line tools such as strace and ltrace for doing system call and library tracing.

Features included in Wind River Linux that go beyond the existing Carrier Grade Linux requirements include the following:

- Coarse resource enforcement that allows memory and scheduling limits to be enforced on a group basis rather than per-process or per-object
- Layer 2 Tunneling Protocol (L2TP version 3) support
- File access tracing that provides extensive logging and notification options for monitoring file access and recording system events
- Redundant virtual routing support, based on the Common Address Redundancy Protocol, that provides the ability to create highly available dynamic routers and gateways
- Feature-rich IPsec environment ranging from simple certificate creation and management to cryptographic reliability and integrity self-tests
- Dynamic multi-category security management tools for virtual guests through libvirt and the SELinux svirt tools
- Support for advanced system architectures that leverage memory and processor latency domains with tools such as numactl and libnuma that enable application and system designers to get the absolute best performance out of their real and virtual machines
- Reliable damage-resistant file system support, detailed in the new CGL 5.0 specification, including data and metadata integrity checking mechanisms

With support for both standalone and clustered systems as well as the PICMG Advanced Telecommunications Computing Architecture (ATCA) specification, Wind River Linux provides highly available solutions for devices at both the network core and edge.

Network-Based Storage Solutions

Wind River Linux provides functionality for centralized logging servers, centralized billing and accounting servers, and share file system servers by integrating technologies such as the distributed replicated block device (DRBD), multiple redundant communication paths to external storage over fiber channel links, ATA over Ethernet, the Oracle Cluster File System version 2 (OCFS2), and Internet Small Computer System Interface (iSCSI).

Board Support Packages

Wind River Linux board support packages (BSPs) are hardware-enablement components that contain elements such as drivers and settings needed to make Wind River Linux support specific hardware.

BSPs are separable configuration components that can be created and added to Wind River Linux at any time. In addition to the BSPs Wind River Linux ships with, boards are added continually according to customer demand and hardware availability. Such additional BSPs are available through Wind River Online Support to customers under an active platform subscription. Also, Wind River Services can create customer-specific BSPs for hardware that is not covered.

A typical BSP includes board-specific configuration files that overwrite or add configuration options defined by the common platform templates. Additional kernel patches included in the BSP can add new device drivers or apply necessary changes to existing Linux code. BSPs can also contain additional user space components or other files.

Wind River has validated proper operation of the Linux run-time for each supported reference board. The supported features are board-specific and depend on availability and maturity of the code in the open source community.

Wind River Linux ships with more than 40 BSPs covering ARM, Intel, x86, MIPS, and PowerPC target processors.

BSPs are also created and shipped asynchronously, after the product is released. Contact Wind River to get an up-to-date supported BSP list with detailed descriptions of supported peripherals.

Applications

Wind River Linux provides more than 550 integrated user space application packages. They implement functionality typical of an embedded Linux run-time. The Wind River build system (LDAT) generates binary RPMs from these sources. LDAT can then use these to generate a root file system.

Origins and Porting

A variety of open source projects forms the origins of the user space code base. About 150 packages are based on traditionally prepackaged trees containing source code, configuration scripts, and makefiles or makefile precursors (i.e., a classic package format). The remaining 400 have source RPMs as their base.

Wind River patches these upstream sources for integration and bug fixing. These packages generally contain the following types of patches:

- **Cross compilation:** Many packages are expected to be compiled on x86 architectures for x86 architectures. This often means host libraries can be referenced or linked in.
- **Multilib:** This ensures that packages can be built for both 32- and 64-bit targets.
- **Other defects:** Wind River ensures that packages are properly integrated together.

The Wind River Linux Distribution Assembly Tool (LDAT) will access the ported components, pass the appropriate cross-compilation parameters, and create a file system matching the target's architecture and the kernel's configuration. Customers can add their own user space packages using the LDAT tool. Instructions for this are included in the product documentation.

