INTRODUCTION TO THE yocto PROJECT
Yocto Project Dev Day

The Yocto Project Developer Day is a one day, hands-on training event that puts you in direct contact with Yocto Project technical experts and developers, held in conjunction with the Embedded Linux Conference.
About Mii

Stephano Cetola
Open Source Firmware
Person
What is The Yocto Project?

The Yocto Project is an open source project that helps developers create custom Linux-based systems that are designed for embedded products regardless of the product's hardware architecture.

--Yocto Project Overview Documentation
Check out the manual(s)

- Yocto Mega-Manual
- BitBake Manual
- Yocto Project Website

Yocto Project Building Blocks

Collaboration in the Open Source Embedded Community

Open Embedded
A line drawn in the sand of a rolling release.

BitBake Build Tool
Make-like build tool with modern-day features.

Poky Reference Distro
The base distro for a well tested system.
Levels of Separation

Yocto in Practice

DISTRO
musl
systemd
x11

MACHINE
kernel
bootloader
drivers

RECIPE
scripts
applications
support

IMAGE
console
manufacturing
kiosk web ui
Keep it Simple

SRC_URI

DEPENDS, RDEPENDS

EXTRA_OECONF, EXTRA_OEMAKE

FILES_*
Images
Start Small

You might not need the kitchen sink.
Start small and scale when you need it.
Choose the right tools

• devtool for recipe creating & editing

• wic for image creation & management

• see the scripts folder for more!
Why use the Yocto Project?
Yocto Builds Packages
So you don’t have to.

- **do_packagedata**
  Creates package metadata used by the build system to generate the final packages

- **do_package**
  Analyzes the content of the holding area and splits it into subsets based on available packages and files

- **do_package_write_rpm**
  Creates the actual RPM packages and places them in the Package Feed area

- **do_populate_lic**
  Writes license information for the recipe that is collected later when the image is constructed

- **do_populate_sysroot**
  Copies a subset of files installed by do_install into the sysroot in order to make them available to other recipes
• **do_populate_sysroot_setscene**
  Yocto cached your sysroots for you. You're welcome.

• **do_populate_lic_setscene**
  Because you're not changing your licensing between builds right?

• **do_package_write_rpm_setscene**
  No change to the package? Write it out again from cache.

• **do_packagedata_setscene**
  Same goes for the meta-data. Update the RPM package feed with cached data.
Yocto Ecosystem

Continuous Integration
autobuilder.yoctoproject.org

Tested Images & SDKs
downloads.yoctoproject.org

Patch Tests & Bug Tracking
patchwork.openembedded.org
bugzilla.yoctoproject.org

Ready-made Recipes
layers.openembedded.org
Yocto Project CI

**AUTOBUILDER**

- **X86**
- **ARM**
- **MIPS**
- **PPC**

---

**MINIMAL**

**X11**

**SATO (MATCHBOX)**

---

**Feature Tests**

- lsb
- musl
- no-X11
- systemd
- rpm
- intel-gpl
- ipk
- SysVinit
- wic
- deb
- ulibc
- eclipse plugins

---

**MULTIPLE TESTS**

- Multi Arch
- Multi Image
- Multi Feature
- Multiverse

---

**Releases**

**Build History**

**Error Reports**

**Image Tests**

**eSDK Tests**

**OE Self tests**
Before you ship-it...

runqemu
if you cannot emulate it it probably won’t boot

Image Tests (runtime)
now that it boots, will it work?

License Manifest File
build/tmp/deploy/licenses
Tune Your Builds

- Collect build metrics
- Identify bottlenecks
- Optimize builds for scale

https://wiki.yoctoproject.org/wiki/Build_Performance
Questions?

stephano.cetola@intel.com

@stephano

#OE #Yocto (IRC)