

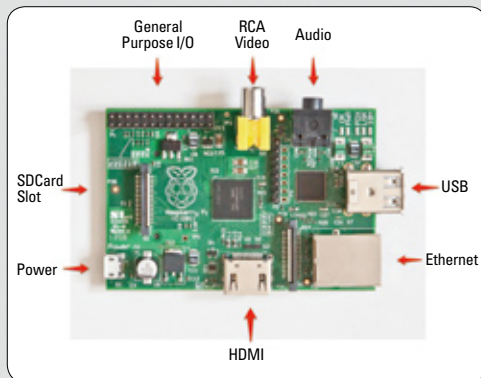
## 전세계 개발자를 열광시킨 Raspberry Pi와 Java Technology의 활용

초소형 컴퓨터 라즈베리파이(Raspberry Pi)의 열풍이 뜨겁게 불고 있다. 2012년 초 처음 발매 이후 벌써 100만대 이상이 판매되었으며, 이미 해외나 국내의 발빠른 개발자들은 이 작은 컴퓨터를 이용하여 다양한 창작품들을 만들어내고 있다. 그리고 얼마 전 구글(Google)이 영국의 어린이 프로그래밍 교육을 위해 라즈베리파이 15,000대를 기부했다는 소식도 언론에 소개되고 있는 것을 보니, 그 열풍을 쉽게 가늠해 볼 수 있을 것 같다. 도대체 신용카드 한 장 보다 조금 큰 크기에 불과한 이 작은 컴퓨터가 무엇이기에 이렇게 화제를 만들어내는걸까? 본고에서는 최근 개발자들 사이에서 화제가 되고 있는 라즈베리파이에 대하여 자세히 알아보고, 최신의 Java기술을 이용하여 이를 좀 더 다양하게 활용할 수 있는 방법에 대하여 알아보려고 한다.

저자 - 최승호 컨설턴트, 한국오라클 Java Embedded GSU(seungho.choi@oracle.com)

### 라즈베리파이(Raspberry Pi)란?

라즈베리파이는 영국의 라즈베리파이 재단이 학교에서 기초 컴퓨터 과학 교육을 증진시키기 위해서 만든, 신용카드 한 장 크기 정도의 기판으로 제작된 싱글보드 컴퓨터이다. 기본적으로 작은 회로보드, ARM 기반의 CPU, 그래픽 프로세서, 이더넷, 그리고 몇가지 핀과 포트로 구성된 단순한 형태의 PC이지만, 데스크탑PC가 할 수 있는 많은 것들을 라즈베리파이에서도 할 수 있다. 운영체제(OS)는 일반적으로 리눅스를 탑재하는 경우가 많지만, 오픈소스 PC인만큼 안드로이드를 탑재하여 사용하기도 한다.



<그림 1> 라즈베리파이 모델B

<그림 1>은 라즈베리파이 모델B를 나타내고 있으며, 그림에서 볼 수 있듯이 화면 출력을 위한 HDMI포트를 비롯하여 다양한 입출력 포트 등이 내장되어 있는 것을 확인할 수 있다.

라즈베리파이의 그래픽 성능이 뛰어나면서도 저렴한 가격(모델 A의 경우 25달러, 모델 B의 경우 35달러)이라는 특징을 갖고 있다. 세부적으로 좀 더 자세히 특징을 살펴보면 브로드컴의 BCM2835 단일 칩 시스템을 사용하며, 이 칩에는 ARM1176JZF-S 700 MHz 프로세서, 비디오코어 IV GPU와 RAM이 들어 있다. 라즈베리파이는 하드 디스크나 솔리드 스테이트 드라이브를 비롯한 내장 디스크 드라이브를 포함하고 있지 않으며, SD카드를 운영체제를 포함한 데이터를 담기 위한 외부저장 장치로 사용하고 있다. 와이파이와 블루투스 모듈이 빠져 있는 것이 아쉬운 부분이지만, 별도로 판매하는 모듈을 사용하면 이러한 기능들도 추가해서 사용할 수 있다. 라즈베리파이 재단은 컴퓨터 교육 증진을 위해 2가지 모델을 시중에 출시하였으며, 각각 25달러(256MB RAM)와 35달러(512MB RAM)의 저렴한 가격으로 책정하였다. 라즈베리파이의 자세한 하드웨어 스펙은 아래와 같다.

