



Yocto Project – Big in Embedded Linux



Rudolf Streif

Director of Embedded Solutions, The Linux Foundation

April 2012



A Linux Foundation **Linux Training** Publication

<http://training.linuxfoundation.org>

Linux Training on the Yocto Project

As a part of our Linux Training Program, The Linux Foundation offers courses designed to help companies and their developers jump-start embedded development projects using the Yocto Project and to give software engineers a solid understanding of the build process from compilation over packaging to creating a bootable image. Participants will step through all phases of embedded Linux development using the Yocto Project with laboratory exercises on an embedded target device to provide hands-on practice you can take directly to your projects. There are currently two Yocto Project courses available:

LF405 Building Embedded Linux with the Yocto Project

LF404 Building Embedded Linux with the Yocto Project: Crash Course

Please visit our [Linux Training website](#), or click on the links above for more information.

Summary

The Yocto Project is a collaboration project providing a coherent toolset to help developers create custom Linux-based systems for embedded products. Through an automated workflow controlled by templates and metadata it is able to create a system from source to image. Preconfigured profiles and targets for various applications and processor architectures jump-start the process for embedded engineers. Hosted by the Linux Foundation as a workgroup it is governed by open-source best practices.

Yocto, the prefix in the metric system denoting a factor of 10^{-24} , gives the name to an open-source collaboration project that greatly streamlines the creation of a custom Linux distribution for embedded devices. However, saying that the name is an understatement is an understatement itself.

The Yocto Project is a complete embedded Linux system development environment with tools, recipes, and documentation. The Yocto toolbox is self-contained and includes compilers, debuggers, packagers, an emulation environment, an application toolkit generator and many more making it the ideal ready-to-go solution for embedded Linux developers. And yet, it is easy to use. The available resources, templates, and documentation cater to both novice and experienced embedded Linux developers.

Community-tested images covering various build profiles across multiple architectures including ARM, MIPS, PPC, 32- and 64-bit x86 provide a solid starting point for new engineering efforts. Specific hardware platforms are supported through Board Support Package (BSP) layers for which the Yocto Project has developed a standard format.

The icing on the cake is the Application Toolkit Generator that automatically creates a complete environment for engineers to develop applications on top of a platform created by the Yocto Project. This environment seamlessly integrates with a plugin for the popular Eclipse IDE and can be shipped by device vendors together with their hardware platform to jump-start application developers.

Why Linux? Why Yocto?

A recent survey of embedded developers conducted by [Embedded Market Forecasters](#) reported that two primary factors contributing to embedded developers' choice of operating systems are cost (44.6%) and the availability of source code (33.1%). From this point of view, Linux seems to fit the bill quite nicely. However, Linux puts up some hurdles that not too long ago were hard for embedded systems and their developers to jump.

Linux requires a file system. For Linux to run it must have a file system with read and write access. Until the ubiquitous availability of inexpensive and long-term reliable flash memory devices that typically meant a hard drive which is not practical for most embedded use cases.

Linux requires a memory management unit (MMU). Linux is a multi-tasking operating system; effective task switching mandates that individual processes have their private space in physical memory that easily can be mapped into the CPU's address space. Typical microcontrollers that are widely used for embedded applications do not provide an MMU. That did not pose too much of a problem in the past since embedded applications were mostly single-tasking and context switching was limited to interrupt service routines (ISRs). However, today's system-on-chips (SOCs) provide multi-core CPUs with MMUs to meet customers' demands for connected devices. No consumer would accept anymore that he or she cannot access the Internet or have a look at the calendar while conducting a phone call on a smartphone.

Linux is non-monolithic and multi-sourced. While this is essentially a good thing because it lets developers easily customize the OS to fit their particular embedded devices it poses the question where to get a Linux distribution from that fits the needs of embedded devices. For desktops and servers this problem has been solved by the many readily available distributions but for embedded devices there is no such thing as a plug-and-play Linux distribution. And that is simply due to the fact that hardware for embedded devices is naturally much more diverse than standard desktop and server computers. That is where the Yocto Project provides a solution.

What is Yocto?

Contrary to popular belief, Yocto is not an embedded Linux distribution but a kit of tools and recipes to create a custom Linux distribution from source. It relieves the embedded developer from identifying the proper packages that make up a Linux distribution and download their sources as well as from setting up a system with the correct set of tools to build these packages and finally assemble them into an image that can be booted on the target device.

The Yocto Project builds upon and extends a series of upstream open-source projects which form its toolkit. With each Yocto release updated versions of these projects with new features and bug fixes are included. Every new release of Yocto also provides updated recipes and templates to track changes, features and bug fixes of the source packages that make up a Linux distribution such as the Linux kernel.

At the center of Yocto is [Poky](#), a platform-independent, cross-compiling layer that utilizes the [OpenEmbedded](#) core. Poky provides the mechanism to download, patch, build and combine thousands of distributed open-source project to form a complete and coherent but fully customizable Linux software stack.

