



WIND RIVER LINUX

Based on the Yocto Project implementation of the OpenEmbedded Core (OE-Core) project, Wind River® Linux leverages the velocity of open source innovation while enabling better control and risk mitigation of the software stack. The Yocto Project uses the build instructions (recipes), configuration files, and other metadata to define the open source base for the Wind River risk mitigation add-ons that help manufacturers during the productization and lifecycle of their devices. These add-ons include security vulnerabilities protection, open source IP compliance artifacts, and product support and maintenance.

TABLE OF CONTENTS

Wind River Linux 9 Changes and Enhancements	2
Host Support	2
Key Components	3
New Use Cases and Enhanced Procedures	4
Wind River Linux Platform Development	5
Platform Project Image Development	6
User Space	7
Toolchain Integration	7
Kernel	8
Extra Defect Fixes with Wind River Simulated Target Development	9
Hardware Support	10
Testing and Validation	10
Legal Compliance	11
Partner Ecosystem	11
Wind River Professional Services	11
Wind River Education Services	12
Wind River Customer Support	12
Lifecycle	12
Appendix A: Kernel-Related Package Version Changes	13

WIND RIVER LINUX 9 CHANGES AND ENHANCEMENTS

The Wind River Linux open source base supports all Yocto Project build commands but also offers simplified configure and build commands. This functionality greatly reduces the learning curve, so users can start implementing the technology much more quickly. It also simplifies the development process by offering configurations that have been tested and options that have already been proven to work.

The Wind River Linux 9 base comes with a set of new and enhanced features:

- **New add-on features:** These include continuous Common Vulnerabilities and Exposures (CVE) protection and open source software (OSS) compliance artifacts
- **New platform project development approach:** No separate installation environment is required.
- **Full support for all Yocto Project commands and work flows:** Commands and work flows match those of the Yocto Project community, in particular for the the Yocto Project 2.2 release.
- **Wind River-specific setup.sh tool for project setup:** The setup.sh tool provided with the Wind River Linux Git repository replaces the configure script and helps set up project configuration while also providing options for listing available project development possibilities.
- **Support for GObject Introspection using qemu:** This uses an object framework for the GNOME desktop.
- **WIC support for generating partitioned images:** This allows the rootfs image that is created among the multiple partitions to be split.
- **Whitelist for the entire meta-openembedded layer:** This includes meta-networking and meta-webserver, in addition to any other packages that users can easily add to a whitelist.
- **OpenCV:** A newly added computer vision library leverages increased computational efficiency.
- **Gperftools:** This provides faster malloc than the glibc.
- **Enabling Yocto Project devtools for application development:** This collection of tools helps development—in particular, user space development—with features such as package import, patch export, local layer, and existing package updates.
- **New features from kernel uprev:** New features include THP for page cache (tmpfs only), a unified control group hierarchy (except CPU control group), eXpress Data Path, ARM64 new features (kprobe, kexec, NUMA), OrangeFS (a new distributed file system), USB 3.1 SuperSpeedPlus, ARM® CoreSight integration with perf, Uprobe support for ARM64, and cryptodev.

Host Support

Wind River Linux can support a variety of modern Linux host operating systems. For convenience, Wind River pre-validates Wind River Linux on a subset of them (defined as validated hosts).

Table 1. Recommended Development Hosts

Operating System	Architecture
Red Hat Enterprise Linux Workstation 7.2	x86 64-bit
OpenSUSE 13.2 and 42	x86 64-bit
Fedora 24	x86 64-bit
Ubuntu Desktop 14.04 LTS and 16.04 LTS	x86 64-bit

KEY COMPONENTS

The product consists of a constant flow of risk mitigation services on top of the community-released source code and a build system that generates an optimized runtime image suitable for embedded devices. The components of the product are referenced by the developer to create a defined runtime image.

