



Yocto Project ADT, Eclipse plug-in and Developer Tools

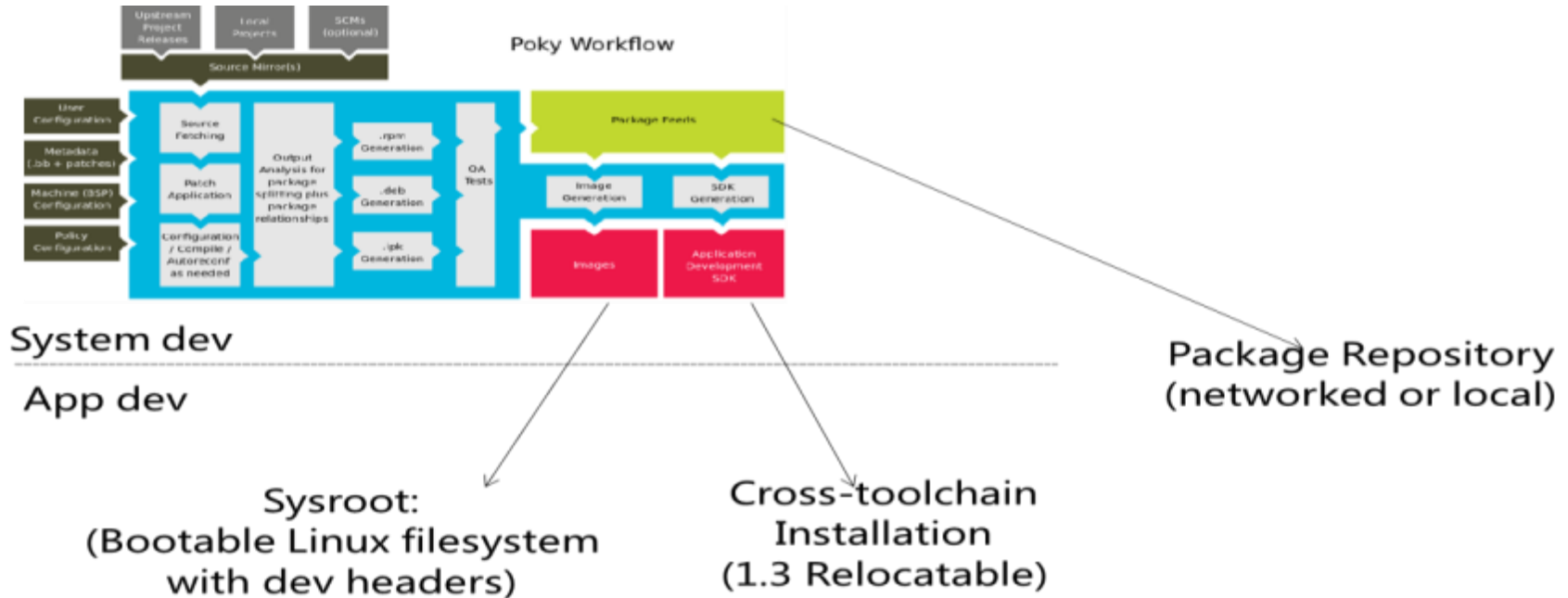
Jessica Zhang

LinuxCon - Japan • Tokyo • 2013

Agenda

- **The Application Development Toolkit**
- **Usage Flow And Roles**
- **Yocto Project Eclipse Plug-in**
- **Interacts With Yocto Project Tools**
- **Self Guided Hands-on Labs**
- **For Cross Develop Kernel Module**
- **Q&A**

