

# A Smart Way to Manage Packages in Yocto Project

Apr 4th, 2016

Fan Xin, Fujitsu Computer Technologies Limited

- Fan Xin, Fujitsu Computer Technologies Ltd.
- Embedded Linux Developer
- In-House Embedded Linux Distributor of Fujitsu
- Our Distribution includes LTSI Kernel and is built with Yocto Project
- Our Distribution is used for
  - IVI, Server System Controller, Storage System, Network Equipment, Printer, etc.



IVI: In-Vehicle Infotainment

# Fujitsu's contribution for Yocto Project

meta-openembedded.git

top contributors by employer (2015-01-01 to 2015-12-31)

|                         |             |
|-------------------------|-------------|
| None                    | 625 (46.1%) |
| Wind River              | 309 (22.8%) |
| Fujitsu                 | 160 (11.8%) |
| Intel                   | 82 (6.0%)   |
| Mentor Graphics         | 34 (2.5%)   |
| O.S. Systems            | 24 (1.8%)   |
| National Instruments    | 19 (1.4%)   |
| MontaVista              | 16 (1.2%)   |
| OpenDreambox            | 12 (0.9%)   |
| Freescale               | 9 (0.7%)    |
| Linaro                  | 8 (0.6%)    |
| ENE A/B                 | 5 (0.4%)    |
| Leica                   | 5 (0.4%)    |
| Aker Security Solutions | 4 (0.3%)    |
| BMW                     | 3 (0.2%)    |

- Package Manager Comparison
- Package Management Problem Analysis in Yocto Project
- Introduction of Smart Package Manager
- Fujitsu's Contribution and Next Step

# Package Manager Comparison

## ■ Package Manager

- A collection of software tools that automates the process of installing, upgrading, configuring, and removing computer programs
- Deals with packages, distributions of software and data in archive files
- Maintain a database of software dependencies and package information

## ■ Common Package Management System

- rpm, deb, ipkg, opkg
- base of package manager

## ■ Common Package Managers

- Advanced Packaging Tool (APT)
- Yellowdog Updater Modified (YUM)
- Dandified Yum (DNF)

[https://en.wikipedia.org/wiki/Package\\_manager](https://en.wikipedia.org/wiki/Package_manager)

[https://en.wikipedia.org/wiki/List\\_of\\_software\\_package\\_management\\_systems](https://en.wikipedia.org/wiki/List_of_software_package_management_systems)

- APT ( Advanced Packaging Tool )
- The Advanced Package Tool, or APT, is a free software user interface that works with core libraries to handle the installation and removal of software
- Used on Debian Linux distribution and its variants.
- Depends on deb package format
- Works with repositories, which are collections of packages,

# Yellowdog Updater Modified (YUM)

- A package manager for automatic install, updates, uninstall package and dependency management.
- Originally developed to manage Red Hat Linux systems at Duke University's Physics department
- Used on RPM-based distributions, such as RedHat
- Yum depends on RPM
- Works with repositories, which are collections of packages,



[https://en.wikipedia.org/wiki/Yellowdog\\_Updater\\_Modified](https://en.wikipedia.org/wiki/Yellowdog_Updater_Modified)



# Dandified Yum (DNF)

- Next generation version of the Yellowdog Updater Modified (yum)
- Using RPM, libsolv and hawkey libraries
- Introduced in Fedora 18 and become the default package manager for Fedora since version 22

# Comparison of Package Managers

| Items                  |      | Apt              | Yum                                  | DNF                          |
|------------------------|------|------------------|--------------------------------------|------------------------------|
| Latest Version         |      | 1.1.10           | 3.4.3                                | 1.1.4                        |
| Support Package Format | .deb | √                |                                      |                              |
|                        | .rpm | √(apt-rpm)       | √                                    | √                            |
| Distribution           |      | Debian<br>Ubuntu | RHEL<br>CentOS<br>Fedora<br>OpenSUSE | Fedora<br>(Since version 22) |
| License                |      | GNU GPL          | GPL v2                               | GPL v2                       |
| Website                |      | www.apt-rpm.org  | yum.baseurl.org                      | dnf.baseurl.org              |

[https://en.wikipedia.org/wiki/Advanced\\_Packaging\\_Tool](https://en.wikipedia.org/wiki/Advanced_Packaging_Tool)

[https://en.wikipedia.org/wiki/Yellowdog\\_Updater,\\_Modified](https://en.wikipedia.org/wiki/Yellowdog_Updater,_Modified)

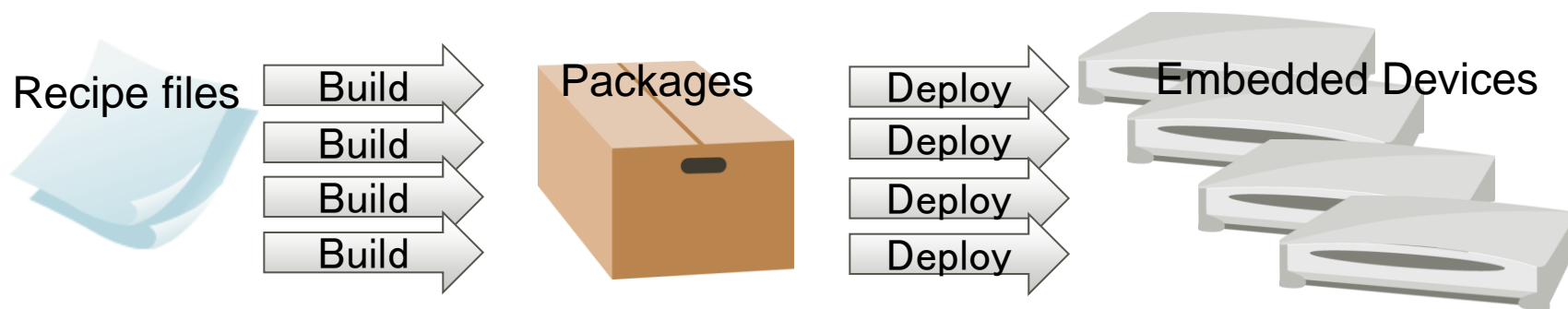
[https://ja.wikipedia.org/wiki/DNF\\_\(%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2\)](https://ja.wikipedia.org/wiki/DNF_(%E3%82%BD%E3%83%95%E3%83%88%E3%82%A6%E3%82%A7%E3%82%A2))

