

# 임베디드 SW 경진대회: 기술교육

webOS OSE SDK



# 목차

- 1. webOS OSE - SDK
- 2. CLI (Command Line Interface)
- 3. Emulator
- 4. Beanviser
- 5. Other Tools

# webOS OSE Software Development Kit



# SDK 란?

---

## SDK 정의

- 앱 개발을 위해 패키지, 테스트, 디버깅을 위한 개발 툴과 API 라이브러리를 포함하는 소프트웨어 개발 키트

## SDK 구성요소

- 개발문서, API, CLI, 에뮬레이터, 모니터링 툴 등

# SDK 란?

## Facebook 공식 SDK

### Android용 Facebook SDK

설치와 앱 실행을 추적하고, 콘텐츠 공유 기능을 제공하고, Facebook을 통한 로그인을 지원합니다.

### iOS용 Facebook SDK

iOS용 Facebook SDK를 사용하면 가장 간편하게 앱 광고의 성과를 추적하고 강력한 공유 기능을 구현할 수 있으며, 사용자에게 Facebook을 통한 앱 로그인 기능을 제공할 수 있습니다.

### JavaScript용 Facebook SDK

소셜 플러그인을 추가하고, API를 호출하고, Facebook 로그인을 구현할 수 있는 다양한 클라이언트 측 기능을 제공합니다.

### PC 게임용 Facebook SDK

친구 찾기, 라이브 방송 등을 통해 게이머를 최고의 게임 커뮤니티와 연결할 수 있습니다.

### PHP용 Facebook SDK

그래프 API에 액세스하고 Facebook 로그인을 활용하기 위한 최신 네이티브 라이브러리입니다.

### Facebook 비즈니스 SDK

광고 API를 사용하는 경우 이 PHP용 Facebook SDK를 사용하면 Facebook 광고 캠페인, 맞춤 타겟, 보고서를 생성할 수 있습니다.

### React용 Facebook SDK

JavaScript를 사용하여 모바일 플랫폼에서 Facebook에 참여를 유도하는 소셜 앱을 만드는 데 도움이 됩니다.

### Swift용 Facebook SDK

Swift용 Facebook SDK를 사용하면 iOS용 Facebook SDK의 모든 API에 대해 Swift에 맞게 조정된 환경을 얻을 수 있습니다.

### tvOS용 Facebook SDK

tvOS용 Facebook SDK를 사용하면 Apple TV에서 뛰어난 소셜 환경을 구축할 수 있습니다.

### Unity용 Facebook SDK

크로스 플랫폼 Unity 게임에 소셜 기능을 쉽고 빠르게 통합할 수 있습니다.

## Google Mobile Ads SDK

The Google Mobile Ads SDK is offered for Android through [Google Play services](#). Your use of the SDK is governed by the Google Developers Site [Terms of Service](#).

Version	17.2.0
---------	--------



## Xcode

Xcode 개발자 도구에는 macOS, iOS, watchOS 및 tvOS용 앱을 만드는 데 필요한 모든 것이 담겨 있습니다.

# webOS OSE SDK - CLI



# CLI 사용하기

---

## CLI 란?

- Command Line Interface
- 웹 애플리케이션을 생성하고, 패키징하고, 인스톨하고, 실행하기 위한 명령어를 제공

## 주요기능

- 프로젝트 생성 (필수파일, 메타파일, 웹 애플리케이션, JavaScript 서비스)
- 웹 애플리케이션 패키징
- 타겟 디바이스 관리
- 웹 애플리케이션 설치/삭제/설치 목록 보기
- 웹 애플리케이션 실행/종료/실행 중인 앱 목록 보기
- 웹 애플리케이션 디버깅

## 다운로드

- 개발자 사이트 > Develop > SDK Tools > Command Line Interface > Download and Install (<http://webosose.org/develop/sdk-tools/cli/download-and-install/>)
- 각 Host OS 별 압축 파일을 다운로드