The Wind River Linux open source base contains the following components:

- **Application packages:** Hundreds of software packages that operate in protected Linux user mode
- **Kernel source:** The 4.8 Linux kernel with many fixes and feature enhancements
- **BSPs:** Hardware enablement components
- **Tools:** Support for OpenEmbedded and Yocto Project-specific tools such as devtool, Toaster, WIC, eSDK
- **Build system:** Yocto Project build system, used by the developer to compile and assemble these components
- **Toolchain:** GCC 6.2 toolchain

Platform development creates a bootable Wind River Linux system for a target, including kernel, file system, and libraries.

Add-on components are delivered as product updates and artifacts:

- **CVE protection for Wind River Linux:** Monthly or quarterly product updates and bimonthly security bulletins. The technology base is kept up to date and protected against security attacks through constant regular maintenance with security patches and critical Linux updates. Fixes to known security vulnerabilities are provided throughout the lifecycle of the product. The Wind River security team is constantly monitoring security vulnerabilities, including specific security notifications from U.S. government agencies and organizations such as the National Institute of Standards and Technology and the United States Computer Emergency Readiness Team, as well as public and private security mailing lists and the CVE database at cve.mitre.org. The team receives alerts from each of these organizations whenever a new security threat arises. Alerts include both community-confirmed and potential vulnerabilities. Wind River mitigates these threats through a four-step approach: monitoring, assessment and prioritization, notification, and remediation.

- **OSS compliance artifacts:** IP and export compliance artifacts. Wind River reduces the risk and costs of open source adoption with the due diligence provided by a licensing compliance and export disclosure program. Whether it concerns protecting the IP developed on top of the OSS base or clearing export classifications, these issues must be addressed early in the supply chain or there can be costly repercussions for downstream customers—fees and fines, business disruption, even lawsuits. The product comes with a compliance envelope—a zipped archive that contains the following:
 - All required licensing data
 - Source code legal notices
 - Export cryptography information associated with the OSS used to construct the product
- **Support and maintenance for Wind River Linux:** Monthly or quarterly product updates, depending on the lifecycle of the platform.

NEW USE CASES AND ENHANCED PROCEDURES

Wind River Linux 9 allows manufacturers to develop applications and packages on top of a Yocto Project-compatible base while providing a flowing stream of support, maintenance, and risk mitigation assets designed to deliver quality and peace of mind when focusing on innovation.

New use cases and procedures are supported in the Wind River Linux 9 base platform, including the following:

- **Installation options:** Git cloning, offline installation (.zip or .udf), and Wind Share Sync. The setup.sh tool provided with the Wind River Linux Git repository helps set up project configurations while also providing options for listing available project development options. Using Git and Repo, users can also download a mirror that subsequent projects can use. This work flow allows a typical “non-connected” user to download and install the product for later use.
- **BitBake build work flow:** In prior releases, the make command was used to configure a platform project. This automatically parsed the project directories and began the build process depending on the specified options. In contrast, Wind River Linux 9 uses BitBake to build all project-related content. This command must be run with a recipe option, which defines exactly what the system will build. This includes building packages, the kernel, and images.
- **OpenCV for processing real-time applications:** The Open Source Computer Vision library was designed for computational efficiency. The library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform. Usage ranges from interactive art to mines inspection, stitching maps on the web or through advanced robotics.
- **gperftools for faster malloc:** gperftools provides fast malloc, working particularly well with threads and STL. This feature satisfies the need for fast malloc by providing libtcmalloc, which has a faster malloc than glibc.

