

build steps of these packages

0,make a full build firstly, otherwise these packages will not build pass.

after the full build is done, you can build below 8 packages one by one.

1)run ". build/envsetup.sh" in the root dir of cupcake source tree

2)run "tapas" command in the root dir of cupcake source tree:

example:

```
root@b361-desktop:/home/b361/svn/cupcake/device# tapas
```

Build for the simulator or the device?

1. Device
2. Simulator

Which would you like? [1]

Build type choices are:

1. release
2. debug

Which would you like? [1]

Product choices are:

1. emulator
2. generic
3. sim
4. dream-open

You can also type the name of a product if you know it.

Which would you like? [generic]

Variant choices are:

1. user
2. userdebug
3. eng

Which would you like? [eng]

3)make

1, bluez

1)in cupcake source tree, replace "external/bluez" folder with content of bluez.tar.gz

2)run ". build/envsetup.sh" in the root dir of cupcake source tree

3)mmm external/bluez

2, openobex

1)extract openobex.tar.gz to "external" folder of cupcake source tree

2)run ". build/envsetup.sh" in the root dir of cupcake source tree

3)mmm external/openobex

3, webkit

1)backup and remove "external/webkit" folder in cupcake source tree

2)extract webkit_open_source.zip(this ZIP file include folder wap, folder webkit and file libwap.a) to "external" folder

3)mkdir -p out/target/product/generic/obj/STATIC_LIBRARIES/libwap_intermediates/

4)copy libwap.a in webkit_open_source.zip to the "libwap_intermediates" folder

5)run ". build/envsetup.sh" in the root dir of cupcake source tree

6)make SUPPORT_WAP=true libwebcore

4, ppp

- 1) backup and remove "external/ppp" folder in cupcake source tree
- 2) extract ppp.tar.gz to "external" folder of cupcake source tree
- 3) run ". build/envsetup.sh" in the root dir of cupcake source tree
- 4) mmm external/ppp

5, javaVM

- 1) extract javaVM.tar.gz to anywhere, enter this extracted folder, remove dir "build_output"
- 2) run "chmod 777 build.sh", then run "./build.sh" command.
- 3) type your jdk(recommend 1.4, other jdk may cause error) full path,
- 4) type Android ndk(recommend 1.5, other ndk may cause error) full path, then the build starts.

6, Linux kernel

- 1) ensure you have installed arm-linux-4.1.1
- 2) extract lnx2625pxa.tar.gz to anywhere, enter this extracted folder
- 3) run "make jadestd_defconfig"
- 4) run "make zImage"

7, blob (tip: must build Linux kernel firstly before build this package)

- 1) ensure you have installed arm-linux-4.1.1
- 2) extract blob.tar.gz to anywhere, enter this extracted folder, enter src/ directory
- 3) run "./setup.sh PRODUCT=JADESTD linux_kernel_path(your Linux kernel path---full path of '6, Linux kernel')"
- 4) run "make"

8, wlan

- 1) extract wlan.tar.gz to "external" folder in cupcake source tree
- 2) backup and remove "wpa_supplicant" folder in "external/" folder of cupcake source tree
- 3) backup and replace "wifi" folder in hardware/libhardware_legacy/ of cupcake with the "wifi" folder in this package
- 4) move "wapi" folder in this wlan.tar.gz to hardware/libhardware_legacy/ of cupcake
(if no wapi folder in wlan.tar.gz, skip this step)
- 5) mv wapi.h to hardware/libhardware_legacy/include/hardware_legacy/wapi.h
(if no wapi.h file in wlan.tar.gz, skip this step)
- 6) run ". build/envsetup.sh" in the root dir of cupcake source tree
- 7) make external/wlan

any build issue please contact zhijia.zheng@borqs.com, tel:13401003877