목표가격	US \$25	US \$35
SoC	브로드컴 BCM2835 (CPU, GPU, DSP, 그리고 SDR SDRAM)	
CPU	700 MHz ARM1176JZF-S core (ARM11 계열)	
GPU	브로드컴 비디오코어 IV, OpenGL ES 2.0, 1080p30 h.264/MPEG-4 AVC 높은 프로파일 디코더	
메모 (SDRAM)	256MB (GPU와 같이 사용)	512MB (GPU와 같이 사용)
USB 2.0 Ports	1	2 (내장 USB 허브 하나 제외)
비디오 출력	컴포지트 RCA (PAL & NTSC), HDMI (rev 1.3 & 1.4), 액정 디스플레이를 위한 디스플레이 시리얼 인터페이스 640×350 부터 1920×1200 이상의 14개의 HDMI 해상도와 여러 PAL, NTSC 표준들	
오디오 출력	3.5mm 잭, HDMI	
내장 저장장치	SD/MMC/SDIO 카드 슬롯	
내장 네트워크	None	10/100 이더넷(RJ45)
로우레벨 주변장치	8 × GPIO, UART, I <sup>2</sup> C 버스, 2개의 칩을 선택할 수 있는 SPI, +3.3 V, +5 V, 접지	
전력	500mA (2.5W)	700mA (3.5W)
전원	MicroUSB에서 받는 5볼트 혹은 GPIO 헤더	
크기	85.60×53.98mm	
무게	45g	
지원중인 운영체제	Debian GNU/Linux, Fedora, Arch Linux ARM, RISC OS	

<표 1> 라즈베리파이의 하드웨어 스펙

라즈베리파이 재단 측에서는 라즈베리파이에 포팅한 데비안(Debian)과 아치(Arch)리눅스, QtonPi 등의 리눅스 배포판을 제공하고 있으며, 라즈베리파이의 주 프로그래밍 언어인 스크래치(Scratch), 파이썬에 BBC 베이직(Basic), C, 펄(Perl)등을 지원할 수 있는 툴들을 계획 중에 있다.

라즈베리파이는 비상업적 환경에서 다양한 분야에 응용될 수 있다. 라즈베리파이를 이용하여 약간의 프로그래밍 과정을 거쳐 게임을 조작할 수 있는 게임패드를 만들기도 하고, XBMC를 설치해 미디어 센터로 활용하거나, 정해진 시간마다 화분에 물을 뿌려주는 간이 스프링클러 등을 제작하기도 한다. 이 외에도 센서와 LCD 패널을 연결하여 온도/습도계를 만들어내는 등, 사용자가 상상하는 것 그대로 라즈베리파이를 이용하여 손쉽게 다양한 형태의 창작물들을 만들어 낼 수 있다.





<그림 2> 라즈베리파이를 활용한 다양한 창작물들

<그림2>는 라즈베리파이를 이용하여 만들어진 다양한 형태의 창작물들을 나타내고 있으며, 그림에서 볼 수 있듯이 라즈베리파이를 이용하여 게임 콘솔부터 미니노트북, 심지어 슈퍼컴퓨터에 이르기까지 다양한 형태의 결과물들이 만들어지고 있다. 지금 이 시간에도 다양한 형태의 창작물들이 만들어지고 있을 뿐만 아니라, 여러 커뮤니티들을 통하여 국내 개발자들의 활동도 꽤나 활발하게 이루어지고 있다.

그럼 이제, 라즈베리파이를 활용할 수 있도록 운영체제를 설치하는 방법을 알아보자.

### 라즈베리파이 환경 준비

서두에서 언급하였다시피 라즈베리파이는 기본적으로 하드웨어만을 제공하기 때문에, 이를 활용하기 위해서는 공식사이트에서 제공하는 운영체제를 설치해 주어야 한다. 오픈소스 PC인 만큼 안드로이드 OS를 비롯하여 다양한 오픈소스 기반의 운영체제를 설치할 수 있지만, 공식사이트에서는 리눅스 커널기반의 운영체제만을 제공한다.

라즈베리파이에 설치할 수 있는 OS는 공식사이트(<http://www.raspberrypi.org/downloads>)에서 확인 및 다운로드 할 수 있으며 이 글을 쓰고 있는 현재, 공식사이트에서는 다음의 네가지 버전의 리눅스 배포판을 제공한다.

