

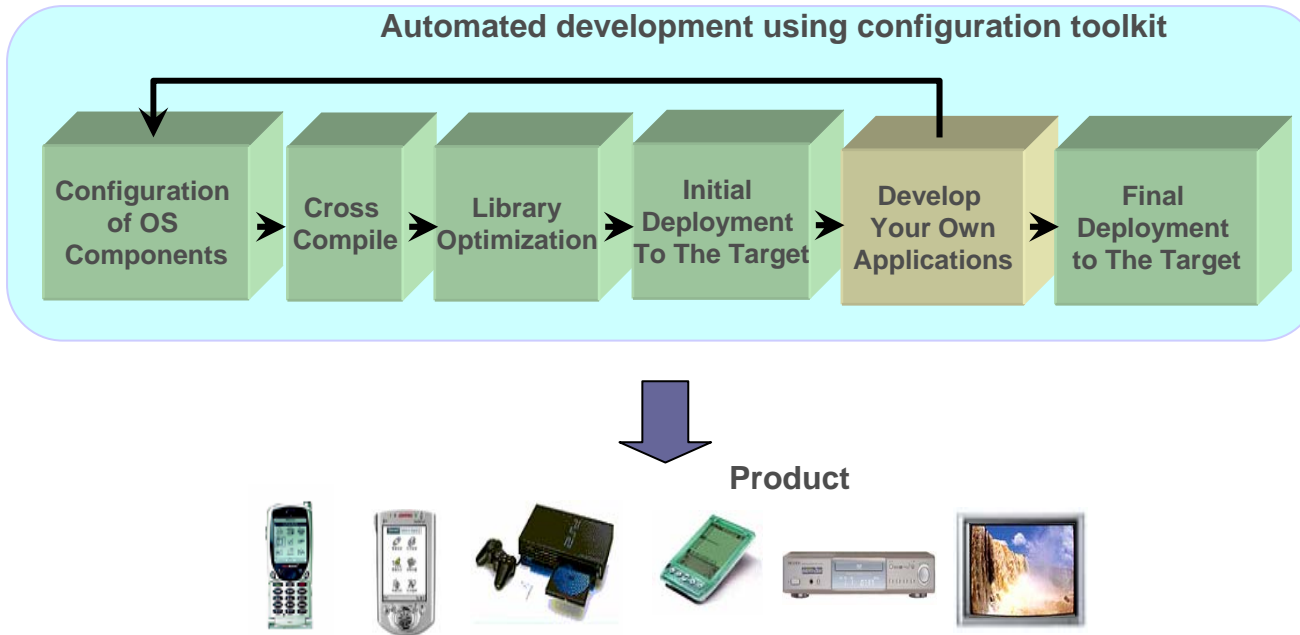
# TargetBuilder



**임베디드 OS 연구팀**  
**정평재 (system9@etri.re.kr)**

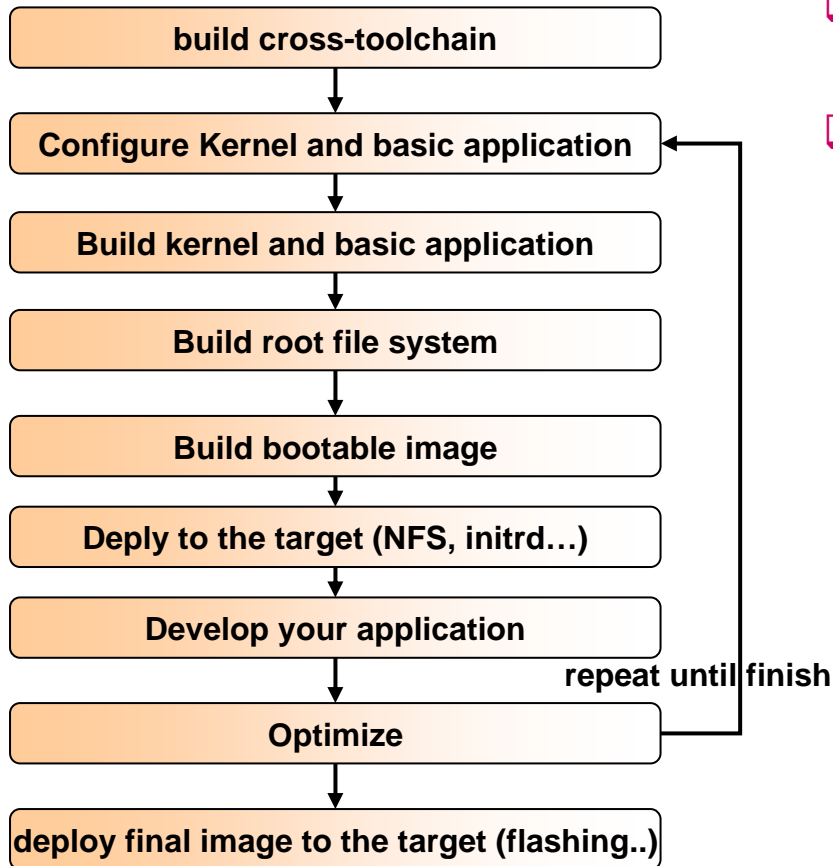
- 임베디드 리눅스 설정 툴킷 소개
- 타겟빌더 소개
- 타겟빌더 작동원리
- 패키징 시스템 작동원리
- QPD파일 설명

- 수동으로 임베디드 리눅스 시스템을 구성하는 작업은 매우 어려움
  - 시스템 어플리케이션 소스 수집, 패치, 컴파일등의 과정이 복잡함
  - 복잡한 작업들은 모두 비용 증가의 원인이 됨
