Power Management Debugging Guide

Feb, 2012

Revision History

Date	Version	Description
2011-03-07	1.00	1st draft
2012-02-29	1.01	Update to Android 4.0 (Ice Cream Sandwich)

TABLE OF CONTENTS

Contents

1. Introduction

This guide shows Android power management - especially Suspend and Resume feature which is based on Linux “Suspend To RAM” and how to setup kernel options to debug Android PM system.

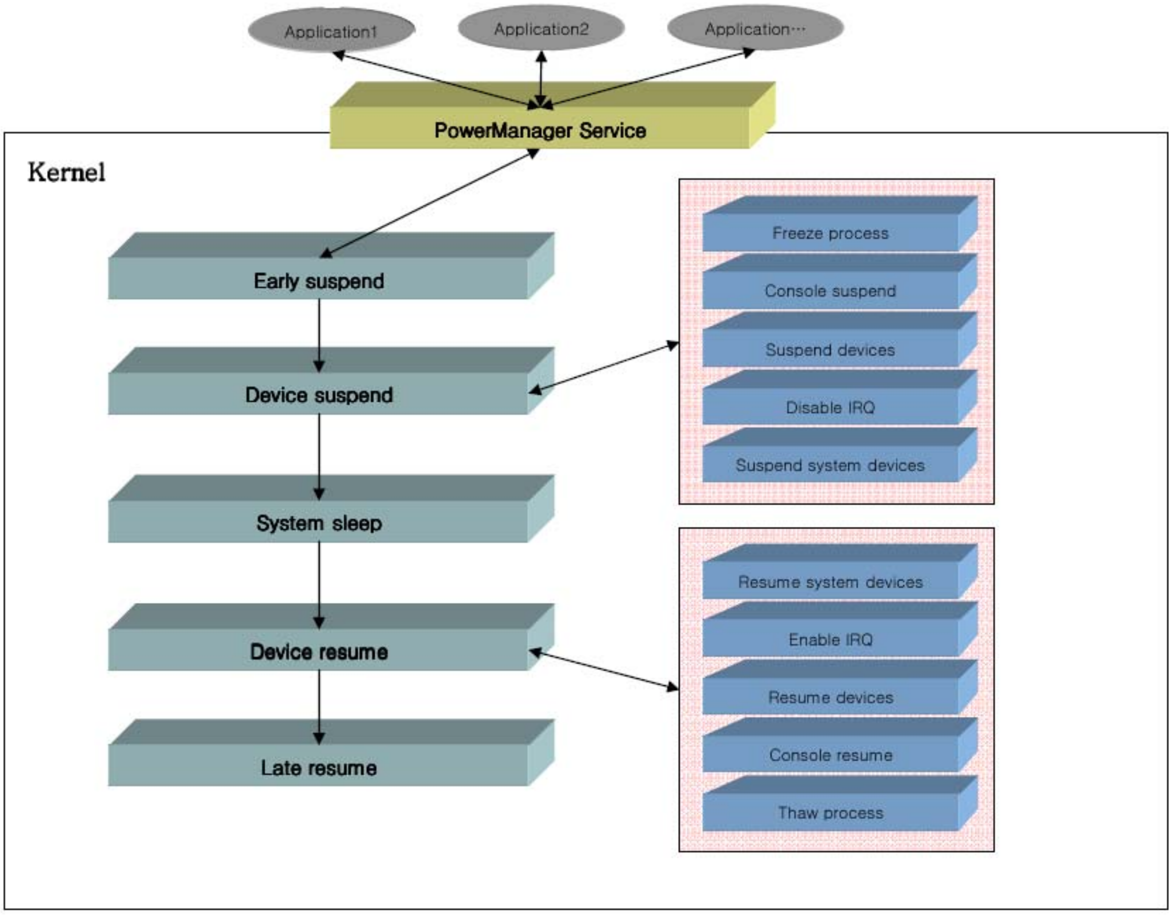
This guide is made for TCC Android Platform based on Ice Cream Sandwich version.

2. Android Power Management

This chapter shows how to suspend system and resume from suspend mode.

Android suspends system based on Linux “Suspend To RAM” and improve it by introducing early suspend and late resume.

2.1. State Transition



To suspend the system and resume from suspend, the system experiences following 5 states.