- Raspbian "wheezy"
- Soft-float Debian "wheezy"
- Arch Linux ARM
- RISC OS

각각의 배포판에 대한 자세한 설명은 라즈베리파이의 공식 사이트에서 참조할 수 있으며, 이 글에서는 Raspbian "wheezy" 리눅스를 기준으로 설명한다.

**Note :** Raspbian "wheezy"는 데비안(Debian) 리눅스를 라즈베리파이에 맞게 최적화한 버전이며, LXDE, Midori, 개발자 도구 및 멀티미디어 기능 그리고 예제 소스코드들을 포함하고 있다.



<그림 3> Raspbian "wheezy" 다운로드 페이지

라즈베리파이에 운영체제를 설치하는 방법은 다음과 같다.

① 설치하고자 하는 리눅스의 이미지를 다운로드 받는다. <그림3>

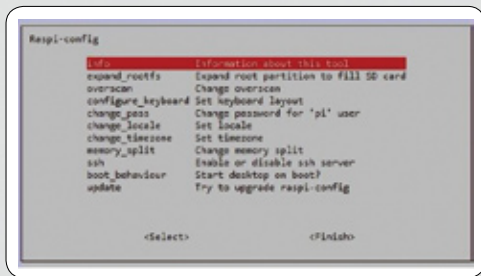
② 다운로드 받은 이미지 파일을 dd(Unix Tool)나 Win32DiskImager(Windows Tool)를 이용하여 SD 카드에 운영체제를 설치한다. 예를들어 dd를 이용한다면, 아래의 명령어로 SD카드에 운영체제를 설치할 수 있다.

```
$ddbs=1Mif=[라즈베리파이 OS Image Path] of=[SDCard path]
```

③ 운영체제의 설치가 완료된 SD카드를 라즈베리파이의 SD카드 슬롯에 넣어준 후, MicroUSB 형태의 전원 케이블을 입력한다.

**Note :** 라즈베리파이에는 별도의 On/Off 스위치가 없다.

④ 전원을 입력하면, 화면에 부팅과 관련된 메시지들이 출력되며, 부팅 과정이 완료된 후 아이디/패스워드를 물어보게 된다. 이때 기본값은 pi/raspberry이다.



<그림 4> raspi-config 환경설정 페이지

⑤ 최초 부팅단계에서 로그인이 완료되면, 자동으로 라즈베리파이의 설정 페이지(rasp-config)가 실행된다.<그림4> 설정페이지에서 다음의 항목을 추가로 설정한다. 참고로 설정 페이지는 이후 sudo raspi-config를 이용하여 다시 실행할 수 있다.

a. expand\_rootfs: root 파티션을 확장한다. (실제 Resize 작업은 이후의 재시작 단계에서 수행된다.)

**Note :** expand\_rootfs는 필수적으로 실행해야하는 작업이다. 왜냐하면, SD카드에 올렸던 이미지는 2기가바이트로 크기가 고정되어 있어서, 메모리카드의 용량에 상관없이 2기가바이트의 크기로 파티션이 설정되기 때문이다. expand\_rootfs를 실행하면 이후의 재시작 단계에서 파티션의 크기를 최대로 확장시켜 준다.



- b. `change_timezone`: Timezone을 설정한다.
- c. `memory_split`: CPU와 GPU에서 사용하게 될 메모리의 양을 설정한다. 3D 환경이나 복잡한 그래픽 연산, 비디오 디코딩의 작업이 필요할 경우 GPU에 128MB를 할당한다. 다음단계에서 다루게 될 JavaFX 애플리케이션을 원활하게 실행하기 위해서는 128MB의 메모리를 할당한다.
- d. `ssh`: SSH서버의 실행여부를 설정한다.
- e. `boot_behavior`: X-Windows Desktop의 자동실행 여부를 설정한다.
- f. Finish를 선택하여 라즈베리파이의 설정을 완료한다.