  - Time-to-market 적용에 어려움
- 자동화된 임베디드 리눅스 시스템 설정 도구를 이용하여 복잡한 작업을 자동화
- 브랜드화된 임베디드 리눅스 시스템의 상품화를 위해서 잘 정리된 설정 툴킷은 반드시 필요



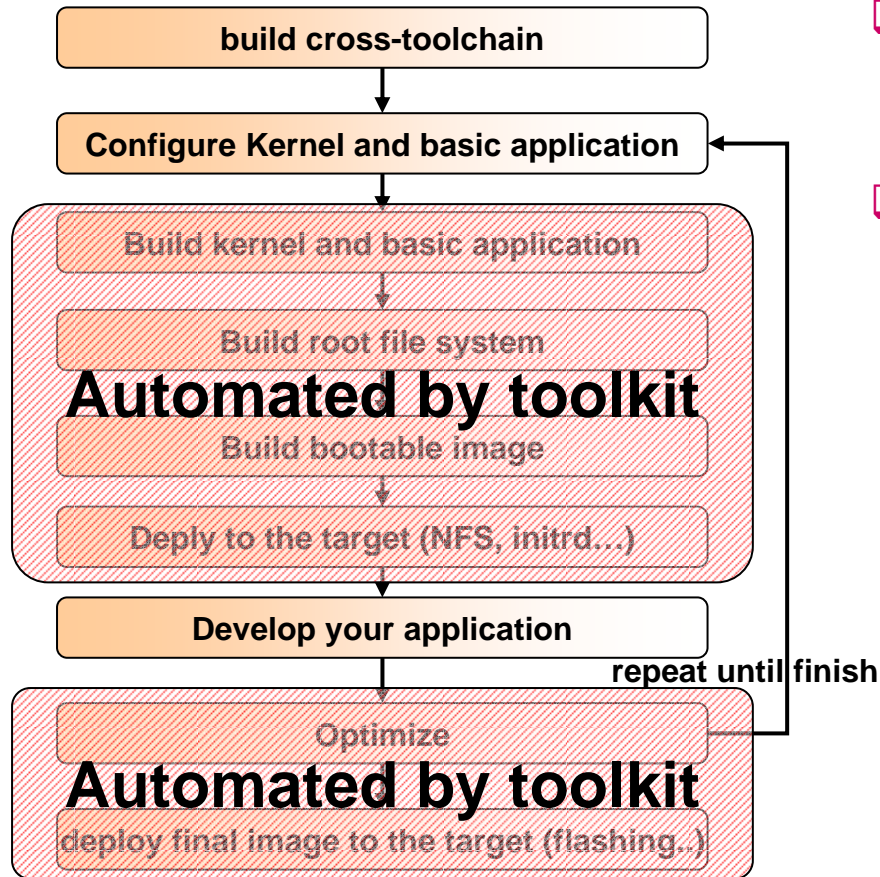
- ❑ Point & Click 설정 기능
- ❑ 다양한 아키텍처 지원
- ❑ 자동화된 의존성 검사 기능
- ❑ 다양한 적재 기능 지원
- ❑ 세밀한 시스템 설정 기능 (파일 리스트, 컴파일 옵션등...)
- ❑ 라이브러리 최적화 기능

