

Introducing OpenEmbedded

Paul Barker
Konsulko Group

Feb 2021

High-level Overview

- Build system for Embedded Linux and more
- Cross-compilation allows efficient development for targets with limited CPU, RAM and storage
- Produces package feeds, final images and any other artifacts you need

Why OE?

- Flexibility
- Wide set of recipes & machines already supported
- Strong community
- Well documented
- Extensively tested
- Excellent progress on reproducible builds
- Integrates with a variety of additional tools

Core concepts

- Recipe – Specifies how to build software packages or images
- Class – Common build configuration used across multiple recipes
- Machine – Target hardware or virtual machine for which software is compiled
- Distro – Selects policy and configuration

Currently Supported Devices

- QEMU & other virtual machines
 - X86, ARM, RISC-V, PowerPC, etc
 - 32-bit and 64-bit
- Generic PC & Server hardware
- BeagleBone & compatible devices
- Most Raspberry Pi models
- Many other popular consumer & hobbyist boards
- Commercial, industrial & specialist systems

Existing Recipes

- C, Python, Java, Rust, Go, ...
- Linux kernel, Zephyr, various hypervisors
- U-boot & other bootloaders
- sysvinit or systemd
- Xfce, LXQT, KDE, Sato, other desktop environments
- Firefox, Chromium, qtwebbrowser, etc
- A multitude of command line tools & services

Project History

- Launched in 2003 to meet the needs of the OpenZaurus project
- Rapidly grew to support a variety of targets & software recipes
- Split into Bitbake (build tool) and OpenEmbedded (metadata) repositories in 2004
- Layer model adopted in 2011, openembedded-core repository created

Additional Tools & Features

- Devtool – Automatic recipe creation
- Wic – Build ready-to-use partitioned images
- Archiver – Support copyleft license compliance
- Package feeds & on-target package management
- SDK

OE & Yocto Project

- Yocto Project launched in 2010
- Builds a ecosystem of technologies, best practices, documentation, training and companies
- Uses OpenEmbedded build system
- Contributes extensive resources to OE
 - Maintenance, documentation, infrastructure & testing

<https://www.yoctoproject.org/>

Yocto Project Releases

- Released every 6 months
- 7 month standard support period
- LTS release supported for 2 years

Release	Date	Support period
3.0 “zeus”	October 2019	9 months, EOL July 2020
3.1 “dunfell”	April 2020	LTS: 2 years, EOL April 2022
3.2 “gatesgarth”	October 2020	7 months, EOL May 2021
3.3 “hardknott”	April 2021	7 months, EOL November 2021

What Next for OE?

- Rust support moving into openembedded-core
- Improving reproducibility
- Hash equivalency service
- Always adding new recipes & target machines

- Current Outreachy interns with Yocto Project
 - Enhance Yocto Project License Tracing
 - Add support for elfutils debug info server

Getting Started with OE

- Quick build guide:
 - <http://docs.yoctoproject.org/brief-yoctoprojectqs/index.html>
- Live Coding tutorials from Josef Holzmayer:
 - https://www.youtube.com/playlist?list=PLD4M5FoHz-TxMfBFrDKfIS_GLY25Qsfyj

Additional Resources (1)

- Source code:
 - <https://git.openembedded.org/>
 - <https://git.yoctoproject.org/>
- Documentation:
 - <https://docs.yoctoproject.org/>
- Wiki:
 - https://www.openembedded.org/wiki/Main_Page
 - https://wiki.yoctoproject.org/wiki/Main_Page

Additional Resources (2)

- IRC: #oe and #yocto on FreeNode
- Mailing lists:
 - <https://lists.openembedded.org/>
 - <https://lists.yoctoproject.org/>

Thanks!