# Comparison of Package Management Systems

| Items                             | rpm         | deb            | opkg                                     | ipkg                                 |
|-----------------------------------|-------------|----------------|--|--------------------------------------|
| Check Package License Information | √           |                |  |                                      |
| File format                       | .rpm        | .deb<br>.udeb  | .ipk                                     | .ipk                                 |
| Used by OpenEmbedded              | √           |                | √  | √                                    |
| Develop Status                    | Active      | Active         | Active ( fork from ipkg )                | Discontinued                         |
| License                           | GPL         | GPL            | GPL v2                                   | GPL v2                               |
| Website                           | www.rpm.org | www.debian.org | git.yoctoproject.org/cgit/cgit.cgi/opkg/ | www.handhelds.org/moin/moin.cgi/lpkg |

[https://en.wikipedia.org/wiki/RPM\\_Package\\_Manager](https://en.wikipedia.org/wiki/RPM_Package_Manager)  
[https://en.wikipedia.org/wiki/Deb\\_\(file\\_format\)](https://en.wikipedia.org/wiki/Deb_(file_format))  
<https://en.wikipedia.org/wiki/Opkg>  
<https://ja.wikipedia.org/wiki/Lpkg>

# Present Situation in Yocto Project



## ■ Flexibility

- Flex to select package before building

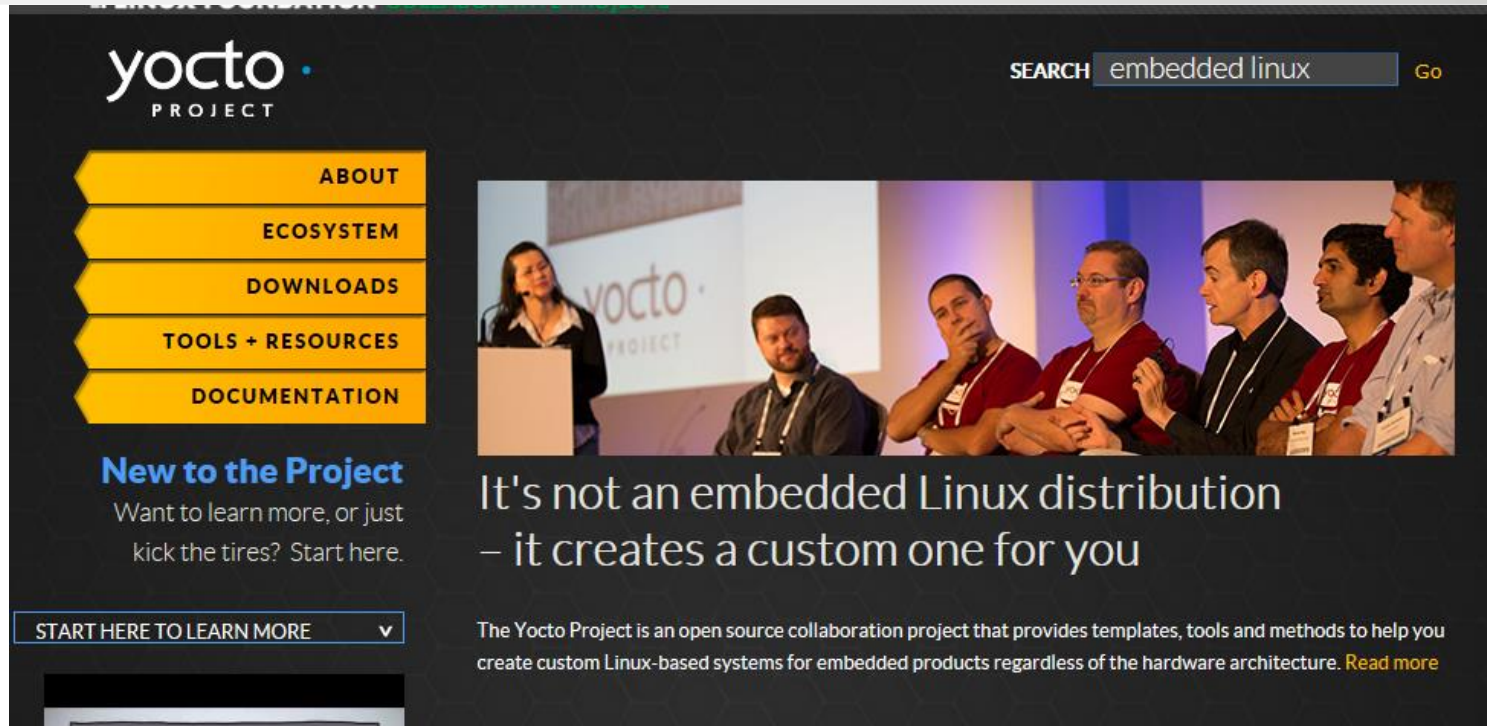
## ■ Efficiency

- Create root file system, binary packages (rpm, ipkg, opkg)

## ■ Optimization

- Easy to optimize for each distribution

# Package Management Problem Analysis in Yocto Project



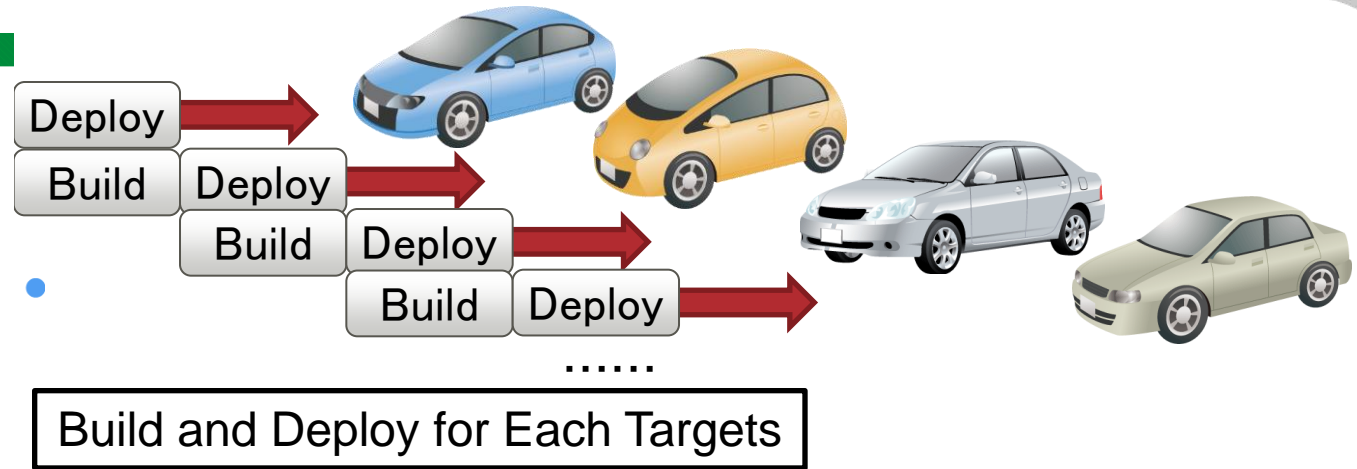
“ Yocto creates a **custom** one for you ”