The Application Development Toolkit



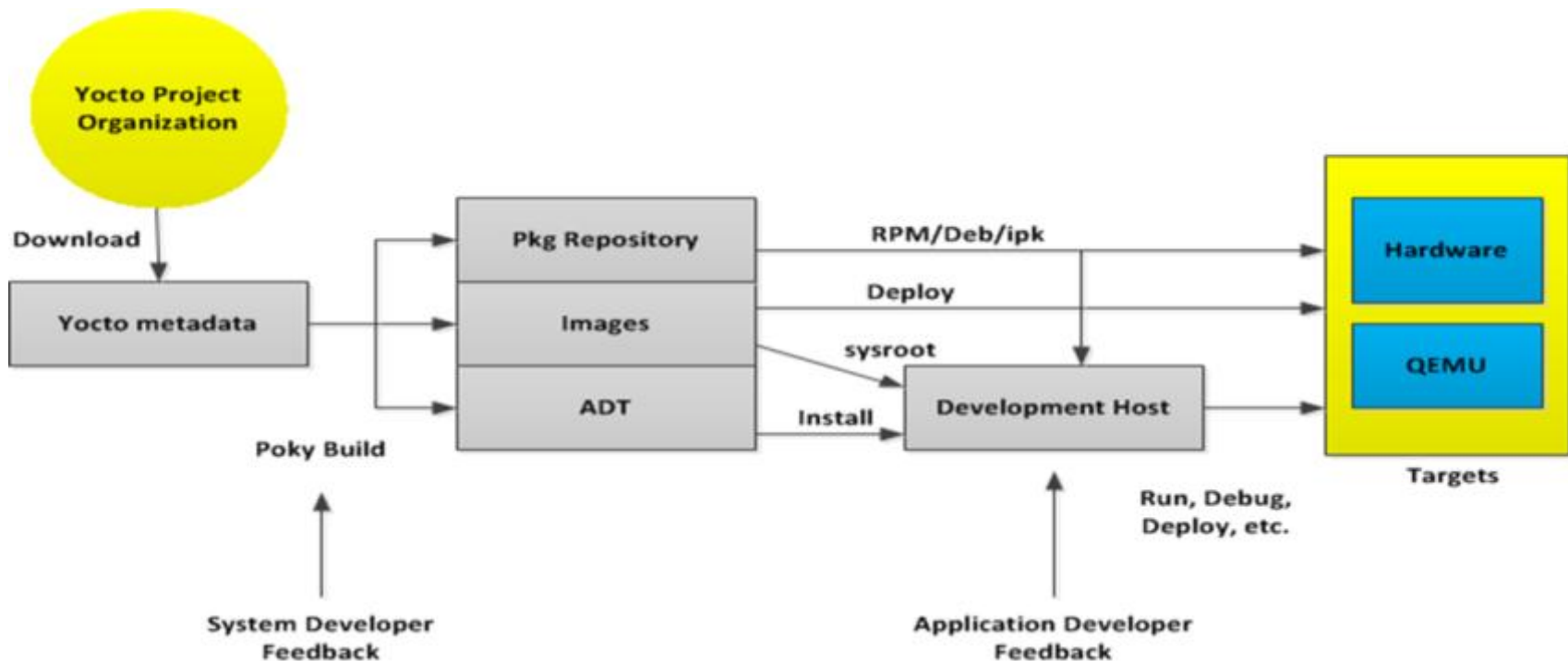
ADT Setup

- **Toolchain + sysroot**
- **SDK tarball**
 - Extract target rootfs as sysroot
- **ADT installer**
 - Customize sysroot and toolchain setup using a configuration file
 - Require a pre-setup repo
- **Embedded in your build tree**
 - Bitbake meta-ide-support
 - Extract target rootfs as sysroot

ADT Improvements

- **Relocatable**
 - Toolchain can be installed at user specified location
- **Generate image matching SDK**
 - `bitbake image-name -c populate_sdk`

