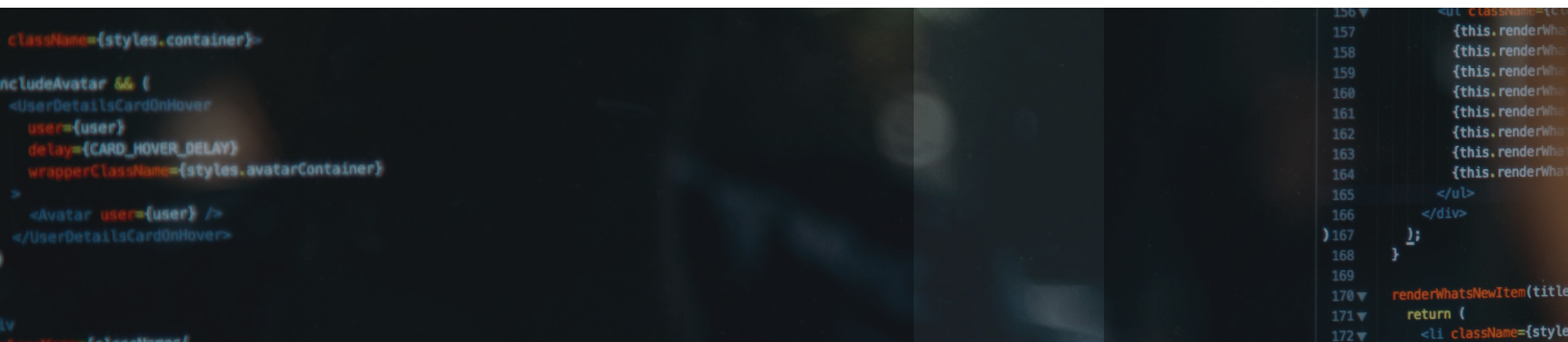# Purpose-Built Wind River Linux Distribution

Create and Customize Your Linux

WNDRVR

As processing power increases each year, so does the number of embedded devices. According to a 2022 forecast from The Brainy Insights, the global embedded systems market is expected to grow at a compound annual growth rate of 5.7% by 2028.

Although some applications require a real-time operating system, using an RTOS comes with several challenges, including cost of development, time-to-market, memory management, security, hardware compatibility, and compliance.

Wind River® Linux, an open source, embedded Linux development platform, provides a supported solution that addresses many of these difficulties while greatly accelerating time-to-market. As a leading contributor to the Yocto Project and other open source communities, Wind River has helped customers deliver embedded solutions across a variety of industries and applications (aerospace and defense, automotive, industrial, medical, telecom) for more than 40 years. And it does this by providing designers with a toolset to create their own purpose-built Linux systems.

The Wind River Linux Binary Distribution

Previously, Wind River Linux was only available in a source-based distribution builder (known as Wind River Linux) that allows embedded developers to build an entire Linux operating system from source for complete customizability. While powerful, that approach has a high learning curve and may not be best choice for development of all embedded solutions. In contrast, the Wind River Linux binary distribution (Wind River Linux Distro) can be used to create and customize your own Linux image in just a few minutes.

Wind River Linux Distro BSP Support
Intel Ice Lake
Intel NUC Kaby Lake
Intel Snow Ridge
Interk Tiger Lake UP3
Intel Stratix 10
Marvell Octeon CN96xx (TX2)
NXP i.MX8 QuadMax MEK
NPX S32G
Raspberry Pi 4
TI DRA829/TDA4xM
Xilinx UltraScale+MPSoC
Xilinx Zynq-7000
.. and more

Figure 1: Wind River Linux Distro hardware support list

---

1 *"Embedded System Market to Hit USD 138.45 Billion by 2028" (press release), The Brainy Insights, February 3, 2022*

Key Benefits of Wind River Linux Distro

Wind River Linux Distro is based on the Wind River Linux LTS 21 source release and supports a variety of Arm® and X86 platforms. With project-based pricing, there are no end-device royalties. And, if needed, a user can opt for extreme customization in the Wind River Linux source-based product.