- The Advantage of custom embedded Linux
- The value of custom
- The balance between custom and minimize fragmentation

- Can not know what package is built due to the packages dependence.
- Have to rebuild the root file system every time when have some updates or modifications for some packages.
- Produce fragmentation

**Flexibility** : Prefer graphical interface and detailed info  
**Efficiency** : Inefficiency, in some degree  
**Optimization** : Produce fragmentation

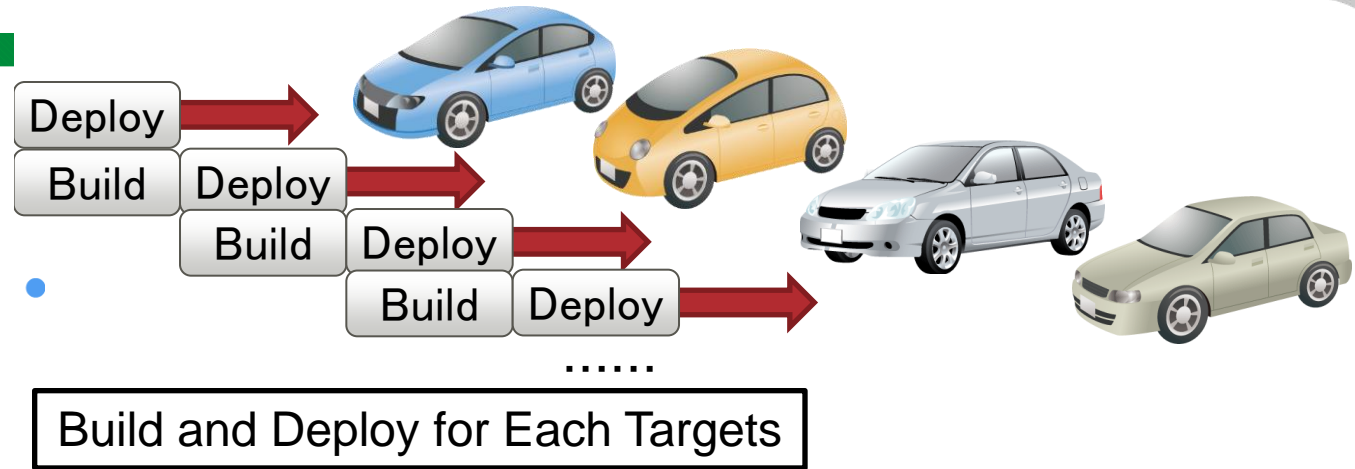
LINUX FOUNDATION COLLABORATIVE PROJECTS



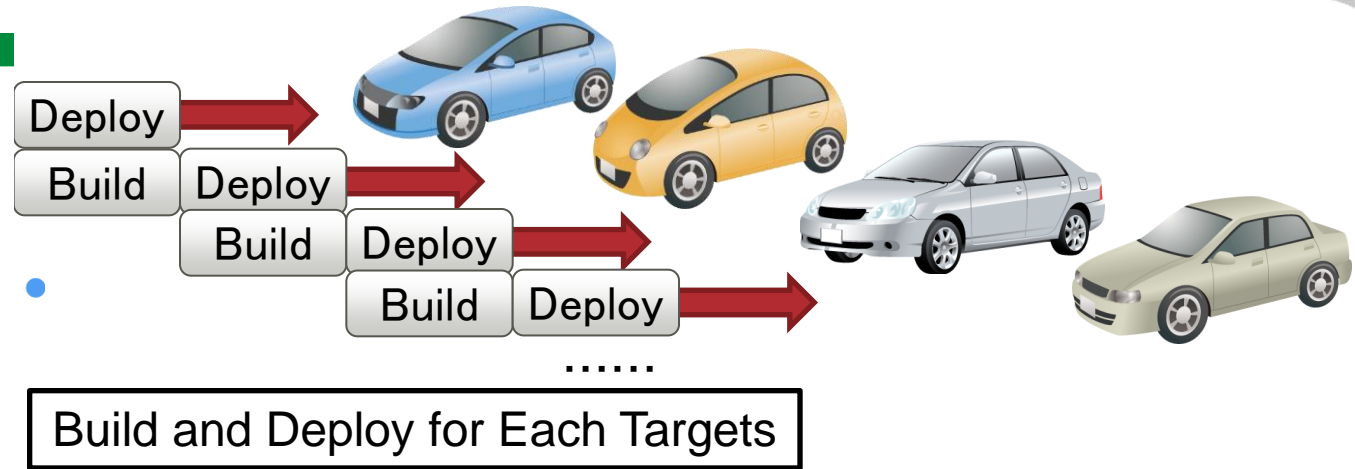
These cars use same architecture and own individual build directory.



□ LINUX FOUNDATION COLLABORATIVE PROJECTS



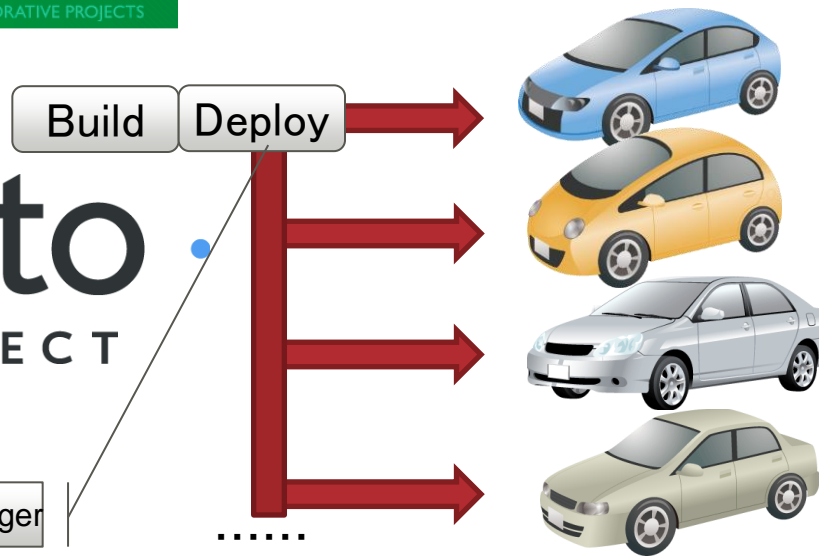
LINUX FOUNDATION COLLABORATIVE PROJECTS



LINUX FOUNDATION COLLABORATIVE PROJECTS



Using Package Manager



## Simulation

Full build time: 5~6 hours  
Deploy time: 1 hour  
Target Type: 50

Before:  
 $\text{Cost} = (5+1) \times 50 = 300 \text{ hours}$   
After:  
 $\text{Cost} = 5 + 1 \times 50 = 55 \text{ hours}$

# Cost Comparison



<https://access.redhat.com/articles/2161461>

Build glibc package and deploy for car A

Build glibc package and deploy for car B

Build glibc package and deploy for car C

Build glibc package and deploy for car A, B, C

- Feb 17<sup>th</sup>, glibc public critical security flaw. This vulnerability should be fixed in all targets immediately.
- For each kind of target, the glibc package has to be built and deployed.