- Early Suspend
- Device Suspend
- System Sleep
- Device Resume
- Late Resume

2.2. Early Suspend

If the power state is changed by user or specified screen waiting time, the system goes into early suspend mode.

When entering early suspend mode, early suspend handler is called and call-back function is executed. LCD will be turn off in this mode.

2.3. Device Suspend

If there is no wake-locked applications and no input for a specified time in early-suspend mode, the system enters into suspend mode.

Before entering suspend mode, all device drivers have to be suspended and it is the main process to enter sleep mode. To enter sleep mode, following operations are performed.

① Freeze Process
User space process and other tasks are stopped.

② Console Suspend
Disabling UART console.
This can be skipped by changing kernel option for debugging purpose.

③ Suspend Devices
Performs call back function to suspend all non-syscore device including platform devices.

④ Disable IRQ

⑤ Suspend System Devices
Performs call back function to suspend syscore device.

2.4. System Sleep

System enters into shut down mode. If there is hardware reset or RTC wake up, wake-up operation will be started. The detailed operation will be different according to machine type and board.

2.5. Device Resume

After wake-up, the required operation for each driver will be done to resume pervious status. Operation procedure is as follows.

① Resume system devices
Performs call back function to resume syscore device.

② Enable IRQ

③ Resume devices
Performs call back for all non-syscore device.

④ Console resume
Activate Console.

⑤ Thaw process
Activate user process and all tasks.