⑥ 에디터를 실행하여 `/boot/config.txt` 파일을 수정한다. 예를들어 라즈베리파이의 콘솔이나 SSH에서 vi 에디터를 사용한다면, 다음의 커맨드를 실행하여 파일을 수정할 수 있다.

```
$ sudo vi /boot/config.txt
```

- a. `/boot/config.txt` 파일의 여러 항목 중, 아래의 항목을 찾아 주석(Comment)을 제거한다.

```
framebuffer_width=1280
```

```
framebuffer_height=720
```

이 단계까지 완료하면, 이제 라즈베리파이를 사용하기 위한 준비는 끝난 상태이다. 아래의 커맨드를 입력하여 라즈베리파이를 재시작하여, 변경한 설정값을 적용한다.

```
$ sudo shutdown -r
```

### 자바 개발킷(Java Development Kit) 설치

자바(Java)가 다른 프로그래밍 언어와 비교하여 가지는 가장 중요한 특징 중 하나는 자바는 플랫폼에 독립적인 방향으로 설계되었다는 것이다. 자바로 작성된 애플리케이션은 자바를 지원하는 모든 운영체제에서 별도의 코드수정 없이 실행될 수 있으며, 그 어떤 언어로 작성된 프로그램보다 훨씬 이식성이 뛰어나다는 장점을 가진다. 자바로 작성된 애플리케이션은 실행할 플랫폼의 개수에 상관없이 한가지의 소스 코드만 있으면 된다. 다른 프로그래밍 언어의 경우에는 애플리케이션을 각각의 플랫폼에서 실행하기 위해서 여러가지의 소스코드가 필요하다. 특히 다양한 GUI가 연관되어 있는 경우나 다양한 하드웨어 스펙이 존재하는 임베디드 시스템의 경우 이러한 이식성은 커다란 강점이자 장점이 될 수 있다.

자바의 중요한 두 번째 특징은 자바가 객체지향적(Object Oriented)이라는 것이다. 객체지향 프로그램은 이해하기 쉬우며, 유지보수와 확장이 필요할 때 소모되는 시간을 줄여준다.

라즈베리파이 재단에서 배포하는 리눅스 운영체제에는 기본적으로 Java가 설치되어 있지 않기 때문에 Java 애플리케이션을 개발하기 위해서는 우선 라즈베리파이에 자바 개발킷(Java Development Kit)을 설치하여야 한다.

얼마 전 오라클은 Java SE8 Developer Preview for ARM 버전을 릴리즈하였다. Java SE8 for ARM 버전은 임베디드(Embedded) 시스템에서 동작하는 애플리케이션을 개발하기 위한 최적의 플랫폼을 제공한다. 이 릴리즈는 또한 라즈베리파이와 같이 ARM 프로세서를 사용하는 리눅스 시스템에서 동작하는 GUI 애플리케이션의 개발을 지원하기 위한 JavaFX for Linux를 포함하고 있다. 또한 OpenGL 기반의 그래픽 가속 기능을 제공하기 때문에 복잡한 3D나 2D 그래픽을 포함하는 애플리케이션의 제작에도 적합한 개발 환경을 제공하고 있다.

**Note :** JavaFX는 자바 언어를 기반으로 하는 RIA(Rich Internet Application) 개발툴킷으로써 다양한 UI 컴포넌트들과 미디어 처리를 지원한다. 기존의 Java Swing과 비교하였을 때, 보다 간편한 방법으로 UI를 구성할 수 있으며, 고수준의 그래픽 작업을 수행할 수 있다. Java 8에서는 JavaFX 8이 자바의 기본 UI로 제공될 예정이다.



<그림 5>JDK & Sample 다운로드 페이지

라즈베리파이에 Java SE8 for ARM을 설치하는 방법은 다음과 같다.

① <http://jdk8.java.net/fxarmpreview/>에서 JDK8(with JavaFX) for ARM 버전과 샘플을 다운로드 한다.<그림 5>

② 다운로드 받은 JDK 파일을 아래의 명령어를 순서대로 실행하여 설치한다.

```
$ mkdir -p /opt
$ sudo tar zxvf fx_ea_download_file_name-C/opt
```

③ 설치가 완료되면, 아래의 명령어를 실행하여 다운로드 받은 설치 파일을 제거한다.

```
$ rm fx_ea_download_file_name
```

④ 정상적으로 Java가 설치되었는지 아래의 명령어를 실행하여 확인한다.

```
$ sudo /opt/jdk1.8.0/bin/java -version
java version "1.8.0-ea"
Java(TM)SERuntimeEnvironment(build1.8.0-ea-b36e)
JavaHotSpot(TM)ClientVM(build25.0-b04,mixedmode)
```

정상적으로 설치가 되었을 경우 아래와 같은 메시지가 화면에 출력된다.