## Simulation

Cost = ( Build Time + Deploy Time ) \* Device Number

= ( 1 + 0.2 ) \* 50 = 60 hours

Cost = Build Time + Deploy Time \* Device Number

= 1 + 0.2 \* 50 = 11 hours

Glibc Build Time: 1 hours  
Deploy Time: 0.2 hours  
Device Number: 50 kinds

# What we should do

Rebuild the root file system  
and redeploy them into production  
every time ?

Use the package manager to manage  
individual package efficiently?



# Introduction of Smart Package Manager

# Smart Package Manager Overview

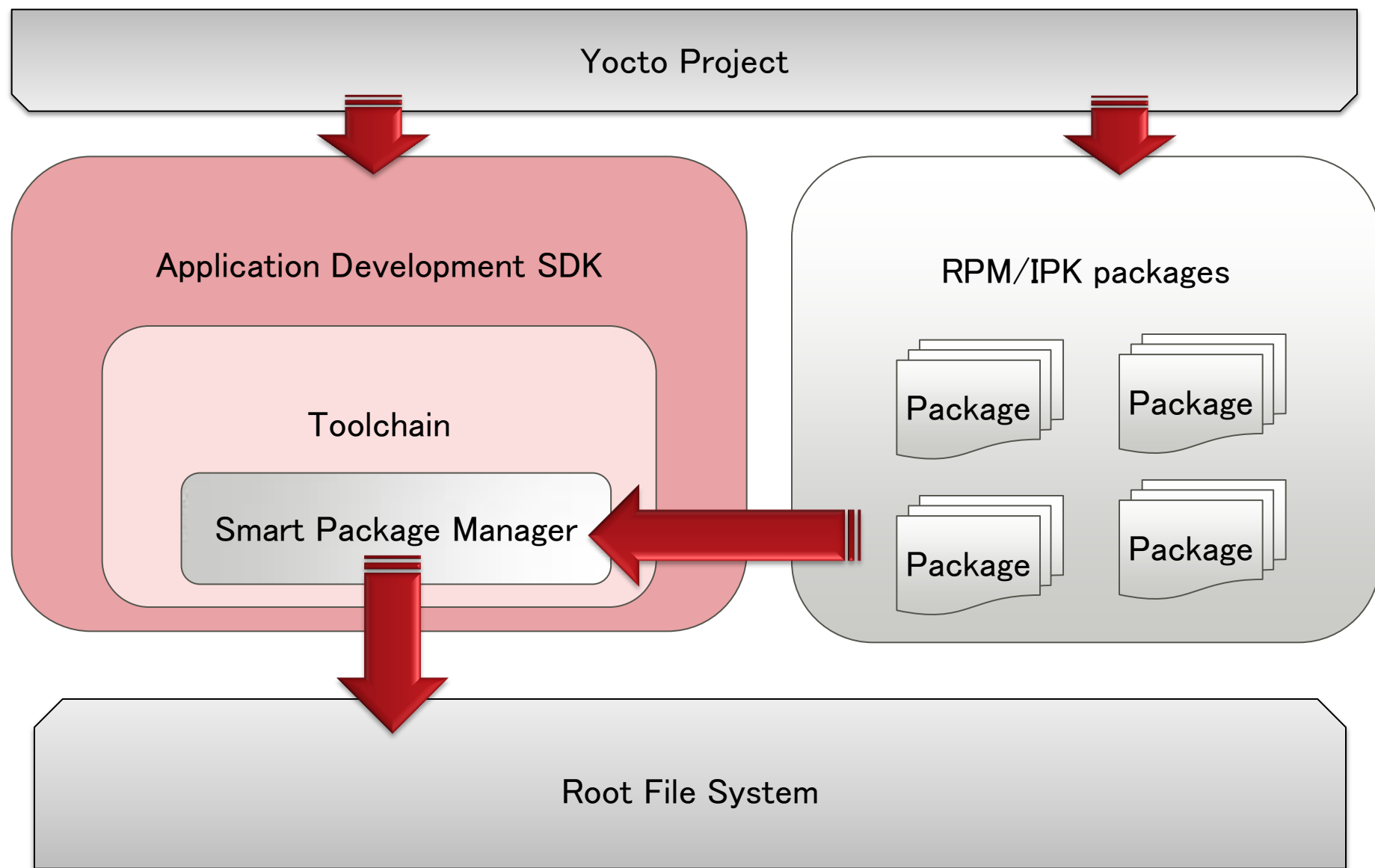
- Aim to create smart and portable algorithms for solving adequately the problem of managing software upgrading and installation.
- Homepage: [smartpm.github.io/smart](http://smartpm.github.io/smart)
- License: GPL v2
- Works with APT, APT-RPM, YUM, etc.
- Started on May 4<sup>th</sup>, 2004, and version 1.0 was released on Aug 14<sup>th</sup>, 2008
- The latest version is 1.5
- Merged into Yocto Project 1.4 (dylan), but unavailable in Yocto Project so far

<https://launchpad.net/smart>

<http://smartpm.github.io/smart/>

<https://www.yoctoproject.org/blogs/khem/2013/get-smart-smart-package-manager>

# Smart Package Manager Workflow



## ■ Modular

- Support PRM, DPKG, and Slackware package management systems

## ■ Smart Transactions

- Thousands of packages and relations are being considered
- Not only find a solution, but find the best solution (install, remove, upgrade, etc).

## ■ Channels

- Many different channel types are supported

## ■ Priority Handing

- Hand integration of multiple channels and setup preferred package versions

## ■ Downloading Mechanism

- Fast parallel downloading mechanism, multiple connections

## ■ ...

<http://smartpm.github.io/smart/>



Smart package manager uses the following command to manage packages.