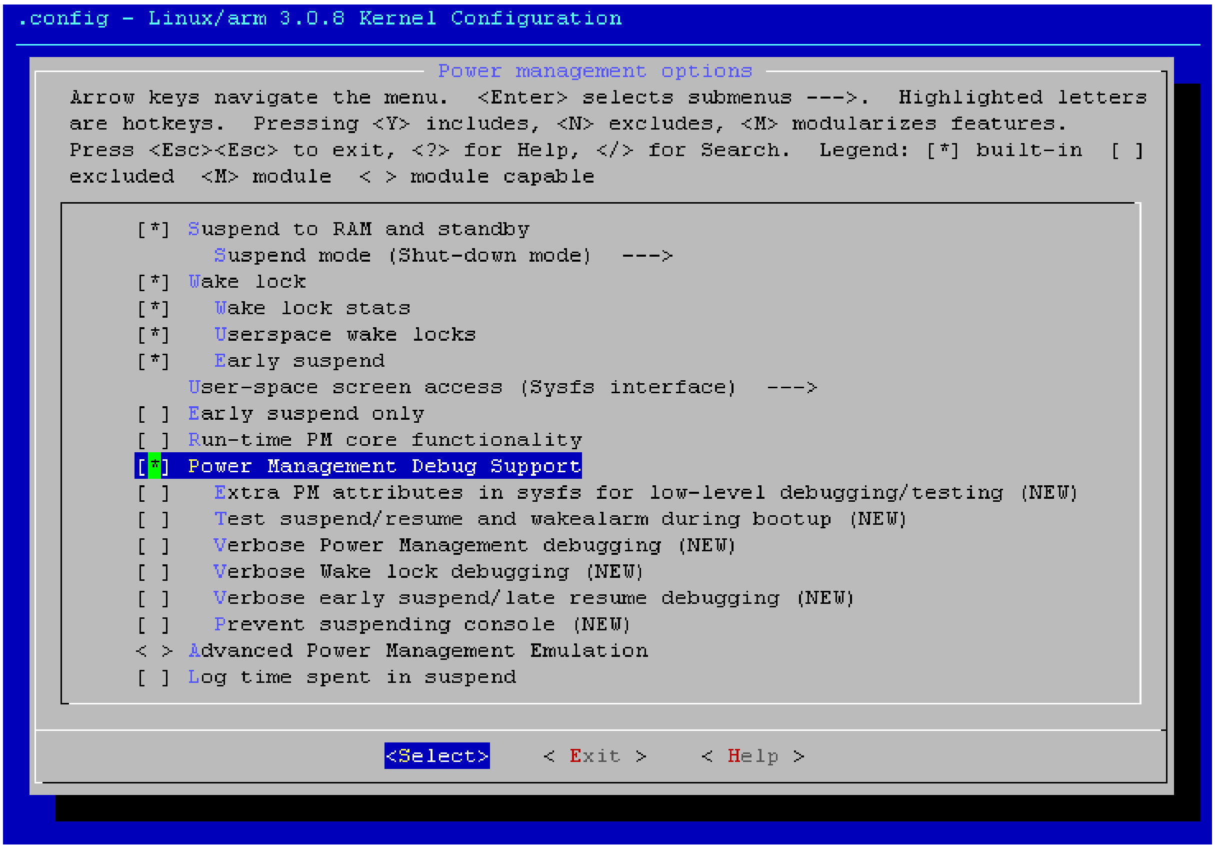
2.6. Late Resume

After finishing device resume, late resume will be performed. LCD will be turned on during late resume.

3. Setting Power management Debugging Option

There is several options to debug power management in kernel menu configuration.
By selecting this option, you can show various events which are happened during state transition.
This chapter shows those options in detail.

After executing kernel menuconfig, select Power management options -> Power Management Debug support. Various sub menu will be displayed.



```
.config - Linux/arm 3.0.8 Kernel Configuration

                          Power management options
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters
  are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.
  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in  [ ]
  excluded  <M> module  < > module capable

        [*] Suspend to RAM and standby
              Suspend mode (Shut-down mode)  --->
        [*] Wake lock
        [*]   Wake lock stats
        [*]   Userspace wake locks
        [*]   Early suspend
            User-space screen access (Sysfs interface)  --->
        [ ] Early suspend only
        [ ] Run-time PM core functionality
        [*] Power Management Debug Support
        [ ]   Extra PM attributes in sysfs for low-level debugging/testing (NEW)
        [ ]   Test suspend/resume and wakealarm during bootup (NEW)
        [ ]   Verbose Power Management debugging (NEW)
        [ ]   Verbose Wake lock debugging (NEW)
        [ ]   Verbose early suspend/late resume debugging (NEW)
        [ ]   Prevent suspending console (NEW)
        < > Advanced Power Management Emulation
        [ ] Log time spent in suspend

                       <Select>    < Exit >    < Help >
```

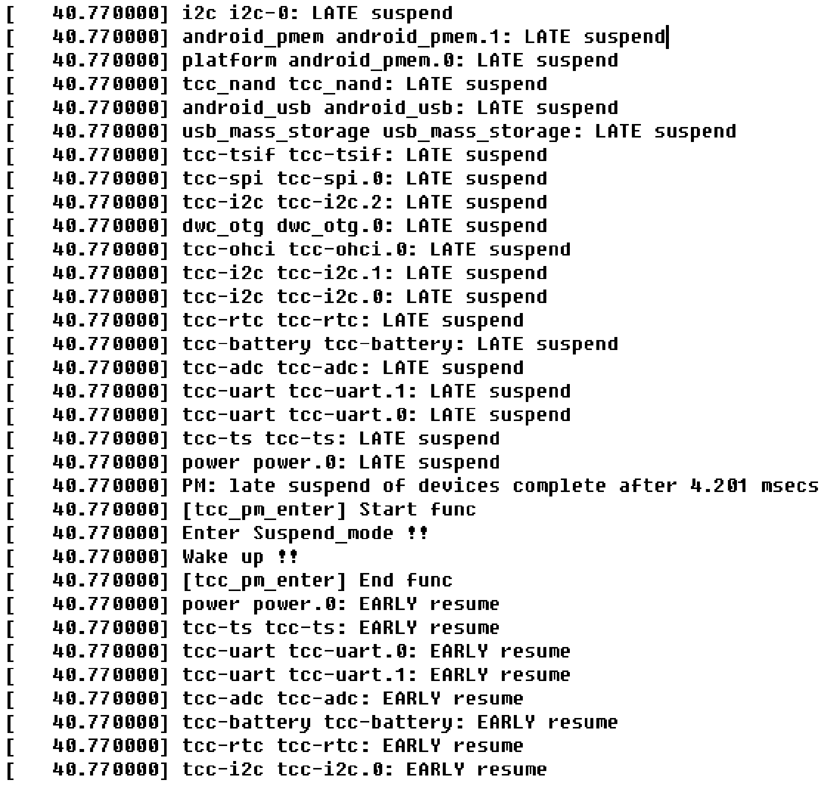
Options to be used to debug is as follows.