The Wind River binary distribution provides micro-start images that install all necessary packages and file systems on first boot. This includes two configurations: a small minimal image and a full-function image with a graphical user interface. The binary packages come in RPM format and include a tool that assembles packages into several types of images. In addition, an SDK helps support platform and application development, while OSTree (a system for versioning updates) and OSTree wrappers are supported for simplification.

The most important benefit of the binary distribution as compared to the source code distribution builder is how quickly a user can create an embedded Linux image. Steps for a basic use case are outlined in the sidebar at right.

Other benefits of Wind River Linux Distro include:

- Micro-start of self-deploying images using OSTree
- Package updates via OSTree
- Package feed enabling adding packages to your image
- Software development kit with the Linux Assembly Tool (LAT)
- Binary container base image on Docker Hub
- Signing to inhibit tampering
- Hardware support for several platforms (see Figure 1)

AN EXAMPLE USE CASE

Let's look at a use case for installing and building an image:

1. **Install the device:** Get the image.
2. **Customize the device:** Get the image and install packages from an RPM feed.
3. **Build an image:** Get the SDK and build the image.
4. **Device update:** Use OSTree to update and use package feed to update the package release.
5. **Build containers:** Pull from the container registry and update from RPMs.
6. **Build a package feed:** Use SDK with the Linux Assembly Tool (LAT) to build images, containers, package feed, and OSTree feed.

A quick-start guide can be found here: docs.windriver. com/bundle/Wind_River_Linux_Distro_Developers_Guide_LTS_21/page/vcb1630021339626.html

OSTree Benefits and Purpose

OSTree performs atomic upgrades of complete file system trees (such as Git). It can do so with incremental updates, rollback, deterministic updates, complete image replacement, and with or without an A/B partition scheme. This makes it well suited for CI/CD. Figure 2 shows a disk view of OSTree updates.
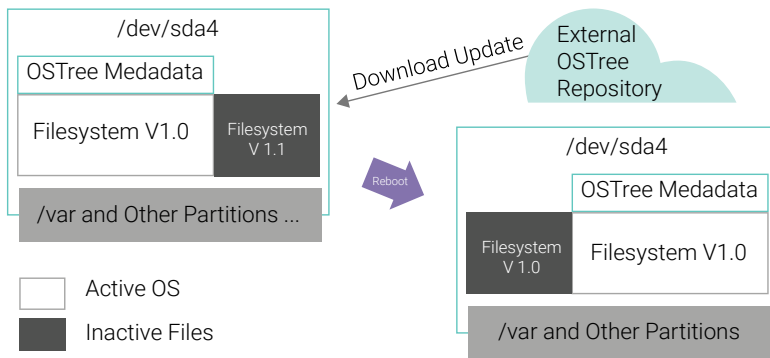

Figure 2: Disk view of OSTree update

Micro-start images (ustart.img) are used in the distribution process. The default image includes a compressed version of an installer as well as init-ramfs, boot loader, kernel, and an OSTree repository; a custom image would just have the boot loader and a network connection to bring over everything else. This process provides a flexible install with fresh file systems and the ability to factory-reset corrupt devices to their original install state. And if a user were to rebuild the binary distribution, they could install directly from a network (and use a network for PXE/Net UEFI/U-Boot) as well as alternate disk layouts (use A/B partitions, add additional partitions, and encrypt rootfs and data using a TPM stored key).

When you customize your image, /etc and /var are persistent by default unless a factory reset or factory upgrade is performed, but a user can use ostree admin unlock to temporarily unlock. And, by default, ustart.img does not enable a console, but this can easily be enabled.

While the initial image lies within Docker Hub, the binary distribution container base (which is roughly 36 MB compressed) contains a package manager and is built from the exact same package feed as the released OS. There's also an option for a smaller reference (only 5 MB), but you must build from source Wind River Linux distribution and select the OPKG Package Manager to build the container base.

The Linux Assembly Tool and a Lego Brick Analogy

The LAT can be used to perform a number of tasks to help manage images, build and publish RPM packages (from any files, including third-party content), generate images from feeds for specific hardware, create customized initramfs boot loaders and custom deployment actions, and generate updated SDKs.