Usage: smart command

## ■ Action commands

- update, install, reinstall, upgrade, remove, check, fix, download, clean

## ■ Query commands

- search, query, newer, info, stats

## ■ Setup commands

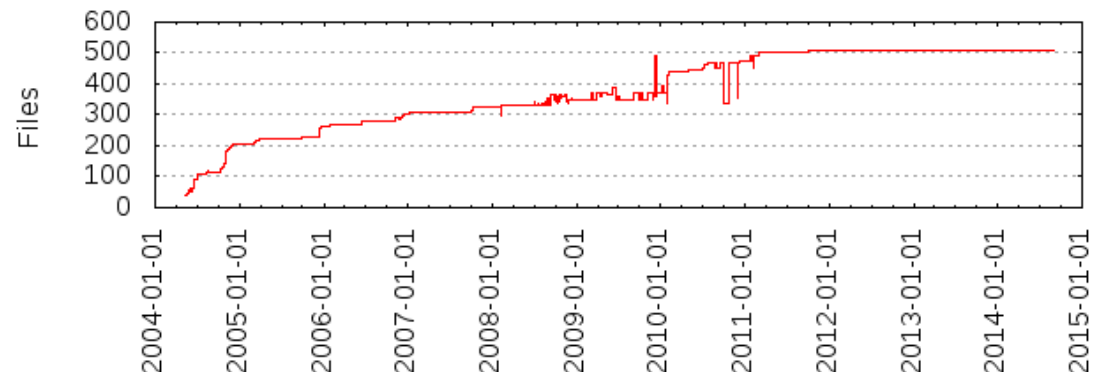
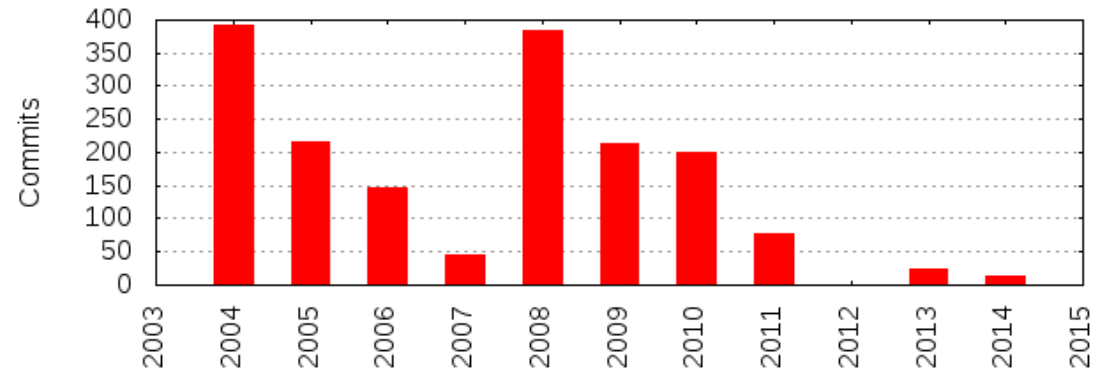
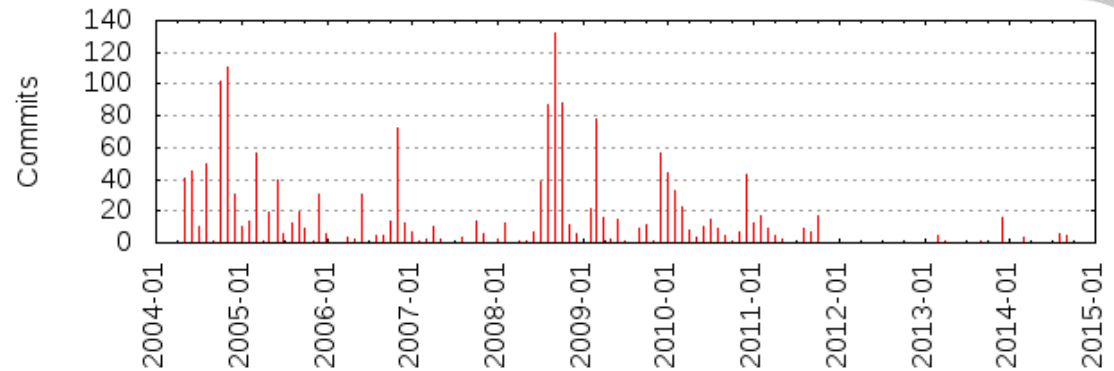
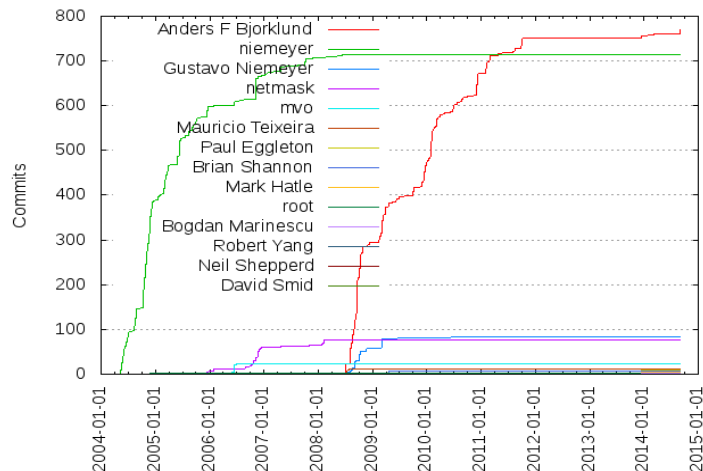
- config, channel, priority, mirror flag

Run “smart command –help ” for more information

# Project Statistics

Statistics about  
Commit change by year month  
Commit change by year, author  
Commit change by author  
File change by year

| Tag Name | Date       | Commits |
|----------|------------|---------|
| 1.5      | 2014-09-05 | 106     |
| 1.4      | 2011-03-03 | 228     |
| 1.3      | 2010-02-12 | 479     |
| 1.2      | 2009-03-05 | 43      |
| 1.1      | 2008-09-09 | 18      |
| 1.0      | 2008-08-15 | 836     |



- Smart still has some problems needed to solve before using it.