- **Uprobe for ARM64:** Uprobe has been supported in Wind River Linux 9 on ARM64. This is a user space dynamic tracing feature that can be enabled in the kernel.
- **WIC:** Support for the OpenEmbedded tool used for creating boot images. WIC supports the automatic generation of default image files, generating partitioned images from existing OpenEmbedded build artifacts.
- **Debug filesystem feature:** This feature allows users to bypass the multiple steps involved in assembling the debugfs directory and creates a debug filesystem that can be used directly in debugging tools such as GDB. This companion debug filesystem contains only the package database and the complementary *-dbg packages for the main filesystem component. This is useful in a production environment for producing a companion filesystem capable of remote system debugging, without requiring corresponding debug symbols or source code on the device.
- **Extensible SDKs and devtool:** By enabling these tools, users can patch existing packages, import applications, create wrapper recipe files, and then package the results (replacing Wind River Linux 8 Import-package).
- **Toaster update:** While Wind River Workbench is no longer supported, users may access their integrated development environment (IDE) of choice with Wind River Linux. A version of Yocto Project Toaster is available, updated by Wind River to include the ability to create a platform project and add templates from the Web-based interface. Using Toaster, developers can browse, add, remove, and import layers using data from the layer index; browse present and available images, recipes, and machines; set configuration variables; and select a target or multiple targets to build or customize images on a per-package basis with dependencies automatically.

WIND RIVER LINUX PLATFORM DEVELOPMENT

With Wind River Linux, users can configure and build a platform project image to develop it according to their needs. There are many options for the board, kernel configuration, user space configuration, and so on. Or the profile option may be used to select the kernel and user space configuration automatically, based on the selected BSP and profile combination.

Integrated Development Environment

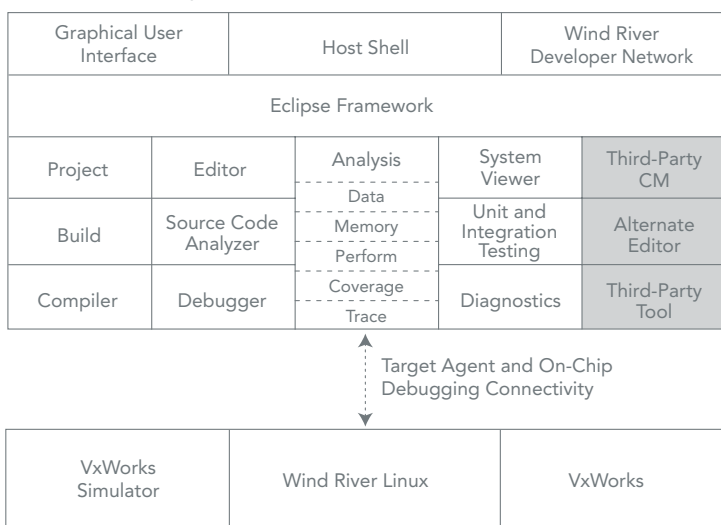


Figure 1. Working environment for building Wind River Linux

Platform Project Image Development

Users start by creating a project using the `setup.sh` tool provided with the Wind River Linux Git repository. The tool also provides options for available project development options.

The build system uses metadata to define all aspects of the platform project image and its applications, packages, and features. Metadata resides in the development and build environments. From a build system perspective, metadata includes input from the following sources:

- **Recipe (.bb) files:** Wind River Linux 9 uses more than 1,000 recipes to organize metadata.
- **Recipe appends (.bbappends):** These allow changing recipe behavior from one or more layers in an easily maintained way.
- **Class (.bbclass) files:** These common build instructions are used by the recipes and build system to control the build process.

The build system uses this metadata as one source of input for platform project image creation.

Configuration and Build

The Yocto Project BitBake build system with Wind River Linux allows users to create multiple builds, customized builds, and a strict version control system, while keeping the development environment pristine and intact.

There are four distributions provided by Wind River:

- **wrlinux:** A full file system, with glibc but without carrier grade–relevant packages or extensions. `wrlinux` is not generally recommended for use with `initramfs` due to its memory requirements.
- **wrlinux-std-sato:** A full file system with glibc, optimized for the Sato graphical (`sato-gui`) interface. Users may extend the graphics capability to use the Xfce desktop with the `feature/Xfce` template. `wrlinux-std-sato` is not generally recommended for use with `initramfs` due to its memory requirements.
- **wrlinux-small:** A much smaller, BusyBox-based file system with glibc, intended for small file systems. This includes a standard kernel, a reduced-size BusyBox with a typical OE-core configuration, `sysvinit`, `systemd`, and `rsyslog`.
- **wrlinux-tiny:** A minimal file system you can use as a starting point for developing very small distributions for resource-constrained devices. This includes a tiny kernel, a tiny BusyBox, and a custom `init`.