# 설정도구의 필요성 (1/2)



- ❑ 최종 개발이 끝날때 까지 이러한 개발 프로세싱이 반복
- ❑ 설정도구 없이 수동으로 개발시 많은 오류 발생 가능성이 있으며, 개발 일정을 맞추기가 어려움

# 설정도구의 필요성 (2/2)



- ❑ 설정도구를 이용하여 개발시에 수동으로 개발시에 비해 상당부분의 작업이 줄어듬
- ❑ 비용절감효과를 기대할 수 있음

# 다양한 임베디드 시스템 설정 도구들 (1/3)



WIND RIVER



# 다양한 임베디드 시스템 설정 도구들 (2/3)



제품명	개발사
TargetBuilder	ETRI
DevRocket	montaVista
ELITE	Lineo
Target Configurator	Timesys
Tornaro II	Windriver
Platform builder	Microsoft
Buildroot, open embedded, scratchbox...	



## ❑ Openembedded

- OpenZaurus, Familiar에서 사용
- <http://oe.handhelds.org/>

## ❑ ScratchBox

- Nokia 770 Internet Tablet에서 사용
- <http://www.scratchbox.org/>

## ❑ BuildRoot

- Uclibc용 root file system 제작에 사용
- <http://buildroot.uclibc.org/>

# 타겟빌더 스크린샷 – 메인 설정(1/2)



The screenshot shows the Target Builder Main interface with the following components highlighted:

- project view:** A callout box pointing to the Navigator on the left side of the window.
- configuration tree Kernel, Application, Target system environment setup view:** A callout box pointing to the central configuration tree.
- symbol property:** A callout box pointing to the Symbol property window on the right.
- Dependency list:** A callout box pointing to the Dependency list table at the bottom of the Symbol property window.

The configuration tree shows the following structure:

- test
  - build
  - conf
  - deploy
  - menu
  - tools
  - .project
  - project.conf
- Networking support
- ISDN subsystem
- Telephony Support
- Input device support
  - Userland interfaces
  - Joystick interface
  - Touchscreen interface
- Event interface
- Event debugging
- Input I/O drivers
- Gameport support
- Serial port line discipline
- ct82c710 Aux port controller
- Parallel port keyboard adapter
- PCI PS/2 keyboard and PS/2 mouse controller
- Input Device Drivers
- Mice
- Joysticks
- Touchscreens
- Misc
- Character devices
- I2C support
- Misc devices
- Multimedia devices
- Graphics support
- Sound
  - Sound card support
  - Advanced Linux Sound Architecture
  - Open Sound System

The Symbol property window shows the following details for INPUT\_TSDEV:

- Symbol: INPUT\_TSDEV
- Prompt: Touchscreen interface
- Type: Tristate
- Value: [dropdown menu]
- Help: Say Y here if you have an application that only can understand the Compaq touchscreen protocol for absolute pointer data. This is useful namely for embedded configurations. If unsure, say N. To compile this driver as a module, choose M here: the module will be called today.

The Dependency list table shows the following data:

Symbol	Prompt	Value
INPUT	Input devices (needed for keyboard, mouse, ...) y	

# 타겟빌더 스크린샷 – 워저드 시스템 (2/2)



The image displays a sequence of screenshots from the Q+ System Project wizard and the Target Builder Main interface. The wizard steps include:

- Select a wizard:** Shows the 'Q+ System Project' wizard selected.
- Architecture Information:** Shows the selection of architecture 'i386' and board 'lepia'.
- Q+ System config Wizard:** Shows the selection of 'Q+ CE Edition Configure Mode (CE edition DEMO)'.