## ■ Problem #1

- There is bug in the rpm and smart command in default toolchain.
- Send the patch to community to fix it
- <http://patchwork.openembedded.org/patch/106097/>

## ■ Problem #2

- Need to configure the environment before using smart, or it can not use normally
- Prepare the script to configure environment automatically

## ■ Problem #3

- Smart Package Manager uses GTK/Qt4 as the graphical framework, but GTK/Qt4 is disabled in toolchain
- Develop Text GUI with python to make smart user-friendly and easy to use

```
Package Installer
|-----| Select package |-----|
[*] zip-dev
[*] zip
[*] zabbix-dev
[*] zabbix
[*] xz-staticdev
[*] xz-locale-vi
[*] xz-locale-pl
[*] xz-locale-it
[*] xz-locale-fr
[*] xz-locale-de
[*] xz-locale-cs
[*] xz-dev
[*] xz
[*] xwud-dev
[*] xwud
[*] xwininfo-dev
[*] xwininfo
[*] xwd-dev
[*] xwd
-----
All Package [6613]   Installed Packages [0] Selected Packages [6613]
-----
SPACE/ENTER:select/unselect  R:seaRch N:Next  B:Back  I:Info  X:eXit
```

# Smart Package Manager Usage

## ■ Install toolchain

```
./libc-x86-meta-toolchain-i586-toolchain.sh
```

## ■ Initial environment

```
./opt/environment-setup-core2-64-linux
```

We use this script to finish the environment initialization,  
such as environment variable setting

## ■ Install package into root file system

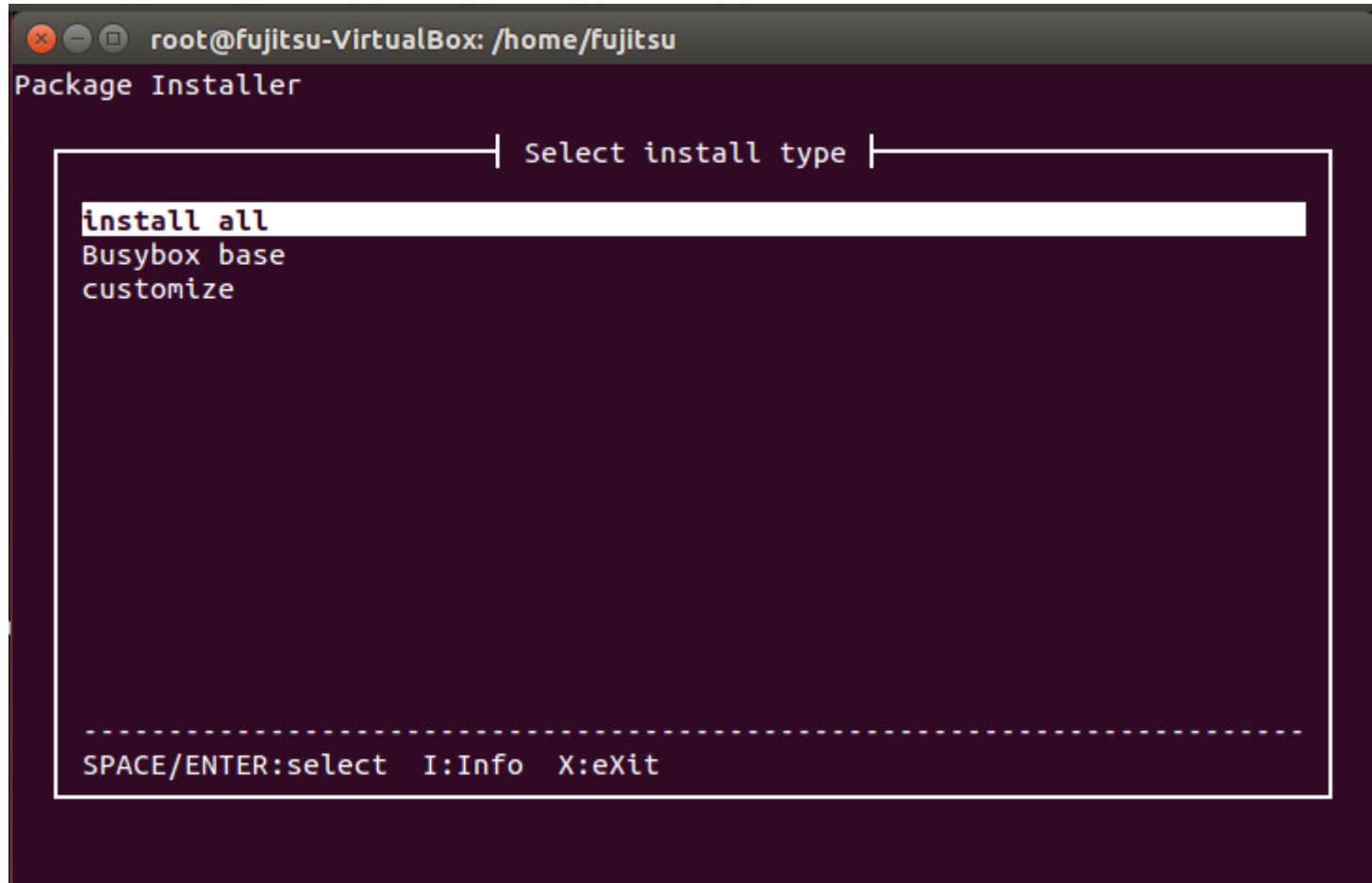
### ■ Command line interface

```
smart --data-dir=${rootfs-dir}/var/lib/smart install
```

### ■ Graphical interface

```
smart --data-dir=${rootfs-dir}/var/lib/smart --interface=tgui
```

# Select install type



```
root@fujitsu-VirtualBox: /home/fujitsu
Package Installer

|-----| Select install type |-----|

install all
Busybox base
customize

-----
SPACE/ENTER:select  I:Info  X:eXit
```