Usage Flow And Roles

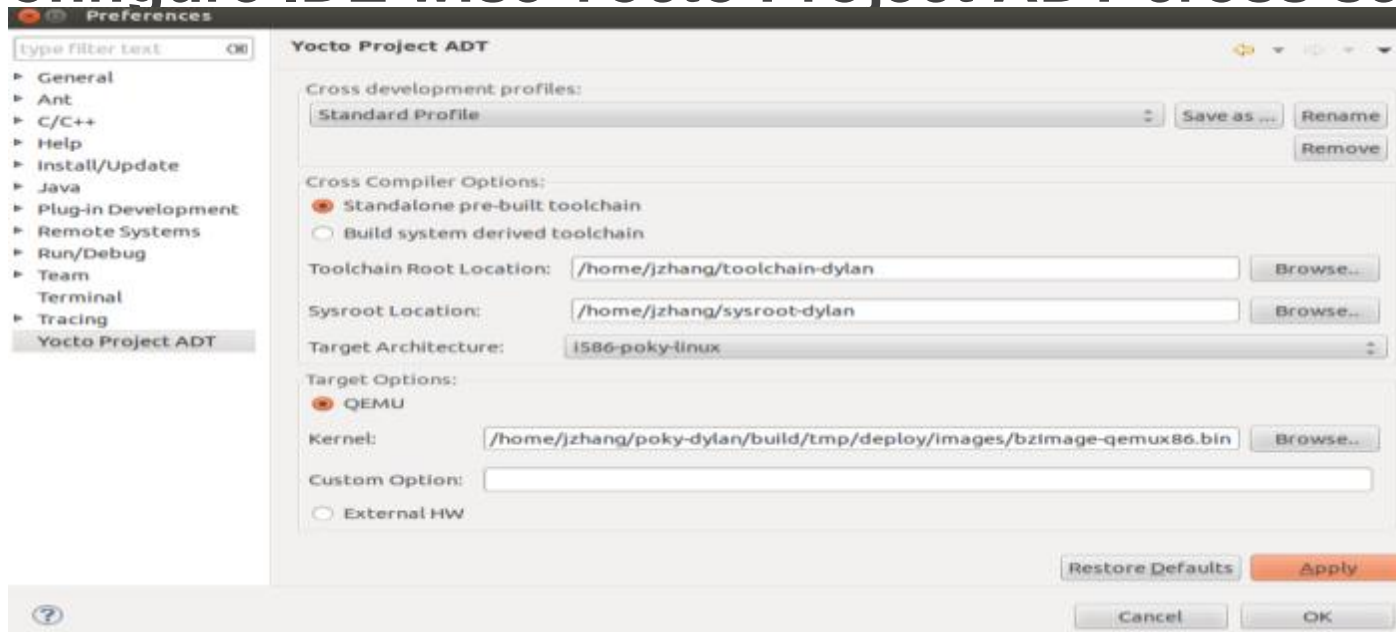


Yocto Project Eclipse Plug-in

- **Built on top via extensions of existing widely adopted eclipse plug-ins**
 - CDT, Linux Tools Project, RSE, TCF
- **Integrates Yocto Project ADT cross development setup into CDT**
 - For gnu autotool based projects
- **Leverage RSE over SSH or TCF and remote agent achieve remote target interaction within IDE**
 - CDT for cross platform build, deploy, run, debug
 - Remote tools interaction, e.g. powertop, latencytop, etc.