<그림 6> Java SE8 for ARM에 포함되어 있는 JavaFX 기반의 애플리케이션

자바의 설치를 완료하였다면, 다운로드 받은 샘플 애플리케이션들을 실행해 볼 수 있다.

<그림6>은 Java SE8 for ARM에 포함되어 있는 JavaFX 예제 애플리케이션을 라즈베리파이에서 실행해 본 것이며, PC 데스크탑 환경에서 볼 수 있는 결과와 동일한 형태의 GUI가 제공되는 것을 확인할 수 있다. 예제 애플리케이션은 다음과 같은 형

식의 명령어를 입력하여 실행할 수 있다.

```
$ sudo /opt/jdk1.8.0/bin/java --cp \
/opt/jdk1.8.0/jre/lib/jfxrt.jar:JavaFX_Sample_path/sample_name.jar \
sample_app_entry_class
```

예를 들어, Stopwatch 애플리케이션을 실행하기 위해서는 아래와 같이 명령어를 입력하여 실행한다.

```
$ sudo /opt/jdk1.8.0/bin/java -Djavafx.platform=eglfb -cp \
/opt/jdk1.8.0/jre/lib/jfxrt.jar:/tmp/JavaFXSamples/Stopwatch.jar \
stopwatch.MainScreen
```



<그림 7> 라즈베리파이에서 아파치 톰캣 서버 구동

이 뿐만 아니라, 자바의 가장 큰 특징 중 하나라고 할 수 있는 WORA(Write Once, Run Anywhere : 한번 개발한 코드는 다양한 플랫폼 어디서나 실행할 수 있다)의 개념을 바탕으로, 이미 보유하고 있는 다양한 애플리케이션 및 코드들을 별다른 포팅작업 없이 바로 테스트해볼 수 있다. 예를 들어, 아파치 톰캣(Tomcat)이나 제티(Jetty)와 같은 오픈소스 기반의 자바 서버

소프트웨어를 설치하여 라즈베리파이를 작은 크기의 애플리케이션 서버(Small Embedded Server)로 변신시키는 것도 가능하다.<그림7>



<그림 8> 라즈베리파이와 자바를 이용한 다양한 디바이스 제어

또한, 라즈베리파이에 <그림 8>과 같이 다양한 USB 디바이스(Device)들을 연결한 후, 자바 API를 이용하여 제어하는 등의 실험적인 시도들도 개발자들 사이에서 꾸준히 일어나고 있다.

지금까지 라즈베리파이와 자바 개발 환경을 구성하는 방법에 대하여 알아보았다.

구슬이 서말이라도 꿰어야 보배라 했듯이, 이제 준비된 환경을 이용하여 애플리케이션을 개발하는 방법을 알아보자.

### 자바 애플리케이션 개발

라즈베리파이에서의 Java 애플리케이션 개발 과정은 데스크탑 환경에서의 그것과 완전히 동일하다.

넷빈즈(NetBeans)를 비롯하여 이클립스(Eclipse), 제이디벨로퍼(JDeveloper), 인텔리제이(IntelliJ) 등과 같은 통합개발환경(IDE)을 이용하여 애플리케이션을 개발할 수 있다.

지금부터 대표적인 자바 통합 개발환경인 넷빈즈(NetBeans)를 이용하여 자바 애플리케이션을 개발하는

방법을 설명한다. 넷빈즈는 임베디드(Embedded)부터 데스크탑(Desktop), 그리고 엔터프라이즈(Enterprise) 애플리케이션까지 자바를 기반으로 하는 모든 애플리케이션의 개발에 필요한 과정(테스트 및 디버깅을 포함하여)을 하나의 틀에서 수행할 수 있는 통합된 개발 환경을 제공해 준다.

넷빈즈를 이용하여 애플리케이션을 개발하는 과정은 대부분의 경우 다음의 과정을 따른다.