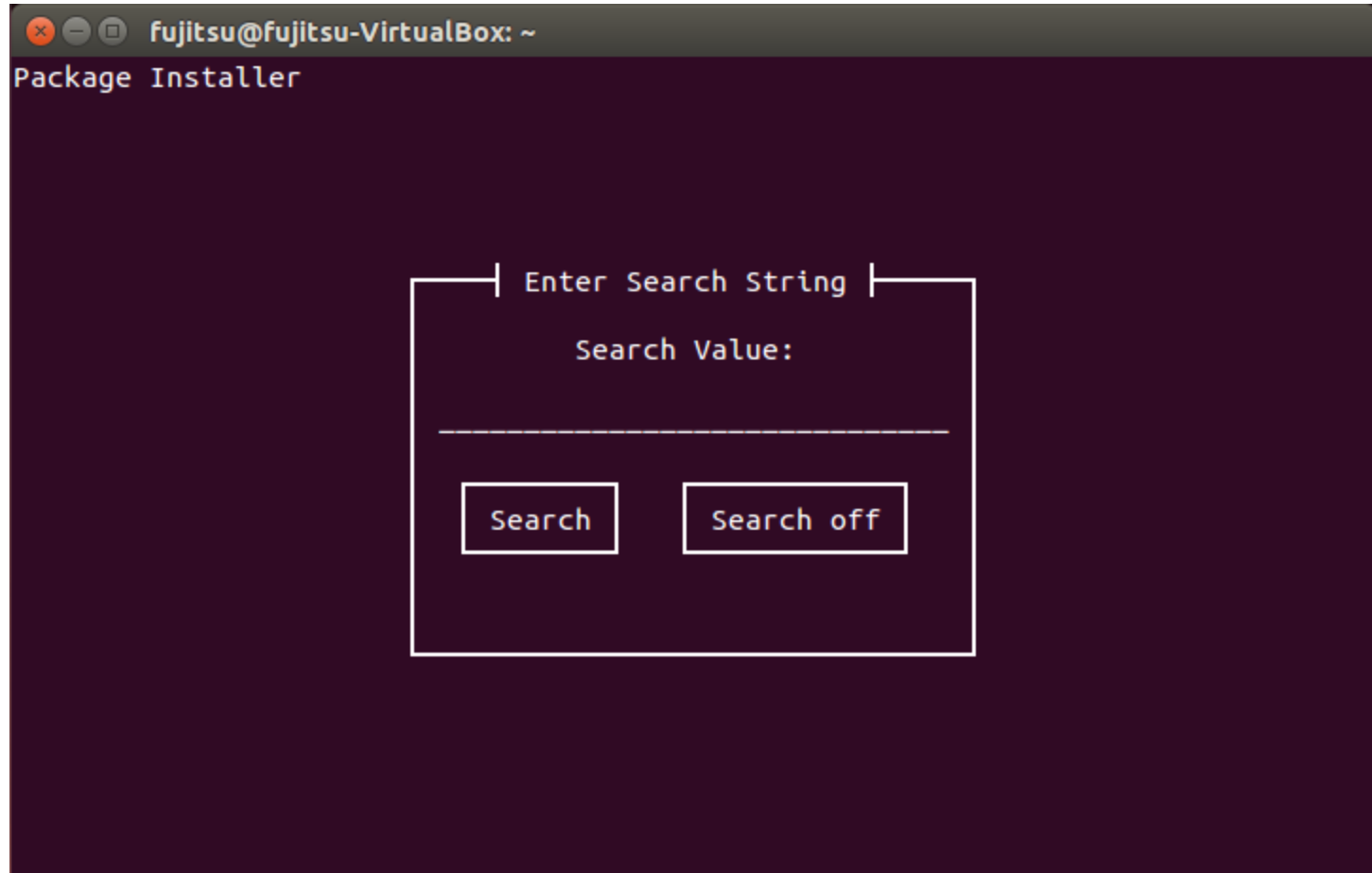
# Select packages

```
root@fujitsu-VirtualBox: /home/fujitsu
Package Installer

|----- Select package -----|

[ ] bzip2-staticdev
[ ] bzip2-ptest
[ ] bzip2-dev
[ ] bzip2
[ ] byacc-dev
[ ] byacc
[*] busybox-udhcpd
[*] busybox-udhcp
[*] busybox-syslog
[*] busybox-ptest
[*] busybox-hwclock
[*] busybox-httpd
[*] busybox-dev
-----
All Package [6613]    Installed Packages [0] Selected Packages [8]
-----
SPACE/ENTER:select/unselect  R:seaRch N:Next  B:Back  I:Info  X:exit
```

# Search package





# Select dbg packages

```
fujitsu@fujitsu-VirtualBox: ~  
Package Installer  
  
Select debuginfo package  
  
[ ] zip-dbg  
[ ] zabbix-dbg  
[ ] xz-dbg  
[ ] xwud-dbg  
[ ] xwininfo-dbg  
[ ] xwd-dbg  
[ ] xvinfo-dbg  
[ ] xtrans-dbg  
[ ] xterm-dbg  
[ ] xstdcmap-dbg  
[ ] xsetroot-dbg  
[ ] xsetmode-dbg  
[ ] xset-dbg  
  
-----  
All Package [792]    Installed Packages [1] Selected Packages [0]  
-----  
SPACE/ENTER:select/unselect  N:Next  B:Back  I:Info  X:eXit
```