Figure 1 depicts the Yocto workflow based on OpenEmbedded. User Configuration, Metadata, Hardware Configuration, and Policy Management determine what Upstream Projects, Local Projects (stored on the developer's system) and what sources from optional Source Change Management systems to include

as well as control the individual process steps of the workflow. A Yocto Project release already contains configuration information and metadata that will produce a working Linux system image only requiring very minimal adaption to the developer's local build environment.

While Figure 1 shows a rather linear process consisting of the steps Source Fetching, Patch Application, Configure/Compile, Output Analysis for Packaging, Package Creation and QA Tests these steps are in fact repeated for each source package and therefore

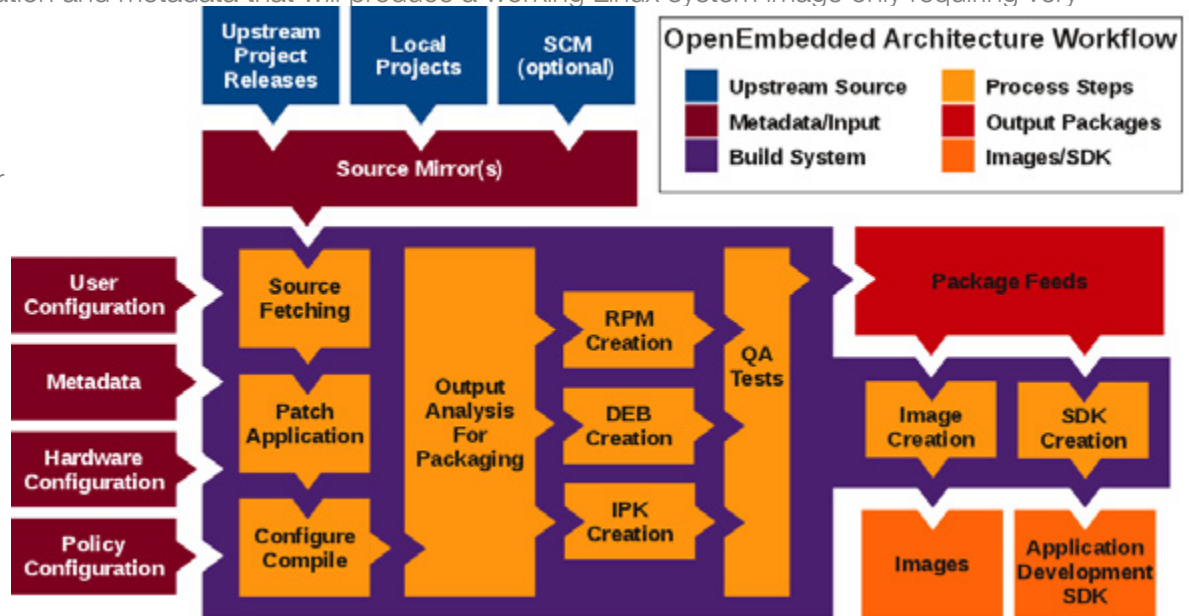


Figure 1.

potentially many thousand times before all Package Feeds have been created and can be combined into an image. Therefore Yocto supports multi-processor and multi-core build systems by automatically calculating dependencies and executing process steps in parallel hence greatly accelerating the entire build process. Of course, Yocto also manages changes and only rebuilds Package Feeds whose input, source, metadata, dependencies, etc. have changed.

To further simplify the process for novice as well as experienced users Yocto provides a series of different profiles that automatically create images for common applications:

- **poky-image-minimal** – A small image just capable of allowing a device to boot.
- **poky-image-base** – A console-only image that fully supports the target device hardware.
- **poky-image-core** – An X11 image with simple applications such as terminal, editor, and file manager.
- **poky-image-sato** – An X11 image with example GUI implementation using Matchbox Window Manager, Sato theme and Pimlico applications. The image also contains terminal, editor, and file manager.
- **poky-image-sato-dev** – An X11 image similar to poky-image-sato but also includes a native toolchain and libraries needed to build applications on the device itself. The image also includes testing and profiling tools as well as debug symbols.
- **poky-image-lsb** – An image suitable for implementations that need to conform to Linux Standard Base (LSB).
- **meta-toolchain** – This profile creates an archive that contains a standalone toolchain that can be used externally to Yocto. The archive also contains QEMU and the scripts necessary to run Yocto QEMU images.
- **meta-toolchain-sdk** – This profile create an archive that contains everything of the meta-toolchain profile but also includes headers and libraries to form a complete standalone SDK.

The above list represents the most commonly used profiles. Many more specialized profiles are available and the list continues to grow.