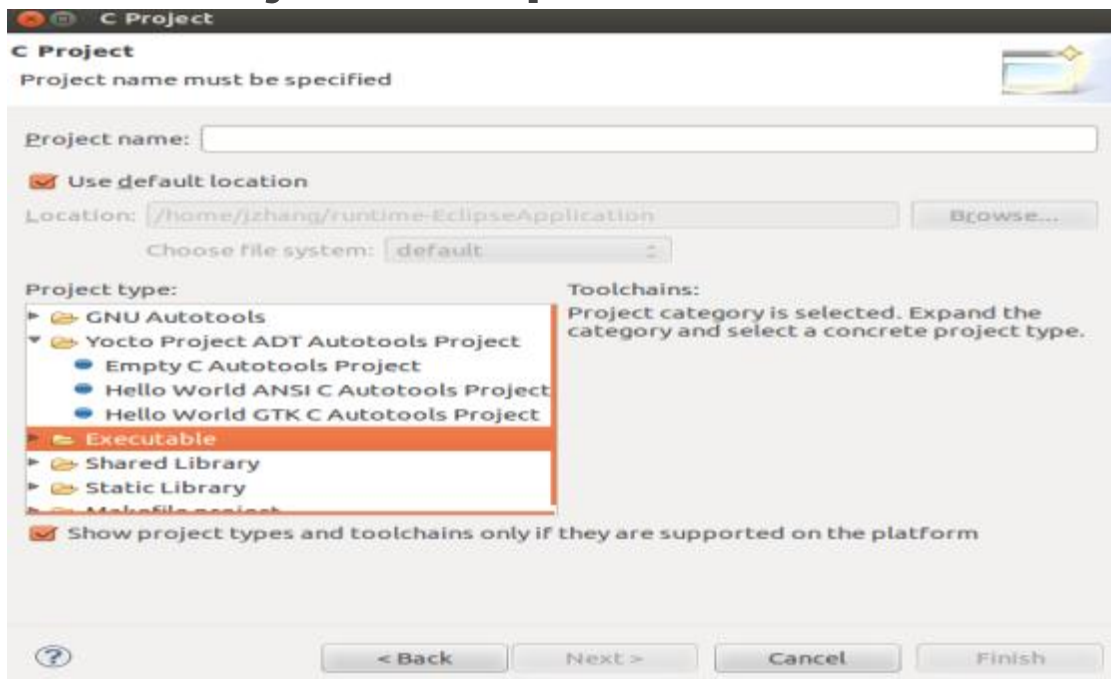
Eclipse IDE for Application Developer

- Configure IDE wise Yocto Project ADT cross setup



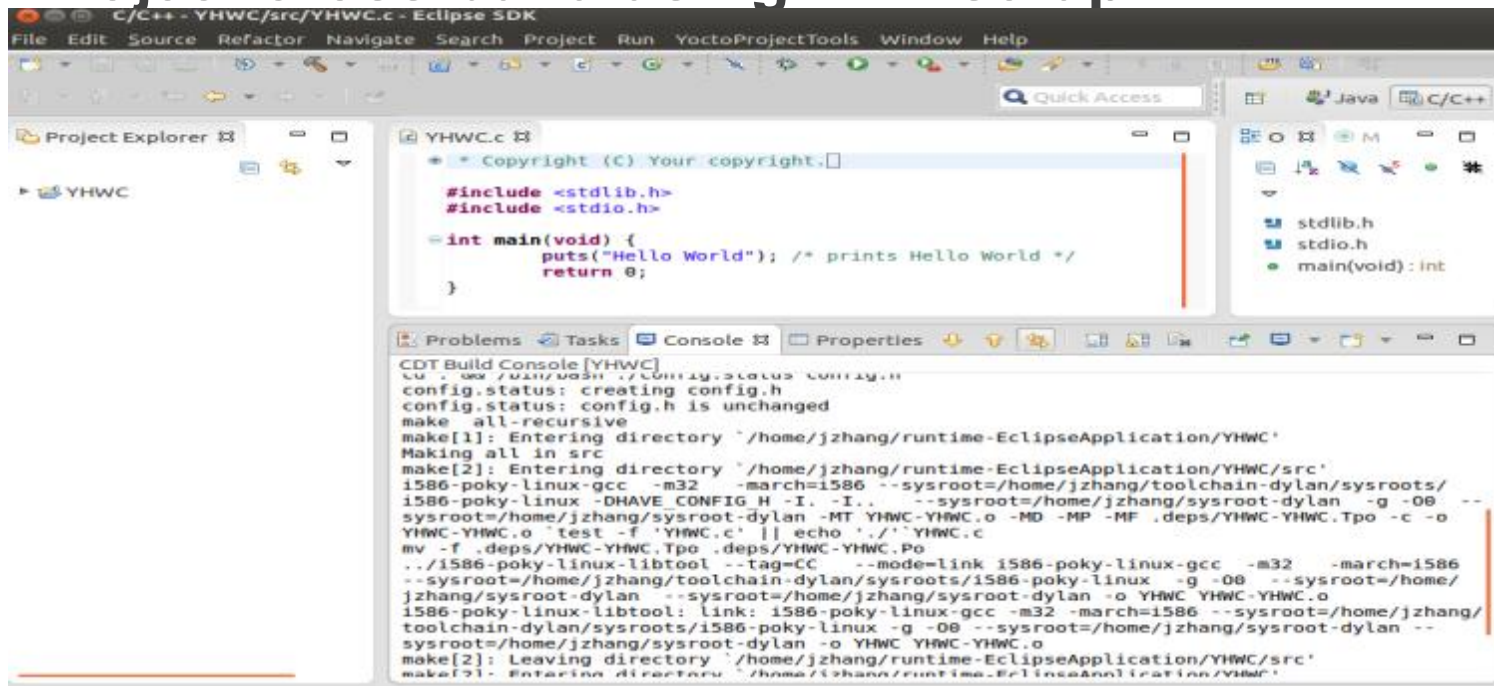
Eclipse IDE for Application Developer

- ADT Project Templates



Eclipse IDE for Application Developer

- Project cross build using ADT setup



The screenshot displays the Eclipse IDE interface for a C/C++ project. The main editor shows the source code for `YHWC.c`, which includes `stdlib.h` and `stdio.h`, and contains a `main` function that prints "Hello World". The Project Explorer on the left shows the project structure. The Console window at the bottom displays the output of a CDT Build Console, showing the configuration process and the successful compilation of `YHWC.o` using a cross-compiler for the 1586-poky-linux target.

```
C/C++ - YHWC/src/YHWC.c - Eclipse SDK
File Edit Source Refactor Navigate Search Project Run YoctoProjectTools Window Help

Project Explorer
YHWC

YHWC.c
* Copyright (C) Your copyright.

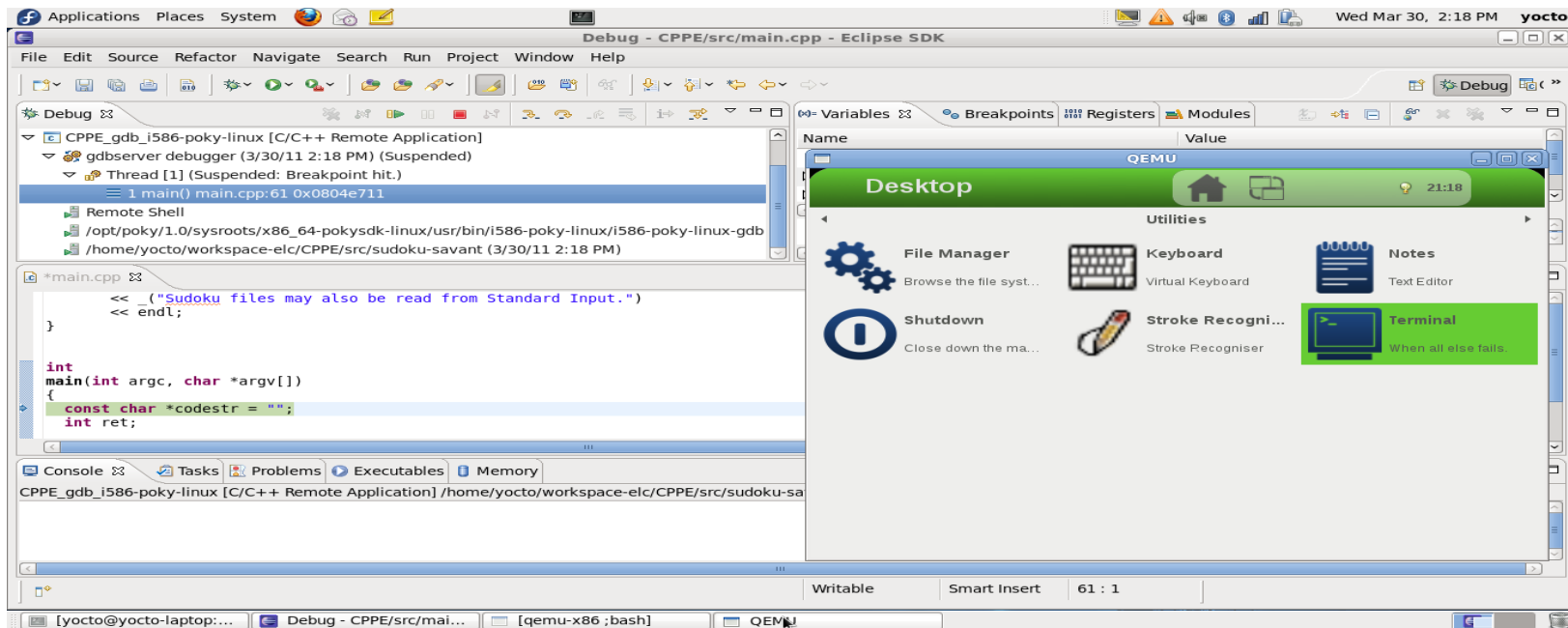
#include <stdlib.h>
#include <stdio.h>

int main(void) {
    puts("Hello World"); /* prints Hello World */
    return 0;
}

Problems Tasks Console Properties
CDT Build Console [YHWC]
cd /home/jzhang/.../config.status config.h