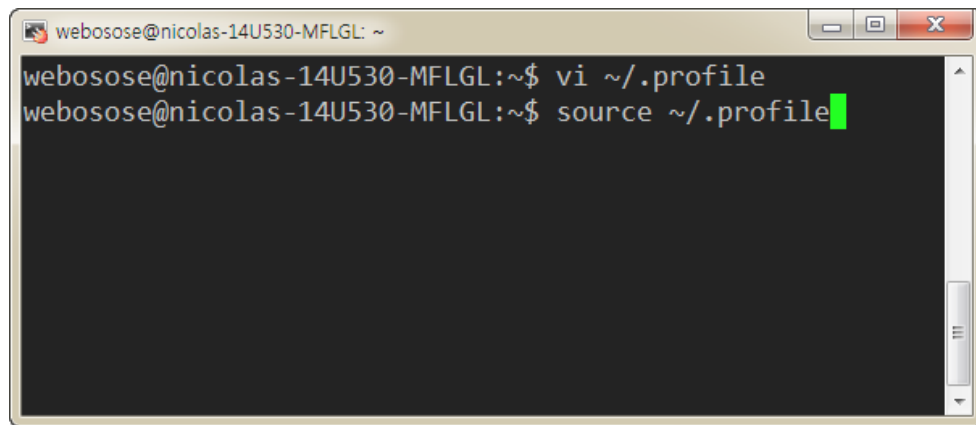
## 설치 (Ubuntu 기준)

- 압축 파일 해제,
- 명령 : `$tar -zxvf ~/Downloads/ares-webos-cli-ose_sdk-linux.tgz`
- ares-cli/bin 디렉토리에서 CLI 명령어 실행 가능



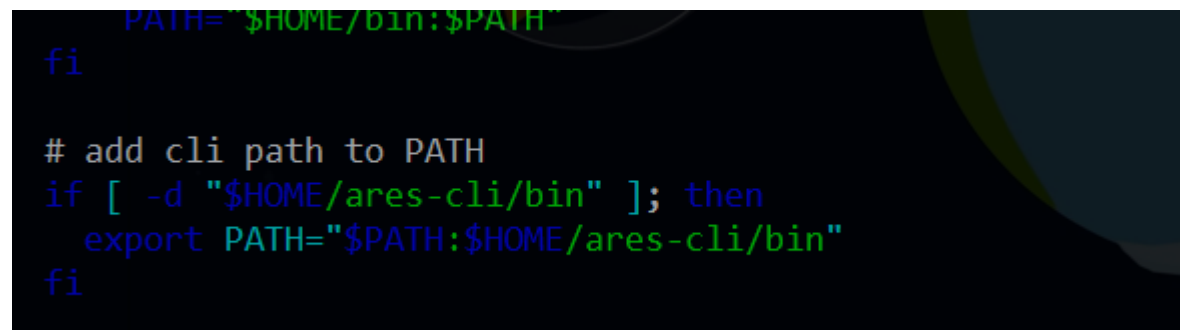
## 🐧 환경 설정 (Ubuntu 기준)

- 어느 경로에서나 CLI가 실행 가능하도록 환경을 설정



```
webosose@nicolas-14U530-MFLGL: ~  
webosose@nicolas-14U530-MFLGL:~$ vi ~/.profile  
webosose@nicolas-14U530-MFLGL:~$ source ~/.profile
```

< .profile 파일 >



```
PATH="$HOME/bin:$PATH"  
fi  
  
# add cli path to PATH  
if [ -d "$HOME/ares-cli/bin" ]; then  
    export PATH="$PATH:$HOME/ares-cli/bin"  
fi
```

- 확인
  - Home 경로에서 ares 명령 실행

## 전체 명령어

Commands	Descriptions
ares	Provides the help menu for using the ares commands.
ares-generate	Creates a web app from a template.
ares-package	Creates an app package file.
ares-setup-device	Manages the target devices.
ares-install	Installs an app on the target device.
ares-launch	Runs or terminates the web app.
ares-inspect	Runs the Web Inspector or Node Inspector for debugging web app or JS service.
ares-server	Runs the Web server for testing local app file.
ares-shell	Executes shell commands in the target device.
ares-push	Pushes file(s) from a host machine to a target device.
ares-pull	Pulls file(s) from a target device to a host machine.

### ares-setup-device

- 등록된 디바이스의 목록을 보거나 추가/삭제/수정 등의 관리를 할 수 있음

```
ares-setup-device
ares-setup-device [OPTION...] --add|-a [TARGET_NAME] [--info|-i [DEVICE_INFO]]
ares-setup-device [OPTION...] --modify|-m [TARGET_NAME] [--info|-i [DEVICE_INFO]]
ares-setup-device [OPTION...] --remove|-r [TARGET_NAME]
ares-setup-device --search|-s
ares-setup-device [--search|-s] --timeout|-t [TIMEOUT]
ares-setup-device --reset|-R
ares-setup-device --list|-l
ares-setup-device --listfull|-F
ares-setup-device --version|-V
ares-setup-device --help|-h
```