Layers

Layers provide a mechanism for separating functional components of the development environment. They are multiple independent collections of recipes, templates, code, configuration files, and packages, typically contained in a layer directory. Multiple layers may be included in a single project, and each layer can provide any combination of features, ranging from kernel patches to new user space packages.

A layer allows the addition of new files, such as the recipes that define a specific package or packages, and machine configuration files that define a board for a new BSP, without modifying the original development environment. Users can create their own layers and organize the content to better suit their development needs, and include or exclude the layers from the project configure and build.

In Wind River Linux, layers reside in the installation (development) environment and the build environment. When users configure and build a platform project image, the layers in the installation provide configuration information depending on their platform project configuration.

Recipes

If users want to include applications or packages in their platform project build, they must have a recipe file associated with them. The build options make it possible to copy and modify an existing recipe file, or create one from scratch.

Templates

Wind River Linux provides feature and kernel templates to simplify development. Once users configure a platform project, templates are added to their platform project directories. Templates are small configuration fragments that customers can use to simplify the potentially complex configurations of the Yocto Project components.

User Space

Customers can use their preferred IDE or the Yocto Project devtool or Extensible SDKs (eSDKs) to develop applications and cross-compile them for the intended target.

The eSDK is a portable and standalone development environment, like an SDK with an added BitBake executive via devtool. eSDKs run on Linux hosts but do not run on Windows (unless developers use the OpenEmbedded CROPS container technology).

Through devtool there is a rich set of commands to:

- Add, modify, or upgrade a recipe
- Show work space status and search available recipes
- Build, edit, or remove a recipe from the workplace

Toolchain Integration

Users compile packages from source code to binaries using the Yocto Project source toolchain.

Among the new updated components:

- binutils 2.27
- GCC 6.2
- gdb 7.11.1
- glibc 2.24

Kernel

Additional kernel build optimizations have been added for the following:

- **Building a project with a dummy kernel:** Using a dummy recipe, users can build just the user space, with no real kernel.
- **Building a kernel with a custom recipe:** Using a feature/custom recipe in the kernel-dev layer (must use the git tree), users can build a third-party kernel.
- **Building a kernel from an external source:** Using the externalsrc mechanism in the Yocto Project (supporting tarball source), users can build a third-party kernel.
- **Extracting kernel build output**

New kernel features and fixes are integrated from upstream, e.g., THP for page cache (tmpfs only); unified control group hierarchy (except CPU control group); eXpress Data Path; and ARM64 new features kprobe, kexec, NUMA, and USB 3.1 SuperSpeedPlus. They are also backported from upstream, e.g., Intel® Memory Protection key and Intel Graphics driver refresh. And they come from Wind River partner alignment and internal product evolution, e.g., out-of-tree features such as support for LTTng 2.8, Aufs 4 and Yaffs2, Uprobe for ARM64, Cryptodev, and BFQ I/O scheduler (Budget Fair Queueing). For a list of kernel-related packages, consult Appendix A.

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.

Customizing the Linux kernel to better fit the particular details of a hardware implementation is almost always a required step in an embedded software development cycle. Users can add or remove options to the Linux kernel by applying patches directly to the source code.

Kernel customization can take the form of simply enabling or disabling kernel configuration options, typically to enable specific drivers and to shrink the final kernel image and runtime load by removing unneeded functionality. Customization can also come in the form of patches, either in-house or third-party, applied to the source code to modify specific areas of kernel behavior. Kernel tree changes consist of patches, or configuration fragments, that are applied to the branch or to the board/target-specific system kernel being built.