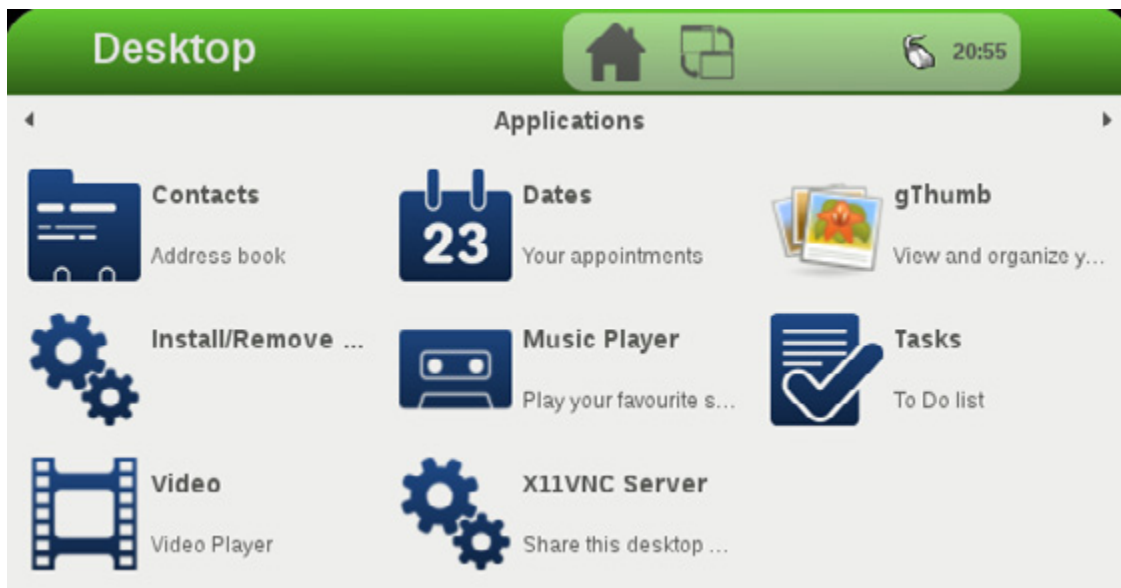


Figure 2. Yocto image created with core-image-sato.

Figure 2 shows the example GUI using Matchbox Window Manager with Sato theme desktop running from an image created with Yocto using the core-image-sato profile.

Who supports the Yocto Project?

The Yocto Project is a Linux Foundation Workgroup. The project was inceptioned by Richard Purdie who has been appointed as a fellow of the Linux Foundation. In this capacity he is working full-time on the Yocto Project, OpenEmbedded, and Poky.

The Yocto Project software is designed and developed through a collaborative effort by an open community of professionals and volunteers and is supported by a series of participating organizations. It is open for collaboration to anyone who is positively contributing. Like many other open-source projects, the Yocto Project employs a governance model consisting of various groups:

- **Contributors** – Professionals and volunteers contributing to the collaborative effort: code developers, documenters, users, etc.
- **Maintainers** – The delegated authorities who review patches that are submitted by contributors and technical leaders, provide feedback to the developers, and accept the ones that are appropriate.
- **Technical Leaders** – Along with a Maintainers the Technical Leaders ensure excellence and functionality.
- **Interest Groups** – These groups define and document the various requirements for the Yocto Project, which are formed around the strongest needs from the users.
- **Architect** – The Project's Architect provides overall technical leadership and resolves conflicts.
- **Advisory Board** – The Project's Advisory Board influences the Interest Groups to ensure that requirements are set for the Project. It's range of responsibilities also includes creating and resolving working groups to support the fulfillment of the project objectives, appointing Maintainers and Technical Leaders, and defining policies and procedures.

Corporate supporters of the Yocto Project include semiconductor companies, operating system vendors, OEMs and others. Through collaboration in the project they effectively are aligning their efforts to everybody's

benefits. Semiconductor companies are lowering their R&D cost for providing software support for their SoCs; OSVs shorten their product development life cycles through reduced customization effort; and OEMs can focus on differentiating their products rather than developing basic and common functionality. Companies contributing to the project today now include Cavium Networks, Dell, Freescale Semiconductor, Intel, LSI, Mentor Graphics, Mindspeed, MontaVista Software, NetLogic Microsystems, RidgeRun, Texas Instruments, Tiler, Timesys, and Wind River, among others.

About The Linux Foundation

The Linux Foundation is a nonprofit consortium dedicated to fostering the growth of Linux. Founded in 2007, the organization sponsors the work of Linux creator Linus Torvalds and promotes, protects and advances the Linux operating system by marshaling the resources of its members and the open source development community. The Linux Foundation provides a neutral forum for collaboration and education by hosting Linux conferences, including LinuxCon, and generating original Linux research and content that advances the understanding of the Linux platform. Its web properties, including Linux.com, reach approximately two million people per month and include important Linux video resources. The organization also provides extensive Linux training opportunities that feature the Linux kernel community's leading experts as instructors.

About the Author

Rudi Streif manages The Linux Foundation's initiatives for embedded solutions working with the community to provide environments and platforms for embedded Linux systems. Mr. Streif has an extensive background in embedded software development and bringing products based on Linux to market.

© Extension Media, (2012). This is the author's version of the work. It is posted here by permission of Extension Media for your personal use. Not for redistribution. The definitive version was published in EECatalog, (August 23, 2011).

The Linux Foundation promotes, protects, and advances Linux by providing unified resources and services needed for open source to successfully compete with closed platforms.

To learn more about our Yocto Project courses or any of our other Linux Training courses, please visit us at <http://training.linuxfoundation.org/>.