Package Categories

Wind River Linux 4 comprises 550 software packages, which can be grouped in the following categories. A full list of packages can be found at the end of this document in "Appendix A: Package Summary by Category." For package details (source package names, binary package names, versions, licenses, etc.), contact Wind River.

It is easiest to consider the package list in terms of categories used:

- Administration
- Base libraries
- Basic C support
- Booting and startup
- Daemons
- Databases
- Debugging
- Development tools
- Devices
- DirectFB
- Documentation and printing
- File systems
- File transforms
- Graphics
- Hardware
- High availability
- Host tools
- Kernel
- Middleware
- Multimedia
- Networking
- Network daemons
- Scripting languages
- Security
- SELinux
- Setup
- Shells
- Sound
- System
- System analysis
- Test
- Utilities
- Virtualization
- Wind River instrumentation
- X Server

Profiles

Wind River Linux ships with several profiles that define preassembled root file systems and kernels for industry-specific functionality as listed previously in the "Key Components" section.

Testing and Validation

Wind River is committed to providing quality products for both proprietary and open-source-based technologies. Our quality policies include formal code inspections, peer reviews, project reviews, program audits, and traceable requirements change management. Wind River Linux was created following a methodical process to thoroughly test key features on every supported reference configuration (defined by development host, kernel and package configurations, and supported board).

Wind River has developed a robust, scalable, and automated build and test infrastructure with more than 4,000 test cases and 301,336 test runs. This infrastructure supports many processor architectures and uses a combination of commercial, open source, and proprietary tests, including LTP Core, LTP Network, LSB, TAHI, and Open POSIX. Wind River uses coverage tools, such as gcov and lcov, to optimize test development and close gaps in existing test suites.

Automated and manual tests of Wind River Linux 4 include the following:

- **Automated boot login:** This tests the booting process of any target architecture for a given kernel and rootfs. The process is completely automated for a set of targets, which helps in determining the boot sanity of the target.
- **CD sanity:** This automation suite covers CD installation on a new release, followed by building the rootfs for various target combinations using prebuilt RPMs. It boots the target with the prebuilt kernel and rootfs and executes KGDB and user-mode tests on the target, then reports the results to the database.
- **Linux Test Project (LTP):** This test suite validates the reliability, robustness, and stability of Linux kernel and its network components.
- **Open HPI:** The Open Hardware Platform Interface is an abstracted interface for managing computer hardware, typically chassis and rack-based servers. HPI includes resource modeling; access to and control over sensor, control, watchdog, and inventory data associated with resources; abstracted System Event Log interfaces; hardware events and alarms; and a managed hotswap interface.
- **Open POSIX:** This test suite is for POSIX 2001 APIs not tied to specific implementations. It provides conformance, functional, and stress testing, with an initial focus on threads, clocks and timers, signals, message queues, and semaphores.
- **Real-Time Feature Testing with Lmbench Realfeel:** Real-Time Feature Testing tests performance. Lmbench is used to measure I/O of the kernel. Realfeel tests scheduler behavior.
- **Saftest:** This tests the Open HPI package.
- **Kernel Feature Test Package (kftp):** This tests kernel features against architecture and hardware configurations.
- **User space:** User space tests verify that user packages have compiled and operated correctly in target run-time systems.
- **LDAT:** This tests build system functionality.
- **Toolchain:** This verifies that toolchains comply and operate correctly in target run-time systems.
- **nmap:** This is a network exploration tool and security scanner.
- **Netperf:** This tool tests networking performance.
- **CGL compliance:** This tests more than 120 P1 requirements.
- **kexec and core-dump, server-based install:** This tests CGL features.
- **TAHI IPv6 conformance:** The TAHI Project is a joint effort to develop and provide verification technology for IPv6. The TAHI IPv6 conformance tests ensure interoperability with IPv6 protocols.
- **iozone:** This is a file system performance test tool.
- **nbench byte:** This tests CPU calculation performance.
- **Footprint:** This tests the static kernel footprint.
- **Static rootfs footprint:** This tests the static rootfs footprint size.
- **Dynamic footprint:** This tests the run-time system footprint.
- **Boot-time:** This measures board boot-up time.
- **Coverity:** The system is tested using Coverity static analysis software.
- **Board-specific:** Along with new BSPs added for release, there are board-specific tests mentioned in the board readme. These tests were executed as part of regression testing on Wind River Linux 4.
- **Documentation:** Documentation for Wind River Linux 4.0 was tested to make sure all steps are properly recorded.
- **Host OS:** Installation is tested on various host OSes supported for Wind River Linux 4 as well as sample application build and debugging from Workbench and platform build.
- **HRT regression:** HRT features for previous releases were tested for regression on the supported platforms.
- **Install bundle:** Installation testing was done for various product structure bundles.
- **RT regression:** Regression testing features in previous releases were tested for regression on the supported platforms.
- **Usability:** Wind River Linux 4 and Workbench 3.2 usability testing is based on the usability testing document.
- **Use case:** The use cases for Workbench, run-time analysis tools, and build system were tested on supported hosts and platforms.
- **Workbench integration:** Wind River tests the feature integration of Workbench with Wind Manage, System Viewer, and run-time analysis tools.
- **Wind River Linux 4 bug fix:** The bugs fixed during various release cycles were tested for the fixes then closed.