- Verbose Power Management debugging
- Verbose Wake lock debugging
- Verbose early suspend / late resume debugging
- Prevent suspending console
- Test suspend/resume and wakealarm during bootup

3.1. Verbose Power Management debugging

By enabling this option, the detailed operating status during Device Suspend and Device Resume will be out.

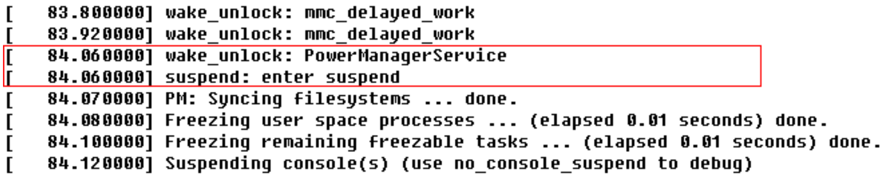
You can check suspend/resume call back function of most device driver include platform device.



```
[   40.770000] i2c i2c-0: LATE suspend
[   40.770000] android_pmem android_pmem.1: LATE suspend
[   40.770000] platform android_pmem.0: LATE suspend
[   40.770000] tcc_nand tcc_nand: LATE suspend
[   40.770000] android_usb android_usb: LATE suspend
[   40.770000] usb_mass_storage usb_mass_storage: LATE suspend
[   40.770000] tcc-tsif tcc-tsif: LATE suspend
[   40.770000] tcc-spi tcc-spi.0: LATE suspend
[   40.770000] tcc-i2c tcc-i2c.2: LATE suspend
[   40.770000] dwc_otg dwc_otg.0: LATE suspend
[   40.770000] tcc-ohci tcc-ohci.0: LATE suspend
[   40.770000] tcc-i2c tcc-i2c.1: LATE suspend
[   40.770000] tcc-i2c tcc-i2c.0: LATE suspend
[   40.770000] tcc-rtc tcc-rtc: LATE suspend
[   40.770000] tcc-battery tcc-battery: LATE suspend
[   40.770000] tcc-adc tcc-adc: LATE suspend
[   40.770000] tcc-uart tcc-uart.1: LATE suspend
[   40.770000] tcc-uart tcc-uart.0: LATE suspend
[   40.770000] tcc-ts tcc-ts: LATE suspend
[   40.770000] power power.0: LATE suspend
[   40.770000] PM: late suspend of devices complete after 4.201 msecs
[   40.770000] [tcc_pm_enter] Start func
[   40.770000] Enter Suspend_mode !!
[   40.770000] Wake up !!
[   40.770000] [tcc_pm_enter] End func
[   40.770000] power power.0: EARLY resume
[   40.770000] tcc-ts tcc-ts: EARLY resume
[   40.770000] tcc-uart tcc-uart.0: EARLY resume
[   40.770000] tcc-uart tcc-uart.1: EARLY resume
[   40.770000] tcc-adc tcc-adc: EARLY resume
[   40.770000] tcc-battery tcc-battery: EARLY resume
[   40.770000] tcc-rtc tcc-rtc: EARLY resume
[   40.770000] tcc-i2c tcc-i2c.0: EARLY resume
```

3.2. Verbose Wake lock debugging

Android PM uses wake lock to control power state. By checking wakelock message, you can investigate overall operation of suspend system.

