# Fixing Incorrect Lid State by Hacking DSDT

When I install a Linux distro to my VAIO notebook, I found that there is an annoying bug with the lid switch. It does not get updated whenever I suspend on lid close, it means `cat /proc/acpi/button/lid/LID/state` will output `state: close`. When I close the lid again, it won't suspend, instead, it will change the state to open. So in order for it to suspend again on lid close after the first suspend, I have to close it, reopen the lid and close it again.

I have tried installing Linux Mint, Fedora, Fuduntu and Xubuntu, but it is not fixed in any of the distros. So, I don't think it is distro problems. While researching this issues (which I spent two full days), I found that Linux got an amazing feature that enable users to dynamically loading DSDT at boot time, there is no need to update the BIOS. So here's the instuctions:

1. Install `iasl` using `yum`, `apt-get` or whatever package management you are using.
2. Extract DSDT:

   ```
   $ sudo cat/sys/firmware/acpi/tables/DSDT > dsdt.aml
   ```

3. Disassemble `dsdt.aml` using the following command, this should create a new file `dsdt.dsl`:

   ```
   $ iasl -d dsdt.aml
   ```

4. Compile it using:

   ```
   $ iasl -tc dsdt.dsl
   ```

5. Fix any compiler errors, warnings and remarks. On my machine, the output is:

   ```
   dsdt.dsl  1352:                         And (CTRL, 0x1E)
   Warning  1106 -                                 ^ Result is not used, operator has no
   effect

   dsdt.dsl  1584:                     0x00000000,         // Length
   Error    4122 -                              ^ Invalid combination of Length and
   Min/Max fixed flags

   dsdt.dsl  2443:                                 Name (_T_0, 0x00)
   ```

```
Remark   5111 -            Use of compiler reserved name ^  (_T_0)


dsdt.dsl  2521:                                 Name (_T_0, 0x00)
Remark   5111 -            Use of compiler reserved name ^  (_T_0)
```

a. The first one is on line 1352 can be fixed simply by changing `And (CTRL, 0x1E)` to `And (CTRL, 0x1E, CTRL)`.
b. The second one is on line 1584, the length should be `Range Maximum` - `Range Minimum` + 1, on my machine, so fire up a hex calculator and start subtracting. On my machine, it's `0xE0000000` (`0xDFFFFFFF` - `0x00000000` + `0x00000001`).
c. The third and fourth line is on line 2443 and 2521, because it uses a reserved name, simply replacing all instances of `_T_0` to `T_0` will stop the complaints. In vim, it is as simple as issuing `:%s/_T_0/T_0/g` in command mode.

6. Once everything is fixed (no errors, warning or remarks), add the following line to `_WAK` method, simply search for `_WAK` in `dsdt.dsl`:

```
If (LNotEqual (0x00, LIDS))
{
Store (0x00, LIDS)
Notify (\_SB.LID, 0x80)
}
```

**NOTE 1:** You might need to change `\_SB.LID` to match your path to `LID` method or on some machine `LID0`. Method name is preceded by an `_` (underscore), so you can search for `_LID` in `dsdt.dsl`. After you found it, you have to determine the scope, scroll up until you found `Scope` keyword that your `LID` or `LID0` method belongs to, inside the bracket is the scope name. It may be in more than one scope, so, it might be `\_PCI0.SB.LID`. If you specify an incorrect path to `LID` method, you will receive the following error:

```
dsdt.dsl 300:  Notify (LID, 0x80)
Error 4068 - ^ Object is not accessible from this scope (LID_)
```

**NOTE 2:** What this function does is just to update the lid state once it is resumed from sleep. According to the ACPICA documentation, `_WAK` method is called by `AcpiLeaveSleepState()` function of ACPI. If the lid is open, the `LIDS` variable is `0x00`, or `0x01` otherwise. So these few lines translate to "if lid state is not open (closed), change lid state to open and call `LID`method".

7. Compile it using `iasl -tc dsdt.dsl`.
8. If no errors, warnings or remarks, add the following lines to `/etc/grub.d/01_acpi`:

```
# Uncomment to load custom ACPI table
GRUB_CUSTOM_ACPI="/boot/dsdt.aml"


# DON'T MODIFY ANYTHING BELOW THIS LINE!
```

```
prefix=/usr
exec_prefix=${prefix}
libdir=${exec_prefix}/lib

. /usr/share/grub/grub-mkconfig_lib
#. ${libdir}/grub/grub-mkconfig_lib

# Load custom ACPI table
if [ x${GRUB_CUSTOM_ACPI} != x ] && [ -f ${GRUB_CUSTOM_ACPI} ] \
   && is_path_readable_by_grub ${GRUB_CUSTOM_ACPI}; then
   echo "Found custom ACPI table: ${GRUB_CUSTOM_ACPI}" >&2
   prepare_grub_to_access_device `${grub_probe} --target=device
${GRUB_CUSTOM_ACPI}` | sed -e "s/^/  /"
   cat << EOF
acpi (\$root)`make_system_path_relative_to_its_root
${GRUB_CUSTOM_ACPI}`
EOF
fi
```

9. Add executable bit to it:

```
$ sudo chmod +x /etc/grub.d/01_acpi
```

10. Copy the new `dsdt.aml` to `/boot`:

```
$ sudo cp dsdt.aml /boot
```

11. Regenerate `grub.cfg`:

```
$ sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

12. Reboot

## References

- Archwiki on DSDT
- Redhat's Bug Report
- Ubuntu's Bug Report 1
- Ubuntu's Bug Report 2
- Somebody's blog on fixing DSDT errors, remarks and warnings
- ACPICA Documentation

Revision #2
Created 5 April 2017 19:46:40 by Tingwai
Updated 2 July 2017 20:28:51 by Tingwai