Optional Add-on Products

Wind River provides other products to implement functionality not available in the base product.

IPL Cantata++ for Wind River Workbench

IPL Cantata++ for Wind River Workbench (formerly Unit Tester), now available for Wind River Linux, 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 the Wind River Workbench development suite 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.

Wind River Workbench On-Chip Debugging

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 provides connectivity between the host development environment and the target device via the JTAG or on-chip debugging interface of the microprocessor that resides 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 regarding the exact code that ran on the target and when.

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

generation. Specifically for Linux development, Workbench On-Chip Debugging provides visibility into hardware and software interactions for kernel and kernel modules and enables development and debug of user space applications. The JTAG-based debug capability is a useful alternative to agent-based debugging in applications where serial, Ethernet, or USB interfaces are not available or in environments where agent instrumentation of the operating system is not desired.

The Wind River Debugger provided with Wind River platforms can be enabled for on-chip debugging. This capability, along with Wind River ICE, Wind River Trace, and Wind River Probe hardware, provides access to significant additional capability within Workbench.

For more information, visit http://www.windriver.com/products/workbench.

Wind River Network Acceleration Platform

Wind River Network Acceleration Platform is a packet processing solution that leverages multi-core technology to deliver Gigabit Ethernet wire-speed performance for ultra-fast IP packet forwarding for network infrastructure equipment. The platform is a comprehensive bundle of Wind River's industry-leading multi-core-ready run-time technologies. It comes standard with control plane operating system support—Wind River Linux or VxWorks—and data plane acceleration software, including a fast real-time bare metal executive and high-performance networking software. The platform also includes a lightweight hypervisor to load and configure individual cores and to provide abstraction of system resources.

The networking-specific software stack includes IPv4/IPv6 packet forwarding protocols optimized for specific multi-core processors. Network Acceleration Platform is designed to provide linear performance scalability. As more cores are dedicated to the data plane, packet throughput performance increases proportionately. This is an important contrast to implementations that are limited by bus, memory, or other resource constraints. While symmetric multiprocessing (SMP) plays an important role in multi-core systems, the packet processing performance curve in an SMP configuration can flatten after only a few cores, yielding diminishing returns as more system resources are allocated to networking tasks.

Wind River's asymmetric multiprocessing (AMP) technologies provide a clean separation of control plane and data plane functions, which enables greater efficiency of multiple processing cores. The data plane cores provide excellent scalability while the control plane is freed from the burden of packet processing. The platform offers flexibility to configure the allocation of cores to the control or data planes to meet a wide range of networking applications.