The **Target Builder Main - Demo : System Editor - Eclipse Platform** interface shows the following components:

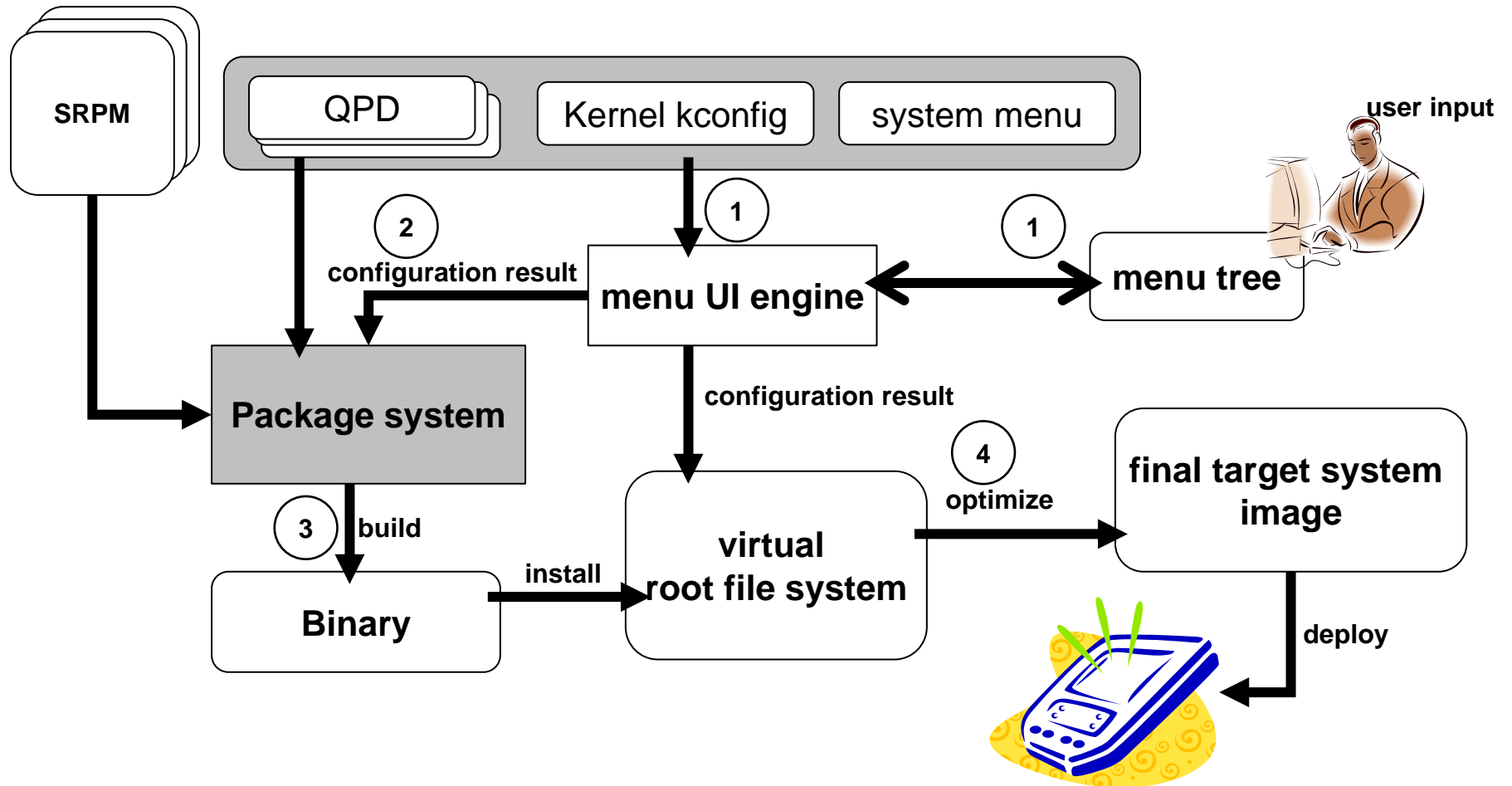
- Navigator:** Displays the project structure, including 'demo', 'build', 'conf', 'deploy', 'menu', 'project', 'project.conf', 'ethnrg', 'ok', 'Test', 'test1', and 'testss'.
- System Editor:** Shows the configuration tree for 'ACPI (Advanced Configuration and Power Interface)'. The 'Button' option is selected under 'ACPI Support'. Other options include AC Adapter, Battery, Fan, Processor, ASUS/Medion Laptop Extras, Toshiba Laptop Extras, Debug Statements, Power Management Timer Support, CPU Frequency scaling, Bus options (PCI, PDMCIA, EISA, MCA, ISA), Executable file formats, Device Drivers, File systems, Profiling support, Kernel hacking, Security options, Cryptographic options, Library routines, Package Configuration, and Target Configuration.
- Search:** Shows the search results for 'ACPI\_BUTTON', including its type 'Button' and value.
- Console:** Shows the execution of the target builder, with the following output:

```
Console
/home/system3/run/iee-workspace/demo/project.conf
/home/system3/run/iee-workspace/test1/project.conf
/home/system3/run/iee-workspace/demo/project.conf
```