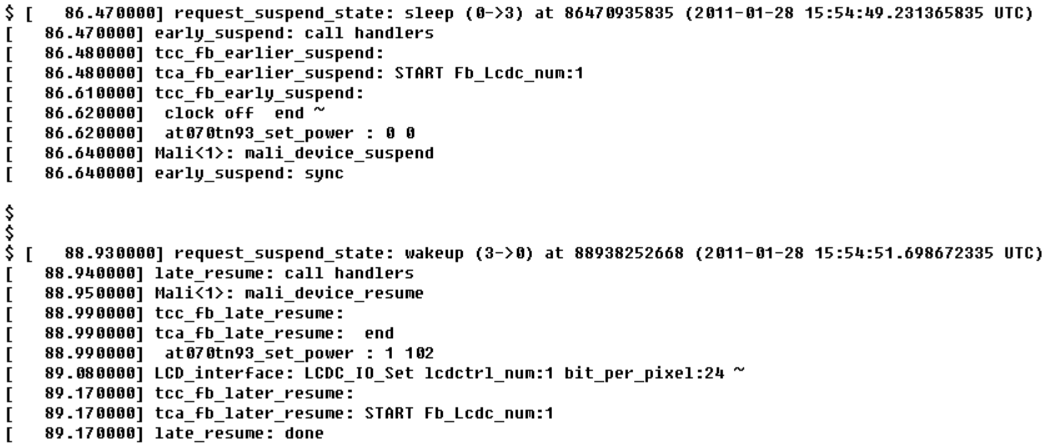


```
[   83.800000] wake_unlock: mmc_delayed_work
[   83.920000] wake_unlock: mmc_delayed_work
[   84.060000] wake_unlock: PowerManagerService
[   84.060000] suspend: enter suspend
[   84.070000] PM: Syncing filesystems ... done.
[   84.080000] Freezing user space processes ... (elapsed 0.01 seconds) done.
[   84.100000] Freezing remaining freezable tasks ... (elapsed 0.01 seconds) done.
[   84.120000] Suspending console(s) (use no_console_suspend to debug)
```

You can check wake lock/unlock of each device and PowerManagerService. You can also find when suspend operation begins.

3.3. Verbose Early Suspend / Late Resume debugging

By enabling this option, you can know when early suspend and late resume begin.



```
$ [   86.470000] request_suspend_state: sleep (0->3) at 86470935835 (2011-01-28 15:54:49.231365835 UTC)
[   86.470000] early_suspend: call handlers
[   86.480000] tcc_fb_earlier_suspend:
[   86.480000] tca_fb_earlier_suspend: START Fb_Lcdc_num:1
[   86.610000] tcc_fb_early_suspend:
[   86.620000]  clock off  end ~
[   86.620000]  at070tn93_set_power : 0 0
[   86.640000] Mali<1>: mali_device_suspend
[   86.640000] early_suspend: sync

$
$
$ [   88.930000] request_suspend_state: wakeup (3->0) at 88938252668 (2011-01-28 15:54:51.698672335 UTC)
[   88.940000] late_resume: call handlers
[   88.950000] Mali<1>: mali_device_resume
[   88.990000] tcc_fb_late_resume:
[   88.990000] tca_fb_late_resume:  end
[   88.990000]  at070tn93_set_power : 1 102
[   89.080000] LCD_interface: LCDC_IO_Set lcdctrl_num:1 bit_per_pixel:24 ~
[   89.170000] tcc_fb_later_resume:
[   89.170000] tca_fb_later_resume: START Fb_Lcdc_num:1
[   89.170000] late_resume: done
```

3.4. Prevent suspending console

To debug the problem which is happened during suspend/resume, debug message has to be shown until that point.

