

# Special Usage Model of the ACPI Control Method Lid Device

Copyright:

© 2016, Intel Corporation

Author:

Lv Zheng <[lv.zheng@intel.com](mailto:lv.zheng@intel.com)>

## Abstract

Platforms containing lids convey lid state (open/close) to OSPMs using a control method lid device. To implement this, the AML tables issue Notify(lid\_device, 0x80) to notify the OSPMs whenever the lid state has changed. The \_LID control method for the lid device must be implemented to report the "current" state of the lid as either "opened" or "closed".

For most platforms, both the \_LID method and the lid notifications are reliable. However, there are exceptions. In order to work with these exceptional buggy platforms, special restrictions and exceptions should be taken into account. This document describes the restrictions and the exceptions of the Linux ACPI lid device driver.

## Restrictions of the returning value of the \_LID control method

The \_LID control method is described to return the "current" lid state. However the word of "current" has ambiguity, some buggy AML tables return the lid state upon the last lid notification instead of returning the lid state upon the last \_LID evaluation. There won't be difference when the \_LID control method is evaluated during the runtime, the problem is its initial returning value. When the AML tables implement this control method with cached value, the initial returning value is likely not reliable. There are platforms always return "closed" as initial lid state.

## Restrictions of the lid state change notifications

There are buggy AML tables never notifying when the lid device state is changed to "opened". Thus the "opened" notification is not guaranteed. But it is guaranteed that the AML tables always notify "closed" when the lid state is changed to "closed". The "closed" notification is normally used to trigger some system power saving operations on Windows. Since it is fully tested, it is reliable from all AML tables.

## Exceptions for the userspace users of the ACPI lid device driver

The ACPI button driver exports the lid state to the userspace via the following file:

```
/proc/acpi/button/lid/LID0/state
```

This file actually calls the \_LID control method described above. And given the previous explanation, it is not reliable enough on some platforms. So it is advised for the userspace program to not to solely rely on this file to determine the actual lid state.

The ACPI button driver emits the following input event to the userspace:

- SW\_LID

The ACPI lid device driver is implemented to try to deliver the platform triggered events to the userspace. However, given the fact that the buggy firmware cannot make sure "opened"/"closed" events are paired, the ACPI button driver uses the following 3 modes in order not to trigger issues.

If the userspace hasn't been prepared to ignore the unreliable "opened" events and the unreliable initial state notification, Linux users can use the following kernel parameters to handle the possible issues:

- A. button.lid\_init\_state=method: When this option is specified, the ACPI button driver reports the initial lid state using the returning value of the \_LID control method and whether the "opened"/"closed" events are paired fully relies on the firmware implementation.

This option can be used to fix some platforms where the returning value of the \_LID control method is reliable but the initial lid state notification is missing.

This option is the default behavior during the period the userspace isn't ready to handle the buggy AML tables.

- B. button.lid\_init\_state=open: When this option is specified, the ACPI button driver always reports the initial lid state as "opened" and whether the "opened"/"closed" events are paired fully relies on the firmware implementation.

This may fix some platforms where the returning value of the \_LID control method is not reliable and the initial lid state notification is missing.

If the userspace has been prepared to ignore the unreliable "opened" events and the unreliable initial state notification, Linux users should always use the following kernel parameter:

- C. button.lid\_init\_state=ignore: When this option is specified, the ACPI button driver never reports the initial lid state and there

is a compensation mechanism implemented to ensure that the reliable "closed" notifications can always be delivered to the userspace by always pairing "closed" input events with complement "opened" input events. But there is still no guarantee that the "opened" notifications can be delivered to the userspace when the lid is actually opens given that some AML tables do not send "opened" notifications reliably.

In this mode, if everything is correctly implemented by the platform firmware, the old userspace programs should still work. Otherwise, the new userspace programs are required to work with the ACPI button driver. This option will be the default behavior after the userspace is ready to handle the buggy AML tables.