### 👉 | 옵션

- 기존에 등록된 디바이스의 목록을 볼 수 있음

```
nicolas@nicolas-14U530-MFLGL:~$ ares-setup-device -l
name      deviceinfo      connection
-----
device    root@10.177.232.180:22  ssh
target    root@10.177.233.75:22  ssh
rpi        root@10.177.232.119:22  ssh
```

```
nicolas@nicolas-14U530-MFLGL:~$ ares-setup-device -F
[
  {
    "profile": "rpi",
    "name": "device",
    "deviceinfo": {
      "ip": "10.177.232.180",
      "port": "22",
      "user": "root"
    },
    "connection": [
      "ssh"
    ],
    "details": {
      "platform": "webos",
      "password": "",
      "description": "webOS Open Device"
    }
  },
  {
    "profile": "rpi",
    "name": "target",
    "deviceinfo": {
```

### 디바이스 추가

- 신규 디바이스를 추가하거나, 기존 디바이스를 수정 혹은 삭제

```
nicolas@nicolas-14U530-MFLGL:~$ ares-setup-device
name      deviceinfo      connection
-----
device    root@10.177.232.180:22  ssh
target    root@10.177.233.75:22  ssh
rpi        root@10.177.232.119:22  ssh

** You can modify the device info in the above list, or add new device.
? Select (Use arrow keys)
[ add
  modify
  remove
```

### ares-generate

- 사용 가능한 템플릿의 목록 확인
- 템플릿 기반으로 프로젝트를 생성

```
ares-generate [OPTION...] APP_DIR  
ares-generate [OPTION...] -t js_service SERVICE_DIR  
ares-generate --list|-l  
ares-generate --version  
ares-generate --help|-h
```

### | 옵션

- 사용 가능한 템플릿 목록 보기

```
nicolas@nicolas-14U530-MFLGL:~$ ares-generate -l
basic          Web App          - (default) basic web app for webOS
webappinfo     Web App Info     - appinfo.json for web app
js_service     JS Service       - js service for webOS
jsserviceinfo  JS Service Info  - services.json, package.json for JS service
webicon        Icon             - web app icon files [80x80, 130x130]
```

### 🐼 기본 웹 애플리케이션 템플릿 생성

- -t 옵션으로 basic 템플릿 선택
- 디렉토리명은 MyFirstApp
- App ID 는 com.sample.app
- Title은 my first app
- version은 1.0.0

```
nicolas@nicolas-14U530-MFLGL:~$ mkdir kbwork
nicolas@nicolas-14U530-MFLGL:~$ cd kbwork
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-generate -t basic MyFirstApp
? app id com.sample.app
? title my first app
? version 1.0.0
Generating basic in /home/nicolas/kbwork/MyFirstApp
Success
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls MyFirstApp/
appinfo.json  icon.png  index.html  largeIcon.png  webOSjs-0.1.0
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```



### 기본 자바스크립트 서비스 템플릿 생성

- -t 옵션으로 js\_service 템플릿 선택
- 디렉토리명은 MyFirstService
- Service ID 는 com.sample.app.service

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-generate -t js_service MyFirstService
? service id com.sample.app.service
Generating js_service in /home/nicolas/kbwork/MyFirstService
Success
```

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls MyFirstService
helloclient.js  helloworld_webos_service.js  package.json  services.json
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

## ares-package

- 애플리케이션 또는 자바 스크립트 서비스를 패키지 파일(.ipk)에 패키징

```
ares-package [OPTION...] APP_DIR [SERVICE_DIR]
```

```
ares-package --version
```

```
ares-package --help|-h
```

## 👤 앱 단독 패키징

- 한 개의 앱을 패키징하고 IPK 디렉터리에 생성

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-package MyFirstApp -o IPK
Progressing...
Creating package com.sample.app_1.0.0_all.ipk in /home/nicolas/kbwork/IPK
Success
```

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls IPK
com.sample.app_1.0.0_all.ipk
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

- 한 개의 앱을 패키징하고 현재 디렉터리에 생성

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-package MyFirstApp
Progressing...
Creating package com.sample.app_1.0.0_all.ipk in /home/nicolas/kbwork
Success
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls
IPK  MyFirstApp  MyFirstService  com.sample.app_1.0.0_all.ipk
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

## 👤 서비스와 함께 패키징

- 한 개의 앱과 서비스를 패키징하고 IPK 디렉터리에 생성

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-package MyFirstApp MyFirstService -o IPK
Progressing...
Creating package com.sample.app_1.0.0_all.ipk in /home/nicolas/kbwork/IPK
Success

nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls IPK
com.sample.app_1.0.0_all.ipk
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

- 한 개의 앱과 서비스를 패키징하고 현재 디렉터리에 생성

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-package MyFirstApp MyFirstService
Progressing...
Creating package com.sample.app_1.0.0_all.ipk in /home/nicolas/kbwork
Success
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls
IPK  MyFirstApp  MyFirstService  com.sample.app_1.0.0_all.ipk
```