Wind River Simics

Wind River Simics creates a high-performance virtual environment in which any electronic system—from a single board to complex, heterogeneous, multi-board, multi-processor, multi-core systems—can be defined, developed, tested and deployed. Wind River Simics removes hardware dependencies that slow product prototyping, facilitates hardware and software co-development, and makes it possible to test early and test often, improving product quality and eliminating late-in-the-game integration testing. Teams using Simics experience dramatic cost savings throughout the development life cycle, reach market 18 months faster, cut a year's time from ecosystem enablement, and produce higher-quality products. Wind River Simics speeds and simplifies development on cutting-edge multi-core hardware.

For more information, visit http://www.windriver.com/products/simics/.

Wind River Hypervisor

Facilitating efficient adoption of multi-core processors, Wind River Hypervisor brings a new level of flexibility to the development of embedded devices. It opens up new opportunities for cost savings through hardware consolidation; it allows developers to leverage multiple operating systems in a single device so they can expand and enhance device functionality; it facilitates the adoption of multi-core processors by increasing reliability and reducing risk; and it provides the new software configuration options required to architect next-generation embedded devices.

An integral part of Wind River's multi-core software solution, Wind River Hypervisor focuses on core real-time values such as high performance, small footprint, determinism, low latency, and high reliability. It is highly optimized for and integrated with VxWorks and Wind River Linux, and it supports other operating systems and a broad range of processor architectures.

Using Wind River Hypervisor in a supervised AMP configuration makes systems easier to configure, reduces complexity by increasing protection between cores, and provides a more scalable and reliable solution.

For developers who are building highly differentiated embedded devices, Wind River Hypervisor offers compelling new capabilities. Now you can build more functionality into smaller form factors, build more scalable and reliable multi-core systems, and consolidate with confidence—with all the efficiencies of a unified development environment and a single-vendor solution.

For more information, visit http://www.windriver.com/products/hypervisor/.

Wind River Tilcon Graphics Suite

Wind River Tilcon Graphics Suite enables the development and deployment of rich user interfaces for multifunctional embedded devices, to create better end-user experiences. Its

unique codeless development architecture enables device software developers to easily produce, maintain, and rebrand user interfaces at a fraction of the cost of traditional graphics software. It boasts a complete set of tools including an interface builder, APIs for application integration, and a robust graphics run-time engine to render the graphic objects. Tilcon Graphics Suite is integrated with Wind River's VxWorks and Wind River Linux operating systems and backed by Wind River's global technical support, customer education, and professional services.

For more information, contact Wind River.

Wind River Test Management

Wind River Test Management is a scalable system that links device development and test teams with a collaborative suite of applications for efficient system testing and defect resolution. The system leverages a unique, dynamic instrumentation technology to measure code coverage, profile performance, and diagnose and repair the system at run-time. The product is designed to manage multiple devices under test at multiple lab locations, maximizing resource utilization and accelerating the testing process.

Benefits of Wind River Test Management 3.1 include the following:

- Higher quality, faster time-to-market, lower cost
- More testing, more often
- Faster defect resolution
- Management of progress, quality, and resources
- Benefit to both development and QA
- Powerful sensorpoint technology
- Open, scalable architecture
- Broad platform support

For more information, visit http://www.windriver.com/products/test_management/.

Legal Compliance

Wind River performs thorough legal reviews of the compilation and documentation of the General Public License (GPL) and other licenses that control each major release of Wind River Linux. Combining human legal expertise and proprietary automated tools, Wind River examines each open source package that comes into the product to identify and resolve potential intellectual property issues before the product is released. Customers receive extensive documentation to assist them in the protection of their intellectual property.

Wind River Linux Developer Community Website

For peer-to-peer discussion, problem solving, and sharing best practices, Wind River now hosts an online developer community at http://developer.windriver.com/. Visit to find use cases, current roadmap information, and other resources pertinent to Wind River Linux customers.