- ❑ **Kconfig 기반의 메뉴시스템**
  - Linux 커널 2.6지원
  - 자동 의존성 검사 기능
- ❑ **RPM기반 패키지 시스템**
  - 소스기반 패키지 및 바이너리 RPM동시지원
- ❑ **라이브러리 최적화**
- ❑ **시스템 사이즈 계산기능**
  - 각 패키지별 사용용량 표시
- ❑ **세밀한 시스템 설정 기능**
  - 설치 파일 설정, 컴파일 옵션
- ❑ **다양한 적재방법 지원**
  - NFS, CD installer, USB boot, ETC...
- ❑ **Standard Qplus 및 Nano Qplus 지원**
  - Nano Qplus 1.6.1e
- ❑ **이클립스 플랫폼**
  - 프로젝트 위저드

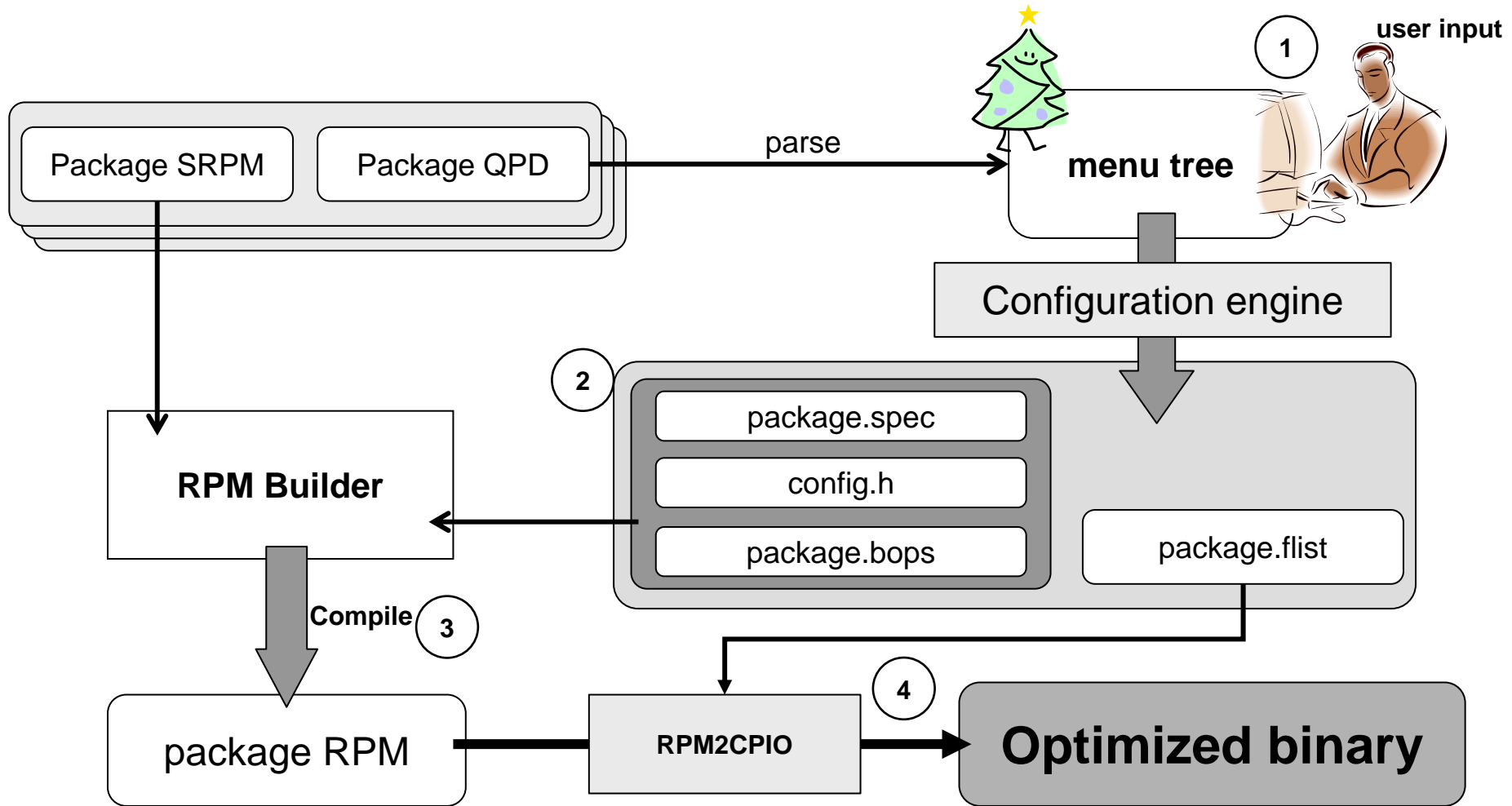
1. 사용자로부터 커널 및 어플리케이션 설정을 입력 받는다.
2. 사용자로부터 받은 설정은 비엔진을 통해서 패키지 시스템으로 전해진다.
3. 패키지 시스템은 비엔진으로부터 받은 설정대로 어플리케이션을 빌드한다.
4. 빌드가 된 어플리케이션들은 사용자 설정에 의해 Virtual rootfile system으로 구성된후 최적화 과정을 통해 타겟에 적재된다.

# 타겟빌더 작동 원리 (2/2)



1. 사용자로부터 어플리케이션 셋팅을 입력받는다.
2. 설정 엔진은 사용자로부터 받은 설정값을 토대로 RPM spec파일을 만든다.
3. RPM 빌드 엔진은 2번에서 제작된 spec파일을 이용해서 어플리케이션을 빌드한다.
4. 제작된 바이너리 RPM파일은 사용자 설정에 따라 필요한 파일만 RPM2CPIO를 이용해서 추출한다.

# 패키징 시스템 작동 원리 (2/2)





- ❑ RPM spec파일의 확장 설정 언어
- ❑ 각각의 옵션별 설치 해야할 파일 리스트 추출
- ❑ 각 옵션간 의존성 기술
- ❑ 컴파일 옵션변경 기능

# Qplus Package Descriptor – QPD (2/3)



Property	Description
%%prompt	Short description of the option
%%files	files to be installed if the option is selected
%%require	Describe dependencies
%%provide	options exporting same provide symbol here can't be selected at the same time

# Qplus Package Descriptor – QPD (3/3)



- example : package descriptor of 'tiny x' package

```
...
%build          #RPM spec's 'build' section
...
%install        #RPM spec's 'install' section
...
%files          #RPM spec's 'files' section
...
%option tinyx
%%prompt tinyx package
%%desc
the tiny x window system provides the base technology
for developing GUI with minimum footprint
%%files
/usr/X11R6/X11
/usr/X11R6/*so*
%option tinyx/xfbdev
%%prompt xfbdev server
%%require 'FB' == y
%%files
/usr/X11R6/xfbdev
...
```

Option name

Install file list when option is selected

dependency