config.status: creating config.h
config.status: config.h is unchanged
make all-recursive
make[1]: Entering directory '/home/jzhang/runtime-EclipseApplication/YHWC'
Making all in src
make[2]: Entering directory '/home/jzhang/runtime-EclipseApplication/YHWC/src'
1586-poky-linux-gcc -m32 -march=1586 --sysroot=/home/jzhang/toolchain-dylan/sysroots/1586-poky-linux -DHAVE_CONFIG_H -I. -I... --sysroot=/home/jzhang/sysroot-dylan -g -O0 --sysroot=/home/jzhang/sysroot-dylan -MT YHWC-YHWC.o -MD -MP -MF .deps/YHWC-YHWC.Tpo -c -o YHWC-YHWC.o `test -f 'YHWC.c' || echo './`YHWC.c
mv -f .deps/YHWC-YHWC.Tpo .deps/YHWC-YHWC.Po
../1586-poky-linux-libtool --tag=CC --mode=link 1586-poky-linux-gcc -m32 -march=1586 --sysroot=/home/jzhang/toolchain-dylan/sysroots/1586-poky-linux -g -O0 --sysroot=/home/jzhang/sysroot-dylan --sysroot=/home/jzhang/sysroot-dylan -o YHWC YHWC-YHWC.o
1586-poky-linux-libtool: link: 1586-poky-linux-gcc -m32 -march=1586 --sysroot=/home/jzhang/toolchain-dylan/sysroots/1586-poky-linux -g -O0 --sysroot=/home/jzhang/sysroot-dylan --sysroot=/home/jzhang/sysroot-dylan -o YHWC YHWC-YHWC.o
make[2]: Leaving directory '/home/jzhang/runtime-EclipseApplication/YHWC/src'
make[1]: Entering directory '/home/jzhang/runtime-EclipseApplication/YHWC'
```