Think of LAT as a set of plastic toy construction blocks, like Legos. But instead of building toys, you're assembling software into an embedded solution. The build system creates "bricks," or apps/libs/packages, which provide each of the Lego pieces (artifacts). Building an RPM may consist of starting with external content or materials (such as glue, plastic, etc.) and then using the LAT genrpm (3-D printer, in this case) to produce custom packages (or Lego bricks, in this analogy). Building a boot loader (appsdk geninitramfs) would essentially build the floor or foundation for these bricks. And while building the containers (appsdk gencontainer) creates rooms or houses on top of this foundation, building the SDK (appsdk gensdk) packages everything up into a kit — sort of like a prepackaged Lego kit.
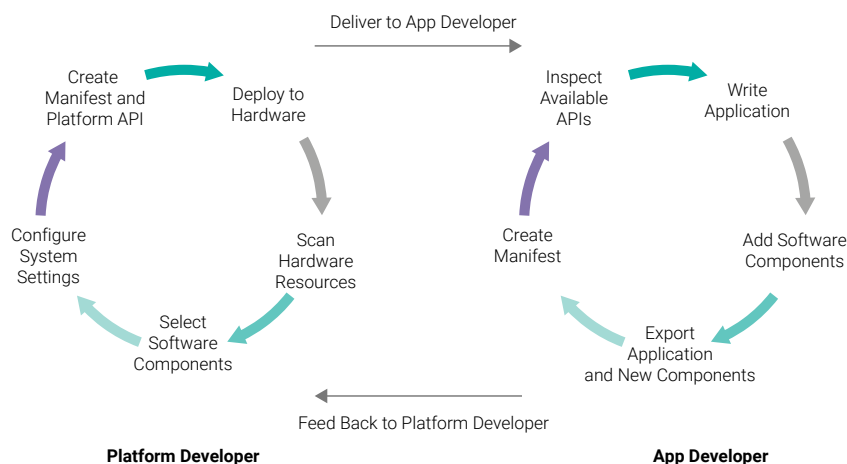
Below is a list of some of the LAT core commands:

- **appsdk genrpm:** Uses yaml file to build RPM from any files, including third-party or original modified content

- **appsdk gencontainer:** Uses yaml file to create a container from RPM

- **appsdk geninitramfs:** Uses yaml file to create a create customized boot loader and/or deployments

- **appsdk genimage:** Uses yaml file to create an image that can contain any content from above, as well as deployment and scripting; you can also use an alternative initramfs generated above

- **appsdk gensdk:** Requires a full build with the initial SDK; yaml allows rolling all inputs above into a new set of artifacts with a new SDK

Figure 3: The LAT and developer lifecycle

Source-Based vs. Binary Distribution

Let's compare the Wind River Linux source-based distribution builder with the Wind River Linux Distro binary distribution and consider why a customer might choose one over the other.

With the source-based approach, a customer can have customized hardware enablement as well as full control over the kernel configuration, with all changes tracked for full reproducibility. This might require a team of capable Linux builders as well as a tightly integrated platform and application team. SDKs can be created, and the result is a fully customized and tailored Linux.

With the binary distribution, a quicker time-to-market can be achieved at a more economical price point. And because no special requirements are needed, minimal changes can be made to Linux, with limited experience necessary. That allows the solution development team to focus on the application and results in a "good enough" Linux-based solution. The binary distribution is based on Wind River Linux LTS 21 – LTS kernel 5.10 and is designed to support creation of customized Linux images in just a few minutes. A customer has the flexibility to customize their own purpose-built Linux while receiving support for a growing variety of Arm and X86 platforms. OSTree automatic updates and the absence of end-device royalties add simplicity while reducing cost.

Embedded solution developers need a quick and easy purpose-built embedded Linux for their edge solutions. Wind River Linux Distro provides a solution, allowing developers to create a purpose-built Linux without the time and expense of starting from scratch.

Download Wind River Linux Distro today and get started building your product:

www.windriver.com/products/linux/download

WNDRVR