Partner Ecosystem

Wind River's world-class partner ecosystem ensures tight integration between our core technologies and those of the premier hardware and software companies we've chosen to build out our solutions. Our partners help extend the capabilities of Wind River Linux by offering out-of-the-box integration and support for key technologies in a number of fast-moving markets. Our team is trained to troubleshoot partner technologies in use with Wind River products, making ours the best-supported ecosystem in the embedded and mobile software industry.

The Wind River partner ecosystem is constantly expanding. Contact us for more details or visit http://www.windriver.com/partners/.

Open Source Community

Wind River has a long history of working with and contributing to the open source community. We were one of the early adopters of the GNU compiler and debugging technologies for use with our VxWorks real-time operating system and Tornado cross-development environment, and we contributed bug fixes and improvements to these projects.

We continue our commitment to enable our customers to successfully leverage open source in their development of next-generation devices through our active involvement in a number of organizations.

Carrier Grade Linux

The Linux Foundation CGL working group is developing the Carrier Grade Linux specification and driving the adoption of Linux in the network infrastructure industry.

Consumer Electronics Linux Forum

CELF drives adoption of Linux in the consumer electronics industry.

Eclipse Forum

The Eclipse Forum works to improve and enhance the core Eclipse framework and also enables Wind River Workbench to be integrated with a wide range of complementary Eclipse plug-ins for software development.

Open Handset Alliance

The Open Handset Alliance is a group of mobile and technology leaders who share the vision for building a better mobile phone and are committed to changing the mobile experience for consumers through the efforts of an open mobile ecosystem.

SCOPE Alliance

SCOPE is an industry alliance committed to accelerating the deployment of carrier grade base platforms for service provider applications.

Service Availability Forum

The SAF is an organization developing APIs and specifications for high availability of hardware and software applications. Its specifications include the Hardware Platform Interface (HPI) specification and the Application Interface Specification (AIS).

OpenSAF Foundation

OpenSAF is an open source project established to develop a base platform middleware consistent with Service Availability Forum (SA Forum) specifications, under the LGPLv2. license. The OpenSAF Foundation was established by leading communications and enterprise computing companies to facilitate the OpenSAF Project and to accelerate the adoption of the OpenSAF code base in commercial products.

TIPC Project

The TIPC Project provides an open solution for communication and messaging.

Professional Services

Wind River Professional Services, a CMMI Level 3–certified organization, enables you to reduce risk and focus on development activities that add value and differentiate design. As part of our comprehensive solutions, Wind River offers a Linux Services Practice, 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 your company's success. Backed by our commercial-grade project methodology, Wind River Professional Services include device design, Linux BSP and driver optimization, software system and middleware integration, and legacy application and infrastructure migration.

Education Services

Education is fundamentally connected not only to individual performance but also to the success of a project or an 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 fit your time, budget, and skills development requirements.

Personalized Learning Program

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

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

Support Services

Wind River Customer Support, a certified Service Capability and Performance (SCP) organization, provides support for Wind River Linux platforms. Your subscription to Wind River Linux includes full maintenance and support, delivered through Wind River's Online Support website and our worldwide support team. Wind River Support includes the development suite and cross-toolchain, Linux kernel, and the reference root file system, as validated on supported boards and development host operating systems. While under subscription, customers receive both maintenance updates and major upgrades.

Technical Support

Wind River works with every customer to help you solve technical support problems. We may not be able to support every configuration of hardware and software that a customer may have selected, but we will do everything we can to provide support. Linux Technical 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, we 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.

Customer Support will provide bug fixes following the process outlined in Wind River's Customer Support User's Guide (CSUG), available at http://www.windriver.com/support/ resources/csug.pdf.

If appropriate, Wind River will submit changes in open source code to the open source project maintainer for inclusion in a future release of the open

source package. Wind River will maintain changes until a new version from the open source project is available and can be released for Wind River Linux.