Unlike other packages in the build system, the kernel is not single-purposed or targeted at a particular piece of hardware. It must perform the same tasks and offer the same APIs across many architectures and different pieces of hardware.

The key to managing a feature-based patching of the Linux kernel is to remove both the distributed control of the patches (subdirectory-based patches.list files) and the hand editing of the patch files.

Replacing these two characteristics with script-based patch list generation and a method to control and describe the desired patches with a top-down approach eases the management of kernel patching complexity. Additionally, direct mapping between BSPs and profiles is easy and increases maintainability. The scc script has been implemented to control the process of patch list generation and feature-based patching.

Wind River provides predefined kernel styles that are specific to users' applications. The kernel styles shipped with the product are described here.

Tiny

Inherited from the Yocto Project, Tiny is the most basic kernel configuration for bringing up the target, and it has a small footprint.

This option is only valid when using the corresponding glibc-tiny file system. As opposed to the standard configuration, it is intended to be a very small device starting point and not a general purpose configuration.

preempt_rt

Wind River Linux provides a conditional real-time kernel type, preempt-rt kernel from the Yocto Project, for certain board and file system combinations. To enable the preemptible real-time feature, users must simply configure their project with the preempt-rt kernel option.

Highlights of this option include:

- Support on four architecture families: Intel, ARM, PPC, MIPS
- Upstream 4.8.11-rt7

EXTRA DEFECT FIXES WITH WIND RIVER SIMULATED TARGET DEVELOPMENT

Embedded systems are typically configured, programmed, and built on a host system and then deployed to a target system. If you have a physical target system or board, you will test your platform and application on that target. Your target might have a network interface and on-board tools through which you can deploy software.

You may not have the actual target available at all times, so it may be more convenient to use a target simulator on your host. Wind River Linux provides simulated target development with QEMU and Wind River Simics®, in addition to supporting a range of hardware target BSPs. Refer to the Wind River Linux Release Notes for information on supported BSPs.

QEMU provides a simulated deployment environment for testing platform projects and applications. It supports many development boards, and it does not require actual target board hardware or networking preliminaries. QEMU deployment offers a suitable environment for application development and architectural level validation. User space and kernel binaries are compatible with the real hardware. QEMU provides a means to rapidly test and debug platform projects and applications using command line options.

Simics creates a high-performance virtual environment in which any electronic system—from a single board to complex, heterogeneous, multi-board, multiprocessor, multi-core systems—can be defined, developed, tested, and deployed. 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 have experienced dramatic cost savings throughout the development lifecycle, reached market 18 months faster, cut a year's time from ecosystem enablement, and produced higher-quality products. Simics also speeds and simplifies development on cutting-edge multi-core hardware. For more information, visit www.windriver.com/products/simics/.

HARDWARE SUPPORT

Wind River Linux 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 the online Wind River Support Network to customers under an active platform subscription. Also, Wind River Professional 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 runtime for each supported reference board. The supported features are board-specific and depend on the availability and maturity of the code in the open source community.

Wind River Linux ships with a wide range of 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.

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 140,000 test runs. More than 30 million lines of code were tested using this Test Automation Framework, which 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.

Testing of Wind River Linux includes the following:

- Sanity testing
- Feature testing
- Regression testing

- Complete build testing
- Full runtime testing
- Documentation testing
- Fix verification
- Out-of-the-box experience (OOBE) testing
- Benchmark/performance testing
- Compliance testing (LSB, IPv6, IPsec, etc.)

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.

Using the Linux Foundation's standard format for recording and exchanging licensing information of a software package, Wind River uses the data generated by the software compliance review process and delivers SPDX information. Wind River supports the SPDX project in various ways:

- Actively contributes to the SPDX technical, legal, and business working groups
- Utilizes SPDX data in software IP compliance review
- Provides a free cloud service where anyone can create SPDX files

PARTNER ECOSYSTEM

The world-class Wind River partner ecosystem ensures tight integration between our core technologies and those of the premier hardware and software companies 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 www.windriver.com/partners.