### ares-install

- 패키지 파일(.ipk)을 지정한 디바이스에 설치 또는 삭제
- 디바이스에 설치된 애플리케이션의 목록을 확인

```
ares-install [OPTION...] PKG_FILE
ares-install [OPTION...] --remove|-r APP_ID
ares-install [OPTION...] --list|-l
ares-install [OPTION...] --listfull|-F
ares-install --device-list|-D
ares-install --version|-V
ares-install --help|-h
```

### 🔍 현재 설정된 디바이스 목록 확인

- -D 옵션으로 현재 설정된 디바이스 목록을 확인

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-install -D
name      deviceinfo      connection
-----
device    root@10.177.232.180:22  ssh
target    root@10.177.233.75:22  ssh
rpi        root@10.177.232.119:22  ssh
```

### 📦 패키지 파일을 디바이스에 설치

- -d 옵션으로 설치할 디바이스 지정 후, 패키지 파일을 설치

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-install -d rpi IPK/com.sample.app_1.0.0_all.ipk
Installing package IPK/com.sample.app_1.0.0_all.ipk
Success
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

### 🔗 디바이스에 설치된 앱 목록 확인

- -l 옵션으로 디바이스에 설치된 앱 목록을 확인
- -F 옵션으로 디바이스에 설치된 앱의 상세 목록을 확인

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-install -d rpi -l  
com.sample.app  
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-install -d rpi -F  
-----  
id:com.sample.app, systemApp:false, removable:true, icons:, defaultWindowType:card, largeIcon:largeIcon.png, main:index.html, type:web, vendor:My Company, transparent:false, uiRevision:2, title:my first app, visible:true, inspectable:true, spinnerOnLaunch:true, trustLevel:default, version:1.0.0, unmovable:false, privilegedJail:false, icon:icon.png, noSplashOnLaunch:false, handlesRelaunch:false, deeplinkingParams:, folderPath:/media/developer/apps/usr/palm/applications/com.sample.app,  
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

### 🔗 디바이스에 설치된 앱 삭제

- -r 옵션으로 디바이스에 설치된 앱 삭제

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-install -d rpi -r com.sample.app  
Removed package com.sample.app  
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```



### ares-launch

- 디바이스에 설치된 애플리케이션을 실행 또는 종료

```
ares-launch [OPTION...] APP_ID
ares-launch [OPTION...] --close APP_ID
ares-launch [OPTION...] --running|-r
ares-launch --version|-V
ares-launch --help|-h
```

### 🐙 현재 설정된 디바이스 목록 확인

- -D 옵션으로 현재 설정된 디바이스 목록을 확인

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-launch -D
name      deviceinfo      connection
-----
device    root@10.177.232.180:22  ssh
target    root@10.177.233.75:22  ssh
rpi        root@10.177.232.119:22  ssh
```

### 🐙 설치된 앱을 원격으로 실행

- -d 옵션으로 실행할 앱의 디바이스 지정 후, App ID로 실행

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-launch -d rpi com.sample.app
Launched application com.sample.app
nicolas@nicolas-14U530-MFLGL:~/kbwork$ █
```

### 디바이스에서 실행 중인 앱 목록 확인

- -r 옵션으로 디바이스에서 실행 중인 앱의 목록을 확인

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-launch -d rpi -r  
com.sample.app  
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

### 디바이스에서 실행 중인 앱을 원격으로 종료

- -c 옵션으로 종료할 앱의 ID를 지정

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-launch -d rpi -c com.sample.app  
Closed application com.sample.app  
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

## ares-inspect

- 노드 인스펙터와 웹 인스펙터를 실행

```
ares-inspect [OPTION...] [--app|-a] APP_ID  
ares-inspect [OPTION...] --service|-s SERVICE_ID  
ares-inspect --version|-V  
ares-inspect --help|-h
```

### 🔗 현재 설정된 디바이스 목록 확인

- -D 옵션으로 현재 설정된 디바이스 목록을 확인

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-inspect -D
name      deviceinfo      connection
-----
device    root@10.177.232.180:22  ssh
target    root@10.177.233.75:22  ssh
rpi       root@10.177.232.119:22  ssh
```

### 🔗 설치된 앱을 원격에서 디버깅

- -d 옵션으로 실행할 앱의 디바이스 지정 후, App ID로 디버거 실행 (링크)
- -o 옵션을 주면 호스트 브라우저에서 바로 실행 (디폴트 브라우저)