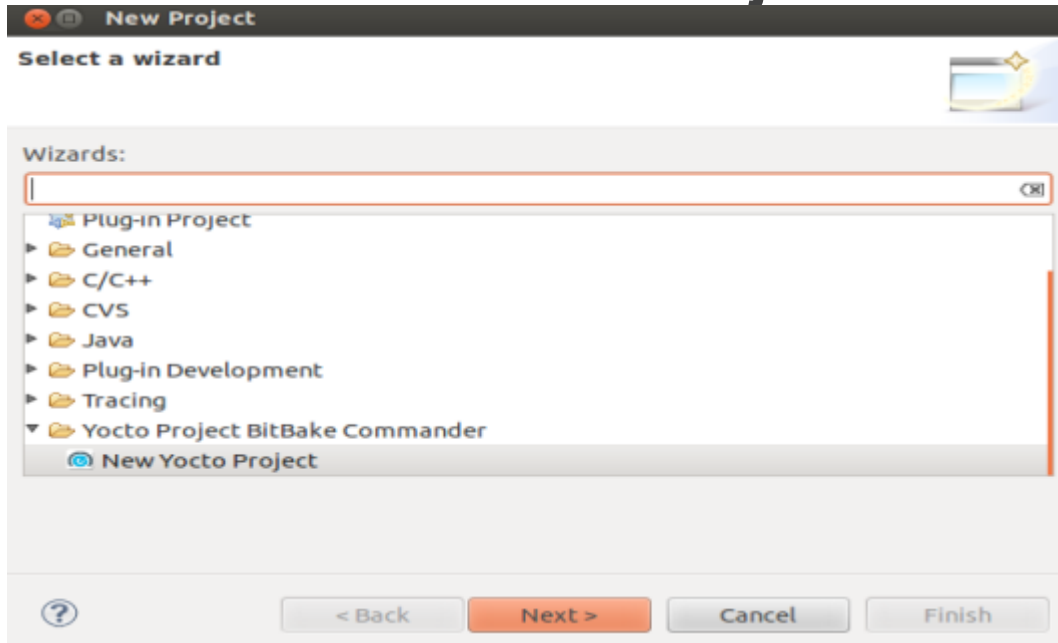
Eclipse IDE for Application Developer

- Cross debugging against qemu



Eclipse IDE for System Developer

- **Bitbake Commander Project**



Eclipse IDE for System Developer

- Meta data editor with syntax highlight and variable hover capability

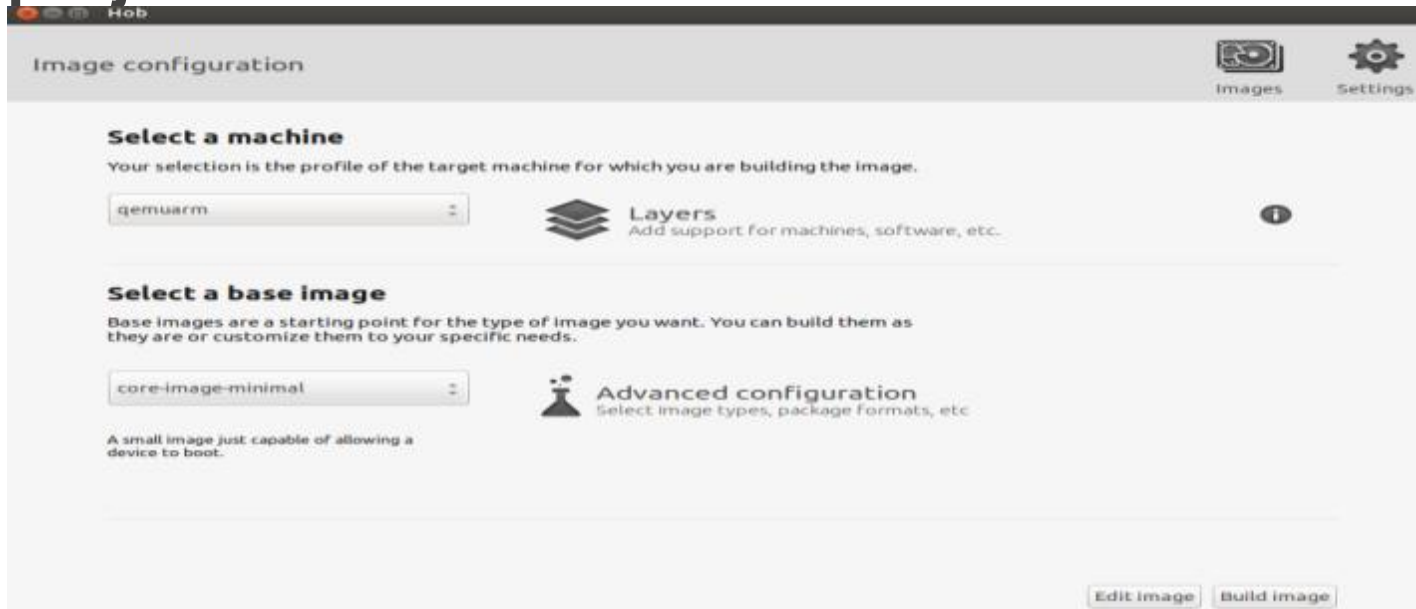


Interacts With Yocto Project Tools

- **Hob**
 - A GUI tool for bitbake for easy customize build configuration
- **Yocto-bsp**
 - A tool for quickly create a standard bsp layer, contains
 - Machine configuration, supporting recipes and README files.
 - allows the user to add (and remove) patches and kernel config fragments to a linux-yocto kernel without having to edit or learn the sordid details of the linux-yocto meta-data
- **Target Analytical Tools**
 - Powertop, latencytop, perf,oprofile,lttng,systemtap