Customers with a valid support or subscription agreement are eligible for all respective updates 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.

Visit Wind River Online Support (OLS) 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. OLS also provides new BSPs, updates to existing packages, patches, manuals, the latest errata, and other announcements about Wind River Linux. Wind River will also provide new contributed Linux packages through our support website. These packages have been contributed by the open source community and are prebuilt and tested with Wind River Linux.

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.

North America, South America, 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. (PT)

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)

Appendix A: Package Summary by Category

Category	Packages
Administration	eel, evlog, fmconf, fmldc, libglade2, libutempter, memstat, monit, openais, quota
Basic C support	binutils, boost, glibc, glibc_localedef, libaio, libatomic_ops, libcap, libdrm, libgcc, libstdcxx, outo, prelink, uclibc, wrs_kernheaders
Base libraries	libevent, userspace-rcu, wrs-lsb
Booting and startup	bootcycle, bootpc, grub, silo, tftp-hpa, yaboot
Daemons	acpid, audit, crontabs, daemontools, esound, fam, iozone, iscsi-initiator-utils, mcelog, ORBit2, pcsc-lite, pulseaudio, quagga, samba, vixie-cron, vsftpd, xinetd
Databases	hwdata, libtermcap, mysql, openldap, postgresql, python-ldap, sqlite, unixODBC
Debugging	eventlog, gdb, gdbserver, logrotate, ltrace, makedumpfile, smartmontools, strace, sysklogd, syslog-ng, sysstat, tcf-agent, trace-cmd, watchdog, valgrind, valgrind-tests, wdbagent-ptrace
Development tools	autoconf, autoconf213, automake, automake14, automake15, automake16, automake17, bison, byacc, chrpath, git, libelf, nasm, patch, pkgconfig, quilt, sample_module, target_toolchain, tc_debug_src
Devices	ccid, device-mapper, device-mapper-multipath, eject, ethtool, ipmitool, kbd, libfakekey, libusb, lm_sensors, MAKEDEV, mingetty, minicom, nbd, openipmi, parted, pciutils, scsidev, setserial, udev, usbutils, vblade
DirectFB	dfbtutorials, directfb, directfb-examples, directfb_headers
Documentation and printing	cups, enscript, foomatic, ghostscript, groff, man, texinfo
File systems	acl, attr, dmapi, drbd-tools, e2fsprogs, fetch-footprint, filesystem, fuse, gphotofs, installsw, lsof, lvm2, mdadm, mobile-basic-flash, mtd-utils, mtools, rdist, rsync, samhain, xfsdump, xfsprogs, yaffs2
Graphics	atk, cairo, cairo, fbset, fontconfig, freetype, gail, galculator, glib2, gnome-desktop, gnome-icon-theme, gnome-keyring, gnome-menus, gnome-mime-data, gnome-panel, gnome-python2, gnome-vfs2, gtk, hicolor-icon-theme, libart-lgpl, libgnome, libgnomecanvas, libgnomeui, libgsf, libjpeg, libmatchbox, libpng, librsvg, libtiff, libvisual, libwnck, matchbox-keyboard, matchbox-window-manager, metacity, nautilus, ncurses, OMAP35x_Graphics_SDK, pango, pycairo, pygobject, pygtk, qt, qt3, qt-embedded, sdl, SDL_image, SDL_mixer, SDL_ttf, tslib