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-inspect -d rpi com.sample.app
Application Debugging - http://localhost:38235/devtools/inspector.html?ws=localhost:38235/devtools/page/807B8C6167C8FA8A72F1883143168019
```

## 설치된 서비스를 원격에서 디버깅

- -d 옵션으로 실행할 앱의 디바이스 지정 후, -s 옵션에 Service ID로 디버거 실행 (링크)
- -o 옵션을 주면 호스트 브라우저에서 바로 실행 (디폴트 브라우저)

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-inspect -d rpi -s com.sample.app.service  
nodeInsptUrl: http://localhost:39792/debug?port=5886
```

### ares-shell

- 디바이스의 Shell로 연결 하거나 바로 실행

```
ares-shell -d TARGET_DEVICE  
ares-shell -d TARGET_DEVICE -r CMD  
ares-shell --device-list|-D  
ares-shell --version  
ares-shell --help|-h
```

### 👤 디바이스 셸로 직접 연결

- -d 옵션으로 디바이스 지정

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-shell -d rpi
>>> Start rpi shell.
>>> Type `exit` to quit.

root@raspberrypi3:~# exit

>>> Terminate the shell, bye.
```

### 👤 디바이스 셸을 실행하지 않고 명령 실행

- -r 옵션으로 명령어 직접 실행

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-shell -d rpi -r "luna-send -n 1 -f luna://com.webos.service.systemservice/deviceInfo/query '{\"parameters\": [\"wired_addr\"]}'"
{
  "wired_addr": "B8:27:EB:29:57:AF",
  "returnValue": true
}
```



# webOS OSE SDK - Emulator



## 🐧 QEMUx86 에뮬레이터

- 애플리케이션과 플랫폼을 가상 환경에서 테스트하기 위한 도구로 라즈베리파이 없이도 주요 기능들을 테스트할 수 있음
- 제약 사항 : x86(x86\_64) / Linux Ubuntu

## 🐧 에뮬레이터 구성

- QEMUx86 용 webOS OSE 이미지 (webos-image-qemux86.vmdk)
- Emulator Package (\$ git clone <https://github.com/webosose-emulator/prebuilt-emulator.git>)

\* QEMUx86용 pre-built 이미지 다운로드 : <http://build.webos-ports.org/webosose/>

### Prerequisites

- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- 깃 설치 : \$ sudo apt-get -f install git

### 이미지 빌드

- 이미지 경로 : BUILD/deploy/images/qemux86/webos-image-qemux86.vmdk

```
$ git clone https://github.com/webosose/build-webos.git
$ cd build-webos
$ sudo scripts/prerequisites.sh
$ ./mcf -p 0 -b 0 qemux86
$ source oe-init-build-env
$ bitbake webos-image
```

## Prerequisites

- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo apt-get -f install git libtool libdrm-dev libepoxy-dev libgbm-dev libegl1-mesa-dev libgtk-3-dev autoconf

## virglrenderer 빌드

- 그래픽을 렌더링하기 위한 라이브러리

```
$ git clone git://anongit.freedesktop.org/virglrenderer
$ cd virglrenderer
$ git checkout virglrenderer-0.6.0
$ ./autogen.sh
$ ./configure --with-glx
$ make && sudo make install
```

## Prerequisites

- \$ sudo apt-get update
- \$ sudo apt-get upgrade
- \$ sudo apt-get -f install libsdl2-dev libpixmap-1-dev

## QEMU 빌드

- virglrenderer 패키지가 빌드 된 후 QEMU를 빌드해야 함

```
$ git clone https://github.com/webosose-emulator/qemu.git
$ cd qemu
$ ./configure --target-list=i386-softmmu --enable-sdl --with-sdlabi=2.0 --audio-drv-
list=alsa,pa
$ make
```

## 🐧 Pre-built 에뮬레이터 클론

- `$ git clone https://github.com/webosose-emulator/prebuilt-emulator.git`

## 🐧 라이브러리 교체

- 64bit Ubuntu
  - `$ cp /usr/local/lib/libvirglrenderer.so* prebuilt-emulator/lib/x86_64`
  - `$ cp <qemu directory path>/i386-softmmu/qemu-system-i386 prebuilt-emulator/bin/x86_64`
  - `$ cp -rf pc-bios prebuilt-emulator/bin`
- 32bit Ubuntu
  - `$ cp /usr/local/lib/libvirglrenderer.so* prebuilt-emulator/lib/x86`
  - `$ cp <qemu directory path>/i386-softmmu/qemu-system-i386 prebuilt-emulator/bin/x86`
  - `$ cp -rf pc-bios prebuilt-emulator/bin`