Interacts with Yocto Project Tools

- Launch hob to build against bitbake commander project meta-data



Use Hob To Do Customization

- Edit Image to customize recipe and package selection

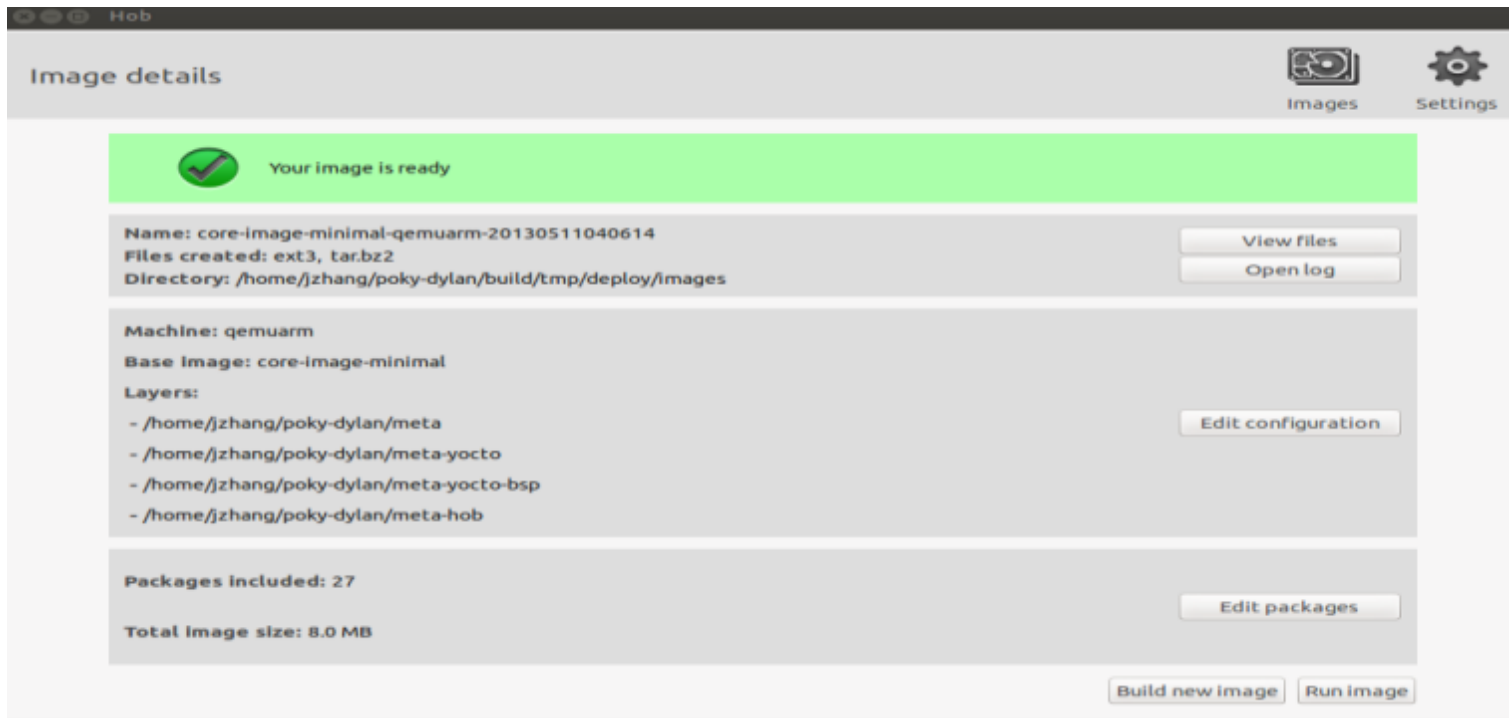
Step 1 of 2: Edit recipes

Included recipes **43** All recipes Package Groups Search recipes by name

Recipe name	Group	Brought in by (+others)	Included
acl	libs	udev	<input checked="" type="checkbox"/>
attr	libs	acl	<input checked="" type="checkbox"/>
base-files	base	packagegroup-core-boot	<input checked="" type="checkbox"/>
base-passwd	base	packagegroup-core-boot	<input checked="" type="checkbox"/>
binutils-cross	devel	gcc-cross-initial (+1)	<input checked="" type="checkbox"/>
busybox	base	packagegroup-core-boot	<input checked="" type="checkbox"/>
depmodwrapper-cross	base	core-image-minimal (+1)	<input checked="" type="checkbox"/>
eglibc	libs	run-postinsts (+28)	<input checked="" type="checkbox"/>
eglibc-initial	libs	eglibc	<input checked="" type="checkbox"/>
expat	libs	gettext	<input checked="" type="checkbox"/>
gcc-cross	devel	run-postinsts (+30)	<input checked="" type="checkbox"/>
gcc-cross-initial	devel	eglibc (+1)	<input checked="" type="checkbox"/>
gcc-runtime	devel	run-postinsts (+27)	<input checked="" type="checkbox"/>
gettext	libs	acl (+3)	<input checked="" type="checkbox"/>
glib-2.0	libs	udev	<input checked="" type="checkbox"/>
init-ifupdown	base	packagegroup-core-boot	<input checked="" type="checkbox"/>
initscripts	base	packagegroup-core-boot	<input checked="" type="checkbox"/>
kmod	base	udev	<input checked="" type="checkbox"/>
libffi	base	glib-2.0	<input checked="" type="checkbox"/>
libgcc	devel	gcc-runtime	<input checked="" type="checkbox"/>