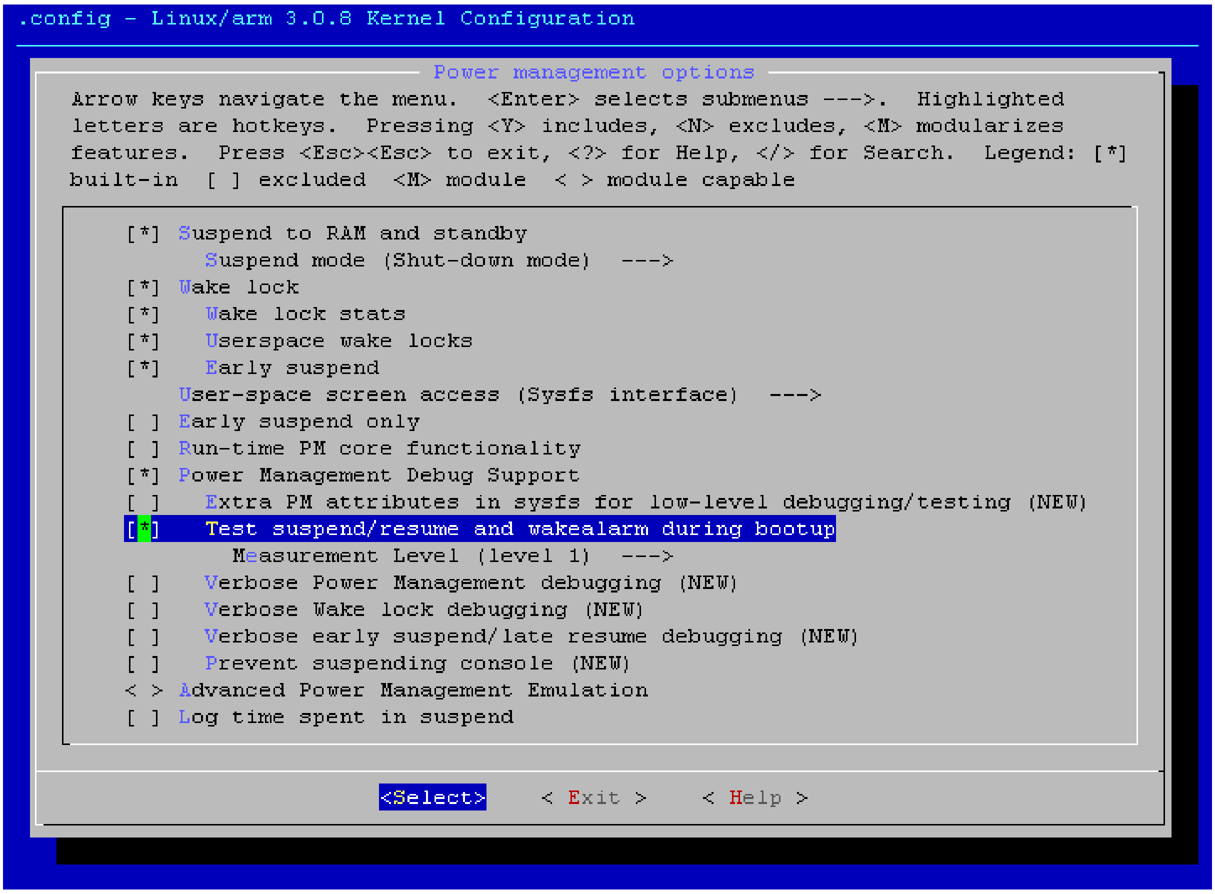
However, console will be disabled during the beginning of suspend, the message is not shown.

By enabling this option, PM driver maintains UART console until entering actual suspend state.

During resume, log message will be out from PM driver.

3.5. Time Measurement option

Power management options
-> Power Management Debug Support
-> Test suspend/resume and wakealarm during bootup.



```
.config - Linux/arm 3.0.8 Kernel Configuration

                    Power management options
  Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted
  letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes
  features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*]
  built-in  [ ] excluded  <M> module  < > module capable

      [*] Suspend to RAM and standby
            Suspend mode (Shut-down mode)  --->
      [*] Wake lock
      [*]   Wake lock stats
      [*]   Userspace wake locks
      [*]   Early suspend
          User-space screen access (Sysfs interface)  --->
      [ ] Early suspend only
      [ ] Run-time PM core functionality
      [*] Power Management Debug Support
      [ ]   Extra PM attributes in sysfs for low-level debugging/testing (NEW)
      [*]   Test suspend/resume and wakealarm during bootup
              Measurement Level (level 1)  --->
      [ ]   Verbose Power Management debugging (NEW)
      [ ]   Verbose Wake lock debugging (NEW)
      [ ]   Verbose early suspend/late resume debugging (NEW)
      [ ]   Prevent suspending console (NEW)
      < > Advanced Power Management Emulation
      [ ] Log time spent in suspend

                  <Select>    < Exit >    < Help >
```

If it takes long time to suspend/resume the system, we have to check time for each stage to enter sleep and resume.

By enabling this option, we can count overall time and the time required for each driver. Target to be measured will be changed according to measurement Level and the target for each level is as follows. Default level is ‘1’.

- In case of level 0, measure the total execution time it takes to PM transition.
- In case of level 1, include level 0, measure the execution time it takes to PM transition of each callbacks about non-syscore devices.
- In case of level 2, include level 1, measure the total execution time it takes to PM transition of syscore devices.
- In case of level 3, it is reserved.