## 🔗 JSON 설정파일 생성

- \$ cd <pre-built emulator directory path>
- \$ mv webos-config-sample.json webos-config.json
- \$ vi webos-config.json

```
{
  "description": "qemux86",
  "name": "webos-image-qemux86-master-20180925121427",
  "vmdk_file_path": "~/swcamp/qemu_image/webos-image-qemux86-master-20180925121427.vmdk",
  "hw.core": "1",
  "hw.ramSize": "1024",
  "hw.accel": "true",
  "hw.gl.accel": "true",
  "debug": "false",
  "portforwarding.SSH": "6622",
  "portforwarding.inspector": "9998"
```

## 에뮬레이터 실행하기

- `$ cd <pre-built emulator directory path>`
- `$ ./emulator webos-config.json`

## Ssh로 에뮬레이터 연결하기

- `$ ssh -p 6622 -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null root@localhost`

## 브라우저에서 연결하기

- 브라우저에서 localhost:9998 입력
- 단, CLI로 설치한 웹 앱만 적용됨



# webOS OSE SDK - Beanviser



## 🐧 빈바이저란?

- 원격으로 디바이스와 통신하여 앱과 서비스의 성능을 측정하거나 디바이스를 모니터링 하기 위한 도구

## 🐧 주요 기능

- System 의 CPU/메모리
- Process 의 CPU/메모리
- 로그 확인
- LS 모니터
- Gstreamer
- 로그파일 비교



## 시스템 요구사항

### Software Requirements

OS	Requirement
Linux	Ubuntu 14.04, 16.04 or Ubuntu Mint (Maya)
macOS	10.12 Sierra and High Sierra (64-bit)
Windows	Windows 7 or later

### Hardware Requirements

HW Component	Requirement
CPU	Intel® Pentium®4 2.0 GHz or faster
Memory	Minimum 3 GB
VGA	1280 x 1024 or greater
Graphic Card	256 MB or more of video memory

## 다운로드

- 개발자사이트 > Develop > SDK Tools > Beanviser > Downloading Beanviser Package
- <http://webosose.org/develop/sdk-tools/beanviser/download-launch-beanviser/>

SDK Download

SDK Terms and Conditions

Before downloading the SDK, you must agree to the terms and conditions. Please read the following terms and conditions carefully before proceeding.

Software Development Kit License Agreement

IMPORTANT: PLEASE READ THE TERMS AND CONDITIONS OF THIS USER LICENSE AGREEMENT ("LICENSE AGREEMENT") CAREFULLY BEFORE DOWNLOADING, INSTALLING AND/OR USING LG SOFTWARE DEVELOPMENT KIT , INCLUDING ANY UPDATES THERETO AND ANY RELATED DOCUMENTATION (COLLECTIVELY, THE "SDK", AS MORE PARTICULARLY DEFINED BELOW). LG ELECTRONICS, INC. ("LGE") IS WILLING TO LICENSE THE SDK TO YOU AS THE INDIVIDUAL, THE COMPANY, OR THE LEGAL ENTITY THAT WILL BE USING THE SDK ("YOU" OR "YOUR") ONLY ON

☐ I have read and agree with the above terms and conditions

Download

## 압축 해제

- Home 디렉토리에 압축 해제

```
nicolas@nicolas-14U530-MFLGL:~$ tar -xvf ~/Downloads/beanviser-linux64-2.0.0.tgz
```

