

Linux ACPI Custom Control Method How To

Author: Zhang Rui <rui.zhang@intel.com>

Linux supports customizing ACPI control methods at runtime.

Users can use this to:

1. override an existing method which may not work correctly, or just for debugging purposes.
2. insert a completely new method in order to create a missing method such as `_OFF`, `_ON`, `_STA`, `_INI`, etc.

For these cases, it is far simpler to dynamically install a single control method rather than override the entire DSDT, because kernel rebuild/reboot is not needed and test result can be got in minutes.

Note

- Only ACPI METHOD can be overridden, any other object types like "Device", "OperationRegion", are not recognized. Methods declared inside scope operators are also not supported.
- The same ACPI control method can be overridden for many times, and it's always the latest one that used by Linux/kernel.
- To get the ACPI debug object output (Store (AAAA, Debug)), please run:

```
echo 1 > /sys/module/acpi/parameters/aml_debug_output
```

1. override an existing method

- a. get the ACPI table via ACPI sysfs I/F. e.g. to get the DSDT, just run `"cat /sys/firmware/acpi/tables/DSDT > /tmp/dsdt.dat"`
- b. disassemble the table by running `"iasl -d dsdt.dat"`.
- c. rewrite the ASL code of the method and save it in a new file,
- d. package the new file (psr.asl) to an ACPI table format. Here is an example of a customized `_SB.AC._PSR` method:

```
DefinitionBlock ("", "SSDT", 1, "", "", 0x20080715)
{
    Method (\_SB.AC._PSR, 0, NotSerialized)
    {
        Store ("In AC _PSR", Debug)
        Return (ACON)
    }
}
```

Note that the full pathname of the method in ACPI namespace should be used.

- e. assemble the file to generate the AML code of the method. e.g. `"iasl -vw 6084 psr.asl"` (psr.aml is generated as a result) If parameter `"-vw 6084"` is not supported by your iASL compiler, please try a newer version.
- f. mount debugfs by `"mount -t debugfs none /sys/kernel/debug"`
- g. override the old method via the debugfs by running `"cat /tmp/psr.aml > /sys/kernel/debug/acpi/custom_method"`

2. insert a new method

This is easier than overriding an existing method. We just need to create the ASL code of the method we want to insert and then follow the step c) ~ g) in section 1.

3. undo your changes

The "undo" operation is not supported for a new inserted method right now, i.e. we can not remove a method currently. For an overridden method, in order to undo your changes, please save a copy of the method original ASL code in step c) section 1, and redo step c) ~ g) to override the method with the original one.

Note

We can use a kernel with multiple custom ACPI method running. But each individual write to debugfs can implement a SINGLE method override. i.e. if we want to insert/override multiple ACPI methods, we need to redo step c) ~ g) for multiple times.

Note

Be aware that root can mis-use this driver to modify arbitrary memory and gain additional rights, if root's privileges got restricted (for example if root is not allowed to load additional modules after boot).