Cancel Build packages


Use Hob To Build Image



The screenshot shows the Hob web interface for an image build. At the top, there are navigation icons for 'Images' and 'Settings'. The main content area is titled 'Image details' and features a green success banner with a checkmark icon and the text 'Your image is ready'. Below this, the image's metadata is displayed: Name: core-image-minimal-qemuarm-20130511040614, Files created: ext3, tar.bz2, and Directory: /home/jzhang/poky-dylan/build/tmp/ deploy/images. There are buttons for 'View files' and 'Open log' next to the directory path. The 'Machine' section shows 'qemuarm' and 'Base image: core-image-minimal'. Under 'Layers', a list of build directories is shown, with an 'Edit configuration' button to the right. The 'Packages included' section shows '27' packages and a 'Total image size' of '8.0 MB', with an 'Edit packages' button. At the bottom right, there are buttons for 'Build new image' and 'Run image'.

Image details

Images Settings

 Your image is ready

Name: core-image-minimal-qemuarm-20130511040614
Files created: ext3, tar.bz2
Directory: /home/jzhang/poky-dylan/build/tmp/ deploy/images

View files
Open log

Machine: qemuarm
Base image: core-image-minimal
Layers:

- /home/jzhang/poky-dylan/meta
- /home/jzhang/poky-dylan/meta-yocto
- /home/jzhang/poky-dylan/meta-yocto-bsp
- /home/jzhang/poky-dylan/meta-hob

Edit configuration

Packages included: 27
Total image size: 8.0 MB

Edit packages

Build new image Run image

Yocto-bsp plug-in

- Wizard like flow for ease of use

yocto-bsp Main page
Enter the required fields(with *) to create new Yocto Project BSP!

Metadata location*:

Build location:

BSP Name*:

BSP output location:

Kernel Architecture*:

Qemu Architecture(* for karch as qemu):

yocto-bsp Properties page

Kernel Settings:
Kernel:

Branch Settings:
Kernel branch:

Create a new branch from an existing one
 Use existing branch

Enable SMP support

BSP specific settings:
 keyboard
 touchscreen

Self Guided Hands-on Labs

- <https://www.yoctoproject.org/tools-resources/presentations/using-eclipse-yocto-project>

ADT And Eclipse For Cross Compile Kernel Modules

- Use cross toolchain and sysroot built by Yocto Project
- Sysroot kernel match target and must contain kernel devs
 - E.g. core-image-sato-sdk
 - Or in local.conf IMAGE_INSTALL_append = “ kernel_dev”
- Under `sysroot/usr/src/kernel` do:
 - make oldconfig ARCH=<arch>
 - make scripts ARCH=<arch>

ADT And Eclipse For Cross Compile Kernel Modules

- **In Eclipse IDE**

- New project -> C Project -> Makefile Project ->Empty Project (Linux GCC as Toolchains)
- In newly created project, e.g. hello, create 2 new files:

`hello.c`

Makefile

ADT And Eclipse For Cross Compile Kernel Modules

hello.c

```
#include<linux/kernel.h>
#include<linux/init.h>
#include<linux/string.h>

int hello_init(void)
{
    printk("Module named hello inserted\n");
    return 0;
}

void hello_exit(void)
{
    printk("Module named hello removed\n");
}

module_init(hello_init);
module_exit(hello_exit);
|
```

ADT And Eclipse For Cross Compile Kernel Modules

- **Makefile**

```
ARCH := x86
CROSS_COMPILE := i586-poky-linux-
obj-m := hello.o

KDIR := /home/jzhang/sysroot-kernel/usr/src/kernel

MAKEARCH := $(MAKE) ARCH=$(ARCH) CROSS_COMPILE=$(CROSS_COMPILE)

all:
    $(MAKEARCH) -C $(KDIR) M=$(PWD) modules

clean:
    $(MAKEARCH) -C $(KDIR) M=$(PWD) clean
```

For Cross Develop Kernel Modules

- **Cmdline**
 - source environment-setup file
 - echo \$PATH
- **Eclipse IDE**
 - Project Properties -> C/C++ Build Environment, Add PATH value equal your cmdline path value
 - Cross build your kernel module
 - Use RSE to deploy hello.ko
 - Remote terminal:
 - insmod hello.ko
 - rmmod hello
 - cat /proc/modules, tail /var/log/messages

Questions?



Yocto Project ADT, Eclipse plug-in and Developer Tools



yocto
PROJECT™


Yocto Project ADT, Eclipse plug-in and Developer Tools



Yocto Project ADT, Eclipse plug-in and Developer Tools



Yocto Project ADT, Eclipse plug-in and Developer Tools



**Thank you for your
participation!**

yocto ·
PROJECT

 THE
LINUX
FOUNDATION