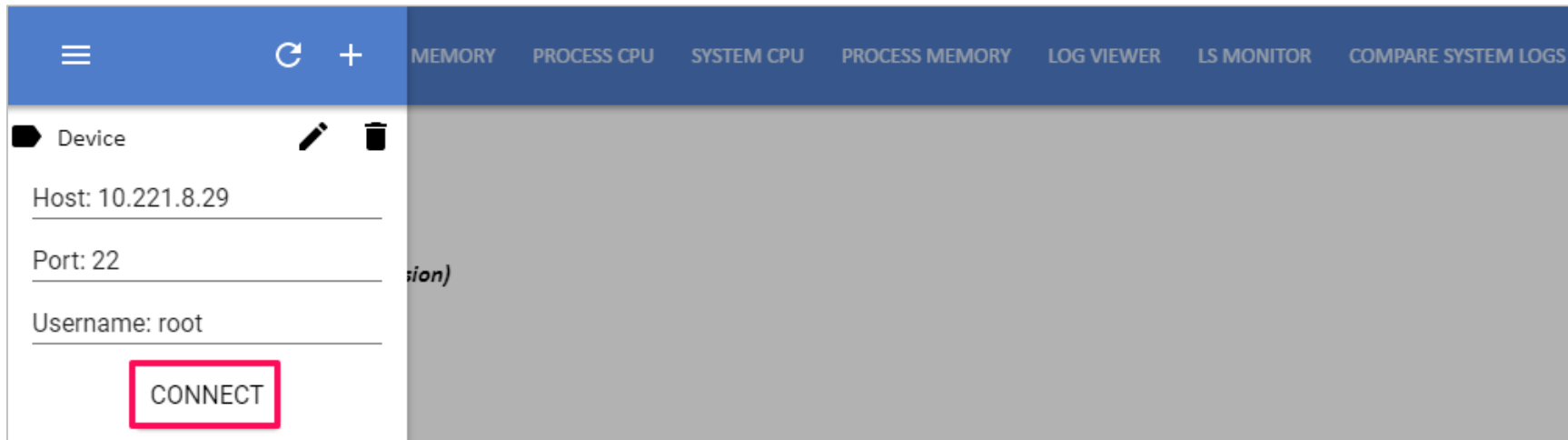
```
nicolas@nicolas-14U530-MFLGL:~$ cd beanviser/  
nicolas@nicolas-14U530-MFLGL:~/beanviser$ ls  
LGE_Open_Source_Software_Notice-Beanviser.html  ares-cli          service  
LGE_Open_Source_Software_Notice-CLI.html        beanviser.sh      version  
ReadMe.txt                                       disable_udc.sh    viewer  
Release_notes.txt                               packaging          webos.ose.beanviser_2.0.0_all.ipk
```

## 빈바이저 실행

- 리눅스 & 맥 : beanviser.sh 실행
- 윈도우 : beanviser.cmd 실행

## 디바이스와 연결

- 설정 버튼을 눌러 나오는 메뉴에서 + 를 클릭하여 디바이스 추가
- 에뮬레이터는 Host : 127.0.0.1, Port : 6622
- Connect 버튼을 눌러 연결



## 라즈베리 파이 추가 및 연결

- 설정 버튼을 눌러 나오는 메뉴에서 + 를 클릭하여 디바이스 추가
- Name : rpi
- Host : < 라즈베리파이에 설정한 주소 >
- Port : 22
- ADD 버튼을 눌러서 신규 디바이스 등록
- CONNECT 버튼을 눌러서 연결

## 측정 해보기

- System 및 Process CPU / Memory
  - Start 버튼을 눌러 확인하기
- Log viewer
  - Start 버튼을 누른 후
  - `ares-launch -d rpi com.sample.app` 을 Host PC에서 실행 후 결과 보기
- LS Monitor
  - Start 버튼을 누른 후
  - `luna-send -n 1 -f luna://com.webos.service.systemservice/clock/getTime '{}'` 을 라즈베리파 이 콘솔에서 실행 후 결과 보기



# webOS OSE SDK - Other



# Luna-send 사용하기

## 🐼 Luna-send 란?

- 디바이스의 Shell에서 Luna Service bus 상의 서비스가 가진 메소드를 호출하기 위한 도구
- 문법 : device# luna-send [OPTIONS...] URL '{JSON object}'

### OPTIONS

Option	Parameter	Description
-h	None	Displays the help.
-P	None	Deprecated
-s	None	Sends a signal.
-a	None	Sends a specified appld in a message. (default is none)
-m	None	Displays service name. (default is none)
-d	None	Turns debug logging on.
-i	None	Turns interactive mode on.
-t	NUMBER_OF_TIMES	Displays average response time after calling this command the number of times.
-n	NUMBER_OF_REPLIES	Exits interactive mode after the number of replies.
-l	None	Displays the order of responses.
-f	None	Displays the response in the JSON format.
-q	QUERY_NAME	Displays a specific query item from responses. (multiple queries can be supplied) e.g.: -q 'returnValue' -q 'queues[0]'
-w	TIMEOUT	Sets exit timeout value.

```
luna-send -n 1 -f luna://com.webos.service.applicationManager/launch
'{"id": "com.webos.app.browser"}'

{
  "returnValue": true
}
```

```
luna-send -n 1 luna://com.webos.settingservice/getSystemSettings '{"keys":["localeInfo"]}'

{"subscribed":false,"method":"getSystemSettings","settings":{"localeInfo":{"locales":
{"UI":"ko-KR"}}},"returnValue":true}
```

```
luna-send -n 1 luna://com.webos.settingservice/setSystemSettings '{"category":"option",
"settings":{"country":"KOR"}}'

{"method":"setSystemSettings","returnValue":true}
```