Hardware	edac-utils, hal, hal-info, liboil, libpciaccess
High availability	corosync, cluster-glue, heartbeat, pacemaker, resource-agents, ucarp
Host tools	chkconfig, db4, elfutils, expat, flex, libtool, libxml2, neon, paxctl, rpm
Kernel	intercept_proprietary, irqbalance, kexec-tools, libkcompat, linux_filter_proprietary, mac-fdisk, pth, small-module-init-tools, scripts, vmc-utils
Scripting languages	expect, lua, perl-libxml-perl, perl-TimeDate, perl-URI, microperl, pcre, perl, perl-Convert-ASN1, perl-LDAP, perl-XML-Parser, python-imaging, python, ruby, swig, tcl
Middleware	ace, dbus, dbus-glib, dbus-python, dsplink, libbonobo, libbonoboui, libIDL, libsoup, paste, startup-notification
Multimedia network daemons: freeradius, hostapd, ippool, openl2tp, openssh, openssh-sftp-only	cdrkit, cdparanoia, GConf2, gst-plugins-bad, gst-plugins-base, gst-plugins-farsight, gst-plugins-good, gst-plugins-ugly, gstreamer, ImageMagick, libcdio, libexif, libgphoto2, libgweather, libksba, libmms, libmtp, libogg, libtheora, python-gst, setmixer, sox, timidity, xulrunner
Networking	agent-proxy, aoetools, apache-ssl, atftp, bind, bluez-libs, boa, curl, dhcp, ecgl-tools, gtkhtml, ifenslave, inetutils, iproute, iptables, iputils, klibc, libnet, libnl, libpcap, librds, lksctp-tools, libssh2, lrzsz, mailx, midbrowser, mipv6-daemon-umip, netcat, net-snmp, net-tools, nfs-utils, nfs-utils-lib, nspr, ntop, portmap, ppp, pyca, radvd, rdate, rsh, sendmail, socat, tcpdump, telnet, tipc_demo, tipcutils, tnftp, traceroute, tunctl, usagi-tool, vlan, wget
Shells security: beecrypt, cracklib, crypto-tools, dropbear, ecryptfs-utils, gnupg2, gnutls, gradm, ipsec-test, ipsec-tools, keynote, keyutils, krb5, libassuan, libgcrypt, libgpg-error, librmisec, libsepol, logcheck, nss, openssl, pam, pam_passwdqc, passwd, racoon2, shadow-utils, strongswan, sudo, tcp_wrappers, xmlsec1	bash, busybox, findutils, gawk, grep, less, sed, tcsh, xerces
SELinux	libselinux, libsemanage, mcstrans, policycoreutils, pyetree, refpolicy, refpolicy-strict, sepolgen, setools
Setup	initscripts, libuser, linux, mm, module-init-tools, procps, psmisc, rng-tools, setup, sysvinit
Sound	alsa-lib, alsa-plugins, alsa-utils, audiofile, lame, libmad, libsamplerate, libvorbis, mutagen, sndfile
System	bluez-utils, checkpolicy, crackerjack, fm, hdparm, ipmi-test, ldcskt, ltp-full, netperf, numactl, ocfs2-tools, oncpu, openhpi, osso-gwconnect, pinentry, pmem, posixtestsuite, robust-tests, saftest, screen, simple_exec_open, simple_exec_proprietary, unionfs, wireless-tools, wpa_supplicant

