

RSDP

From OSDev Wiki

RSDP (Root System Description Pointer) is a data structure used in the ACPI programming interface.

In ACPI Version 1.0 it has this structure:

```
struct RSDPDescriptor {
    char Signature[8];
    uint8_t Checksum;
    char OEMID[6];
    uint8_t Revision;
    uint32_t RsdAddress;
}
```

since Version 2.0 it has been extended, and the following new fields have been added:

```
struct RSDPDescriptor20 {
    RSDPDescriptor firstPart;

    uint32_t Length;
    uint64_t XsdtAddress;
    uint8_t ExtendedChecksum;
    uint8_t reserved[3];
};
```

Contents

- 1 Detecting the RSDP
- 2 Validating the RSDP
 - 2.1 Detecting ACPI Version
 - 2.2 Checksum validation
- 3 Explaining the fields
 - 3.1 Always present
 - 3.1.1 Signature
 - 3.1.2 Checksum
 - 3.1.3 OEMID
 - 3.1.4 Revision
 - 3.1.5 Rsd Address
 - 3.2 Since Version 2.0
 - 3.2.1 Length
 - 3.2.2 Xsdt Address
 - 3.2.3 Extended Checksum
 - 3.2.4 Reserved

- 4 What's next?

Detecting the RSDP

The RSDP is either located within the first 1 KB of the EBDA (Extended BIOS Data Area) (a 2 byte real mode segment pointer to it is located at 0x040E), or in the memory region from 0x000E0000 to 0x000FFFFF (the main BIOS area below 1 MB). To find the table, the Operating System has to find the "RSD PTR " string (notice the last space character) in one of the two areas. This signature is always on a 16 byte boundary.

Validating the RSDP

Once you have found the RSDP Table you have to see what version of ACPI the BIOS is using, then you must check that the checksum is valid.

Detecting ACPI Version

The ACPI Version can be detected using the Revision field in the RSDP. If this field contains 0, then ACPI Version 1.0 is used. 1 stands for ACPI Version 2.0, 2 for ACPI Version 3.0, and so on...

Checksum validation

Before the RSDP is relied upon you should check that the checksum is valid. For ACPI 1.0 (the first structure) you add up every byte in the structure and make sure the lowest byte of the result is equal to zero. For ACPI 2.0 and later you'd do exactly the same thing for the original (ACPI 1.0) part of the second structure, and then do it again for the fields that are part of the ACPI 2.0 extension.

Explaining the fields

Always present

Signature

This 8-byte string (not null terminated!) must contain "RSD PTR ". It stands on a 16-byte boundary.

Checksum

The value to add to all the other bytes (of the Version 1.0 table) to calculate the Checksum of the table. If this value added to all the others and casted to byte isn't equal to 0, the table must be ignored.

OEMID

The specification says: "An OEM-supplied string that identifies the OEM"

Revision

The revision of the ACPI. Larger revision numbers are backward compatible to lower revision numbers. See Detecting ACPI Version for further information.

RsdT Address

32-bit physical (I repeat: physical) address of the RSDT table.

Since Version 2.0

Length

The size of the entire table since offset 0 to the end.

Xsdt Address

64-bit physical address of the XSDT table. If you detect ACPI Version 2.0 you should use this table instead of RSDT even on x86, casting the address to `uint32_t`.

Extended Checksum

This field is used to calculate the checksum of the entire table, including both checksum fields

Reserved

3 bytes to be ignored in reading and that must not be written

What's next?

Well, you should now parse the RSDT (or XSDT)

Retrieved from "<http://wiki.osdev.org/index.php?title=RSDP&oldid=17406>"

Category: ACPI

- This page was last modified on 1 January 2015, at 12:54.
- This page has been accessed 24,531 times.