# Luna-send 사용하기

---

## 🔗 Luna-send 실습하기?

- 디바이스의 Shell 또는 CLI의 ares-shell 명령 사용
- Notification API 문서를 참조하여 토스트 생성
  - Method : createToast
  - 파라미터 : sourceId & message


```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-shell -d rpi -r "luna-send -n 1 -f luna://com.webos.notification/createToast '{\"sourceId\": \"com.webos.notification\", \"message\" : \"Hello\"}'"
{
  "toastId": "com.webos.notification-1557995184539",
  "returnValue": true
}
```



# ls-monitor 사용하기

## 🐼 ls-monitor 란?

- 디바이스의 Shell에서 Luna Service bus 상의 메시지를 보기위한 도구
- 문법 : device# ls-monitor [OPTIONS...]

### OPTIONS

Option	Parameter	Description
-h, --help	None	Displays the help.
-f, --filter	SERVICE_NAME	Filters by service name (or unique name). e.g. <i>-f com.webos.foo</i> or <i>--filter=com.webos.foo</i>
-l, --list	None	Lists all luna services connected to a hub.
-s, --subscriptions	None	Lists all subscriptions in the system.
-i, --introspection	SERVICE_NAME	Lists service methods and signals. e.g. <i>-i com.webos.foo</i> or <i>--introspection=com.webos.foo</i>
-v, --api-version	SERVICE_NAME	Displays the version of the service. e.g. <i>-v com.webos.foo</i> or <i>--api-version=com.webos.foo</i>
-m, --malloc	None	Lists memory allocation data from all services in the system.
-d, --debug	None	Prints extra output for debugging monitor. <div> Be careful of unbounded memory growth when using the debug option.</div>

-c, --compact	None	Prints compact output to fit terminal. The followings indicate the symbols in the message printed in the Console. <ul style="list-style-type: none"><li>• &gt;*: Signal</li><li>• &gt; : Cancel method call</li><li>• &gt;: Method call</li><li>• &lt;: Reply</li></ul> e.g. <i>TX c.w.s.activemanage &gt;  TX c.w.s.activemanage/private/cancel</i> <div> This option takes precedence over the debug option.</div>
-j, --json	None	Prints JSON-formatted output for easy parsing. <div> This option takes precedence over the debug option.</div>
-t, --sort-by-timestamps	None	Sorts output by timestamp instead of serial.
--dump-hub-data-csv	None	Dumps hub data in CSV format.

# ls-monitor 사용하기

## 🐼 ls-monitor 실습?

- -i 옵션으로 notification 서비스의 카테고리 와 메소드들 보기

```
root@raspberrypi3:~# ls-monitor -i com.webos.notification

METHODS AND SIGNALS REGISTERED BY SERVICE 'com.webos.notification' WITH UNIQUE NAME 'GwesQbtk' AT HUB

"/":
  "getNotificationInfo": {"provides":["private","notifications","all","notifications.query"]}
  "getNotification": {"provides":["private","notifications","all","notifications.query"]}
  "closeAllAlerts": {"provides":["private","notifications","all"]}
  "removeAllNotification": {"provides":["private","notifications","all"]}
  "enable": {"provides":["private","notifications","all"]}
  "disableToast": {"provides":["private","notifications","all"]}
  "closeInputAlert": {"provides":["private","notifications","all"]}
  "createNotification": {"provides":["private","notifications.manage","notifications","all"]}
  "enableToast": {"provides":["private","notifications","all"]}
  "getToastNotification": {"provides":["private","notifications","all"]}
  "createInputAlert": {"provides":["private","notifications","all"]}
  "closePincodePrompt": {"provides":["private","notifications","all"]}
  "removeNotification": {"provides":["private","notifications.manage","notifications","all"]}
  "getAlertNotification": {"provides":["private","notifications","all"]}
  "disable": {"provides":["private","notifications","all"]}
  "createPincodePrompt": {"provides":["private","public","notifications","all"]}
  "closeAlert": {"provides":["private","public","notifications","all"]}
  "createAlert": {"provides":["activities.callbacks","private","public","notifications","all"]}
  "getInputAlertNotification": {"provides":["private","notifications","all"]}
  "getPincodePromptNotification": {"provides":["private","notifications","all"]}
  "closeToast": {"provides":["private","public","notifications","all"]}
  "createToast": {"provides":["activities.callbacks","private","notifications.manage","public","n
otifications","all"]}
```

# ABOUT webOS ?

QUESTIONS ?

? ANSWERS



# webOS

Open Source Edition