# Installing

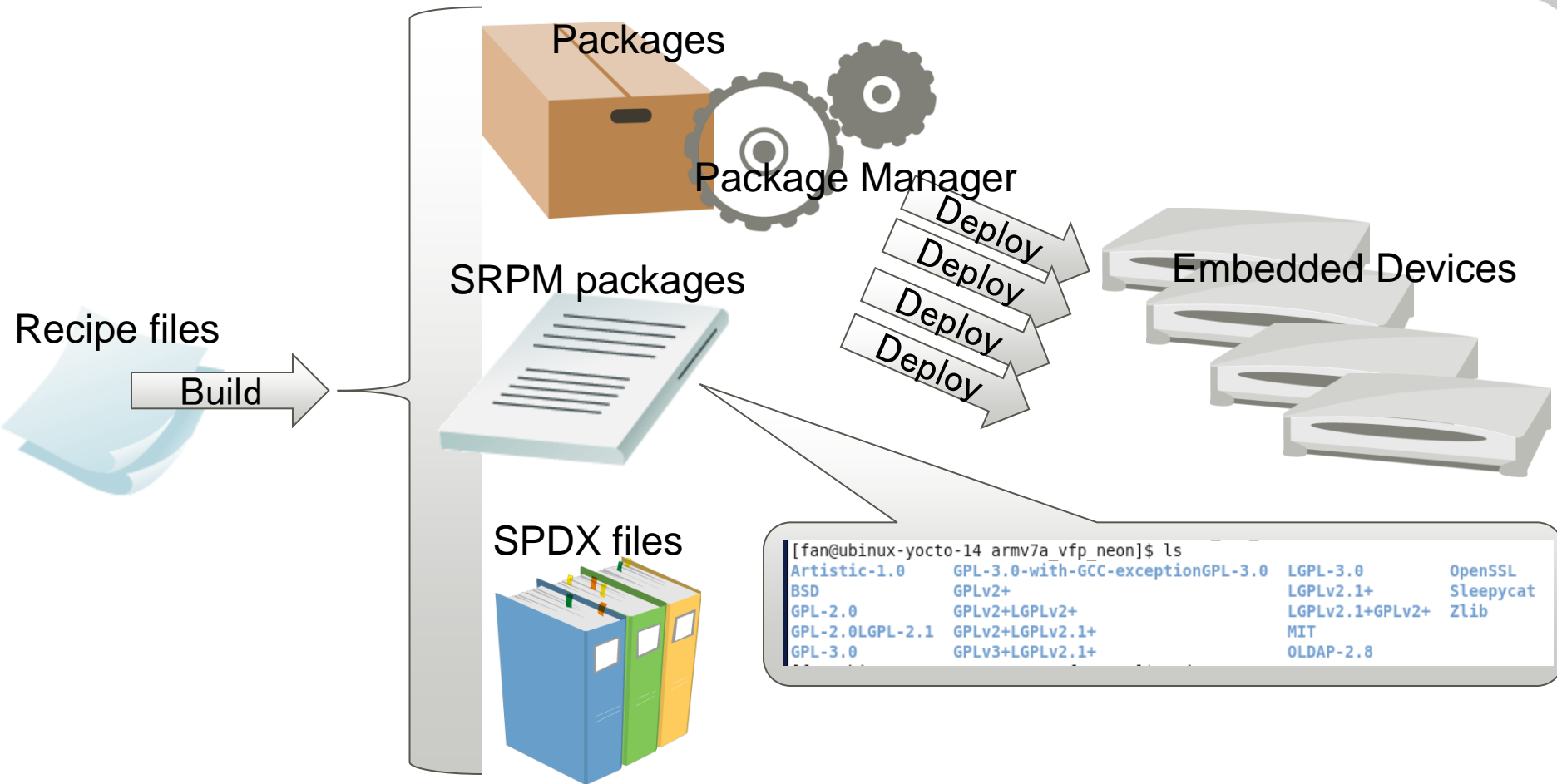
```
root@fujitsu-VirtualBox: /home/fujitsu
491:Installing gconf-locale-.. ##### [ 7%]
492:Installing glibc-charmap.. ##### [ 7%]
493:Installing mc-locale-wa ##### [ 7%]
494:Installing binutils-loc.. ##### [ 7%]
495:Installing glibc-charmap.. ##### [ 7%]
496:Installing glibc-binaries ##### [ 7%]
497:Installing findutils-loc.. ##### [ 7%]
498:Installing python-smartp.. ##### [ 7%]
499:Installing flex-locale-tr ##### [ 7%]
500:Installing libpopt-local.. ##### [ 7%]
501:Installing indent-locale.. ##### [ 7%]
502:Installing binutils-loc.. ##### [ 7%]
503:Installing sed-locale-ro ##### [ 7%]
504:Installing cracklib-loc.. ##### [ 7%]
505:Installing psmisc-locale.. ##### [ 7%]
506:Installing gtk+-locale-cs ##### [ 7%]
507:Installing networkmanage.. ##### [ 7%]
508:Installing tar-locale-de ##### [ 7%]
509:Installing libglib-2.0-l.. ##### [ 7%]
510:Installing tzdata-atlantic ##### [ 7%]
511:Installing libatk-1.0-lo.. ##### [ 7%]
512:Installing imsettings-lo.. ##### [ 7%]
513:Installing kbd-locale-el ##### [ 7%]
```

# Created Root Filesystem

```
fujitsu@fujitsu-VirtualBox: ~  
fujitsu@fujitsu-VirtualBox:~$ ls work/smart-rootfs-x86-full-rpm/  
bin  dev  home  lib      media  oe_install  proc  sbin  sys  usr  www  
boot  etc  init  linuxrc  mnt     opt         run  srv  tmp  var  
fujitsu@fujitsu-VirtualBox:~$
```



# Conclusion and Next Step



- Accompanied with the package files, SRPM packages and SPDX files are created to manage license information
- SRPM Packages are classified by license type

- Send our patches to community, but not accepted
- In order to solve the problem, need to active the community again

## Patchwork smart/rpm: nativesdk should get RPM\_ETC

Project: oe-core : [patches](#) : [project info](#) : [other projects](#)

|            |  |
|------------|--|
| Submitter  | <a href="#">Bian Naimeng</a>                             |
| Date       | Oct. 23, 2015, 2:10 a.m.                                 |
| Message ID | <1445566236-9759-1-git-send-email-biannm@cn.fujitsu.com> |
| Download   | <a href="#">mbox</a>   <a href="#">patch</a>             |
| Permalink  | <a href="#">/patch/106097/</a>                           |
| State      | New  |
| Headers    | <a href="#">show</a>                                     |

### Comments

[Bian Naimeng](#) - Oct. 23, 2015, 2:10 a.m.

The config file of rpm will be installed into SDKTARGETSYSROOT for each architecture, so RPM\_ETCRPM should be set to SDKTARGETSYSROOT/etc/rpm.

Signed-off-by: Bian Naimeng <biannm@cn.fujitsu.com>

---

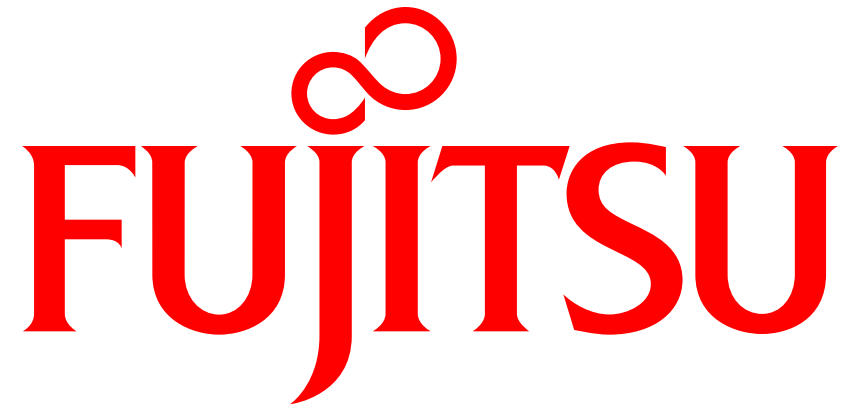
```
meta/recipes-devtools/python/python-smartpm_git.bb | 14 ++++++-----
meta/recipes-devtools/rpm/rpm_5.4+cvs.bb           | 22 ++++++-----
meta/recipes-devtools/rpm/rpm_5.4.14.bb             | 22 ++++++-----
3 files changed, 38 insertions(+), 20 deletions(-)
```

<http://patchwork.openembedded.org/patch/106097/>

- Summary the current package managers and problems
  - Compare the package managers
  - Introduce package management problem in Yocto Project
  - The package manager is necessary
- Contribution to Ycoto Project
  - Introduce the Smart Package Manager
  - Fix the bugs and make Smart Package Manager easy to use
  - License management
- Not stick to Smart Package Manager, but want to solve package management problem



- Continue to make contribution for Yocto Project
  - The interface of Smart is not so smart currently.
  - Fujitsu will improve the interface to make it easier to use.
- Remote access over a network connection like APT, YUM and DNF
- Add the option to create image automatically



shaping tomorrow with you