Appendix A: Package Summary by Category (Cont.)

Category	Packages
System	bluez-utils, checkpolicy, crackerjack, fm, hdparm, ipmi-test, ldcskt, ltp-full, netperf, numactl, ocfs2-tools, oncpu, openhpi, osso-gwconnect, pinentry, pmem, posixtestsuite, robust-tests, saftest, screen, simple_exec_open, simple_exec_proprietary, unionfs, wireless-tools, wpa_supplicant
System analysis	latencytop, ltt-control, lttv, oprofile, perf, PowerTOP, ust
Test	application_args_proprietary, crypto_proprietary, cut-all, cut-ecgl, cut-mutex, cyclictest, death-notify, gen_coredump, hello_proprietary, lmbench, low_latency_mem_proprietary, m4, mailbox_proprietary, named_block_proprietary, network-cgl4, openais-test, perl-net-telnet, queue_proprietary, revoke, traffic_gen_proprietary, uart_proprietary, wifitest, xreg, xts, zebra
Utilities	at, bc, bootlogger, bridge-utils, coreutils, desktop-file-utils, diffstat, diffutils, file, findutils, gettext, gmp, gnome-doc-utils, gtk-doc, make, mhash, mktemp, mpatrol, newt, ntp, popt, readline, slang, sysfsutils, syslinux, time, timezone, ustr, util-linux/libblkid, which
Various	bsdjail, cyrus-sasl, ed, ElectricFence, freeglut, galculator, gdbm, gpm, vim
Virtualization	kvm, libvirt, qemu-kvm
Wind River instrumentation	wbagent-ptrace, wr-coverageagent, wr-opagent, wrproxy
X Server	libdmx, libfontenc, libICE, libosso, libSM, libX11, libXau, libXaw, libxcb, libXcomposite, libXcursor,, libXdamage, libXdmcp, libXevie, libXext, libXfixes, libXfont, libXfontcache, libXft, libXi, libXinerama, libxkbfile, libxkbui, libxklavier, libXmu, libXpm, libXrandr, libXrender, libXres, libxslt, libXt, libXTrap, libXtst, libXv, libXvMC, libXxf86dga, libXxf86misc, libXxf86vm, marquee-plugins, mesa, mesa-demos, sapwood, ttf-arphic-ukai, ttf-arphic-uming, ttf-dejavu, ttf-kochi, ttf-unfonts-core, xcb-proto, xgtk, xkeyboard-config, xorg-x11-apps, xorg-x11-drv-i810, xorg-x11-drv-fbdev, xorg-x11-drv-keyboard, xorg-x11-drv-mouse, xorg-x11-drv-vesa, xorg-x11-drv-void, xorg-x11-filesystem xorg-x11-fonts, xorg-x11-font-utils, xorg-x11-proto-devel, xorg-x11-server, xorg-x11-server-utils, xorg-x11-twm, xorg-x11-util-macros, xorg-x11-utils, xorg-x11-xauth, xorg-x11-xbitmaps, xorg-x11-xinit, xorg-x11-xkb-utils, xorg-x11-xtrans-devel, xrestop, xterm

Appendix B: Supported Target Boards

Supported and unsupported boards can be found in the Wind River Linux Platforms area of the Wind River Online Support website: http://www.windriver.com/products/bsp_web/.

Appendix C: Supported Development Hosts

The following table contains a complete list of supported development hosts with the necessary updates. It lists which hosts support the Application Developer package only and which hosts also support the Platform Developer package.

Host Support for Wind River Linux 4.0 and Wind River Workbench 3.2.3 on Application Developer	Host Support for Wind River Linux 4.0 and Wind River Workbench 3.2.3 on Platform Developer
x86 32-bit Windows XP Pro SP3	No
x86 32-bit Windows 7 RTM	No
x86 32-bit and 64-bit Red Hat Enterprise Linux WS 5 Update 5	x86 32-bit and 64-bit Red Hat Enterprise Linux WS 5 Update 5
x86 32-bit Red Hat Enterprise Linux WS 4 Update 8	x86 32-bit Red Hat Enterprise Linux WS 4 Update 8
x86 32-bit and 64-bit Ubuntu Desktop 10.04	x86 32-bit and 64-bit Ubuntu Desktop 10.04
x86 32-bit and 64-bit Fedora 13	x86 32-bit and 64-bit Fedora 13
x86 32-bit and 64-bit Novell SUSE Linux openSUSE 11.2	x86 32-bit and 64-bit Novell SUSE Linux openSUSE 11.2
x86 32-bit and 64-bit Novell SUSE Linux Enterprise Desktop 11.0	x86 32-bit and 64-bit Novell SUSE Linux Enterprise Desktop 11.0
x86 32-bit Novell SUSE Linux Enterprise Desktop 10.2	x86 32-bit Novell SUSE Linux Enterprise Desktop 10.2

Note that although development may be possible on other Linux distributions and versions, Wind River has not certified the product on them.

For more details on features of Wind River Linux, contact Wind River or visit http://www.windriver.com/linux/.