WIND RIVER PROFESSIONAL SERVICES

A CMMI Level 3–certified organization, Wind River Professional Services offers a unique mix of embedded and vertical market expertise. We offer consultative thought leadership, deep technical capabilities, and innovative industry solutions to help you overcome your most strategic and pressing development challenges.

As part of our comprehensive solutions, Wind River offers a Linux Services Practice, with focused offerings that help customers meet strict market deadlines while keeping development costs down. Our industry-specific offerings span the entire project lifecycle, including architecture, design, development, porting, integration, and maintenance services; and we leverage our state-of-the-art platform simulation and test tools to accelerate deliverables and provide valuable reporting and documentation. Our global organization provides flexible engagement options for consulting, training, and support that will meet your project resourcing requirements and budget. For more information, visit www.windriver.com/services.

WIND RIVER EDUCATION SERVICES

With more than 30 years of embedded software experience, Wind River provides education services in every region of the world. We offer flexible training options to meet your business and learning needs, including public, private, and custom courses. For your specific project challenges, Wind River Mentoring provides coaching by experienced engineers to help you integrate Wind River solutions into your environment. And when you're too busy to attend a whole class, our on-demand learning options provide around-the-clock access to advanced and specialized topics. All of our education services are led by expert engineers who are closely connected to the Wind River technical community for access to specific expertise. For Wind River Linux we offer deeply technical hands-on courses, including RTOS to Linux Migration Essentials, Wind River Linux Application Development, Wind River Linux BSP Development, and many more. For more information, visit www.windriver.com/education.

WIND RIVER CUSTOMER SUPPORT

Wind River Linux is backed by our award-winning global support organization. With more than 150 experts worldwide, you can get the help you need in the language and time zone that work best for you. Our online Wind River Support Network provides multifaceted self-help options, including an active Q&A forum. Optional premium services, including designated support engineers and hosting of customer-specific environments, offer the fastest possible time-to-resolution. We are proud to have achieved Service Capability and Performance certification, recognized as the gold standard for delivering world-class customer support. For more information, visit www.windriver.com/support.

Lifecycle

The Wind River Linux release model follows a one-year update policy; that is, one major release version every year. In effect, this means that every 12 months Wind River introduces a kernel update for the Wind River Linux base platform and other architectural changes deriving from this. Software patches and minor updates to the current product releases are delivered via rolling cumulative patch layers (RCPLs) on a monthly basis during the first two years and on a quarterly basis afterwards. Updates do not trigger any major kernel (first and second digit), toolchain, or user space package revisions (unless a critical issue requires an up-rev; please refer to the RCPL policy for additional information). RCPLs are accessible using the Wind River electronic software delivery mechanism, and information

about each RCPL, including technical content, is available on the Support Network. Certain product enhancements and new BSPs may also be included in these monthly patches, and RCPLs are mandatory installations. All RCPLs go through the Wind River standard product validation tests.

Every major version of Wind River Linux is maintained and supported over several years. When customers want to continue running legacy versions of Wind River Linux and wish to continue receiving maintenance and technical support services, Wind River Linux Long Term Support offers an extension of the product lifecycle. It includes e-Support, which provides continuous access to the Support Network; Enterprise Support, which adds live support and maintenance; and customer-specific maintenance. For more information, contact your local account team or wr-support-info@windriver.com.

APPENDIX A: PACKAGE VERSION CHANGES

Wind Rver Linux 8	Wind River Linux 9
Libvirt 1.2.19	Libvirt 1.3.5
Libhugetlbfs 2.19	Libhugetlbfs 2.19+
Lxc 1.1.4	Lxc 2.0.4
CRIU 1.6	CRIU 2.5
Open vSwitch 2.4	Open vSwitch 2.4