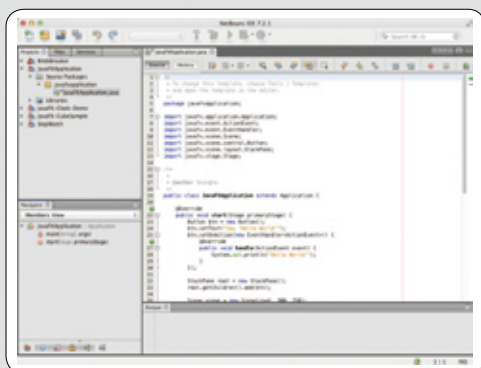
먼저 새로운 프로젝트를 생성한다. 새로운 프로젝트 생성의 첫 번째 단계는 프로젝트의 속성을 설정하는 부분으로, 개발하고자 하는 애플리케이션의 종류(Categories)를 먼저 선택한 후, 프로젝트에 대한 속성(Projects)을 결정한다. 각각의 종류는 그에 맞는 프로젝트 속성을 포함하고 있다.

<그림 9>

다음으로 선택한 프로젝트의 종류에 따라



<그림 9> 애플리케이션 및 프로젝트 속성 설정



<그림 10> 프로젝트 윈도우 및 코드 윈도우





<그림11>JavaFX UI를 쉽게 개발할 수 있는 Scene Builder

적절한 속성값을 입력한 후, 새로운 프로젝트를 생성한다.

넷빈즈는 사용자의 입력값에 기반하여 새로운 프로젝트를 생성하며, 이렇게 생성된 프로젝트에는 바로 실행하여 결과를 확인할 수 있는 템플릿(Template) 코드를 포함하고 있다. 이러한 코드 템플릿이나 자동 완성 등의 기능을 이용하면 보다 쉽게 애플리케이션을 개발할 수 있다.<그림10>

그리고 JavaFX와 함께 제공되는 씬 빌더(Scene Builder)는 애플리케이션의 UI를 시각적인 환경에서 개발할 수 있는 개발 환경을 제공할 뿐만 아니라 JavaFX의 주요 기능 지원 및 넷빈즈를 비롯한 기타 다른 자바 개발 환경과 쉽게 상호 작용할 수 있다.<그림 11>

이 외에도 넷빈즈는 자바 개발환경 뿐만 아니라 HTML5 애플리케이션, 자바스크립트(Javascript), 루비(Ruby), 그루비(Groovy), 파이썬(Python), C/C++ 등과 같은 다양한 언어의 개발환경을 지원하고 있다.

넷빈즈에 대한 자세한 내용은 넷빈즈의 공식 사이트에서 확인할 수 있다.

### 라즈베리파이, 그 무한한 가능성...

라즈베리파이 재단은 지난 2012년 12월, 라즈베리파이용 응용프로그램(앱)을 내려받을 수 있는 앱 장터인 파이스토어(Pi Store)를 오픈하였다. 파이스토어는 컴퓨터 디자인, 애플리케이션, 팁을 공유하는 허브 역할을 한다. 이를 통하여, 전세계 사람들의 다양한 아이디어를 공유할 수 있으며, 원한다면 내가 만든 아이디어를 전세계 사람들과 공유할 수 있다.

이 외에도 라즈베리파이는 좀 더 다양한 분야에서 활용할 수 있는 무한한 잠재력을 지니고 있으며, 이러한 가능성에 대한 다양한 연구가 방대한 온라인 커뮤니티 등을 통해서 이루어지고 있다. 본고를 통해서 라즈베리파이에 흥미를 느끼기 시작했다면 주저하지 말고 지금 당장 라즈베리파이의 매력 속으로 빠져들어보자.

라즈베리파이 공식 사이트 : <http://www.raspberrypi.org>  
 라즈베리파이 라눅스 다운로드 사이트 : <http://www.raspberrypi.org/downloads>  
 라즈베리파이 위키피디아 페이지 : <http://en.wikipedia.org/wiki/Raspberrypi>  
 JavaSE Embedded on the Raspberry Pi : <http://www.oracle.com/technetwork/articles/java/raspberrypi-1704896.html>  
 JavaFX Overview : <http://www.oracle.com/technetwork/java/javafx/overview/index.html>  
 JDK 8(with JavaFX) for ARM Early Access : <http://jdk8.java.net/fxarmpreview/>  
 JDK 8(with JavaFX) for ARM Early Access Document : <http://jdk8.java.net/fxarmpreview/javafx-arm-developer-preview.html>  
 NetBeans IDE : <http://netbeans.org/downloads/>  
 Apache Tomcat : <http://tomcat.apache.org>  
 Jetty : <http://jetty.codehaus.org/jetty/>  
 Win32DiskImager 다운로드 페이지 : <https://launchpad.net/win32-image-writer/+download>