TCG ACPI Specification

Family “1.2” and “2.0”

Level 00 Revision 00.37

December 19, 2014

Contact: admin@trustedcomputinggroup.org

TCG Published

TCG

Disclaimers, Notices, and License Terms

Acknowledgements

The writing of a specification, particularly a security specification, takes many hours for both development and review. This specification is no exception with roughly 50 individuals involved in the process. The TCG would like to acknowledge the contribution of those individuals (listed below) and the companies who allowed them to volunteer their time to the development of this specification.

Special thanks are due to Carey Huscroft, who served as Chair of the Server Working Group during the development of this specification.

The TCG would also like to give special thanks to Ronald Aigner who was the editors of this specification.

Contributors:

Emily Ratliff, AMD
Gary Simpson, AMD
John Mersh, ARM Ltd.
Ronnie Thomas, Atmel
Günter Fuchs, Atmel
Chris Berg, Atmel
Bill Jacobs, Cisco
Amy Nelson, Dell
Andreas Fuchs, Fraunhofer Institute for Secure Information Technology
Carey Huscroft, Hewlett Packard
Russ Herrell, Hewlett Packard
Thomas Ford, Hewlett Packard
Lan Wang, Hewlett Packard
Tim Block, IBM
Monty Wiseman, Intel
Lee Rosenbaum, Intel
Robert Hart, Johns Hopkins University, Applied Physics Lab
Shiva Dasari, Lenovo
Randy Springfield, Lenovo
Ronald Aigner, Microsoft
Paul England, Microsoft
Eugene Samsonov, Microsoft
Dan Morav, Nuvoton
Jae-Woo Park, Oracle
Dick Wilkins, Phoenix
Tom Brostrom, United States Government

Contents

List of Tables

1 Scope

This specification defines the framework of necessary ACPI tables and basic methods to be used on a TCG compliant platform. The table and ACPI namespace objects provide enough information to the operating system to enable access to the TCG compliant hardware in a platform.

2 Introduction

The intention of this specification is to provide a framework for all platform types that employ ACPI. These platform types can draw from this specification as necessary, and may further specify ACPI functionality that is required for a particular platform type.

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1
buffer

data structure used for transport to and from the TPM

3.2
CLEAR

bit with a value of zero (0), or the action of causing a bit to have a value of zero (0)

3.3
command

the values sent to the TPM to indicate the operation to be performed

3.4
octet

eight bits of data

Note: On most modern computers, this is the smallest addressable unit of data.

3.5
response

values returned by the TPM when it completes processing of a command

3.6
SET

bit with a value of one (1), or the action of causing a bit to have a value of one (1)

3.7
TPM Device Reset

resetting of TPM internal state due to _TPM_Init

3.8
Trusted Platform Module
TPM

implementation compliant with the *TPM Main Specification; Family 1.2* or *TPM Library Specification; Family 2.0; Level 00; Revision 99* or later.

4 Abbreviations

For the purposes of this document, the following abbreviations apply.

Abbreviation	Description
_CID	Compatibility ID
_HID	Hardware ID
TPM	Prefix for an indication passed from the system interface of the TPM to a Protected Capability defined in a TPM specification
ACPI	Advanced Configuration and Power Interface ([ACPI 5.0] refers to version 5.0 of the ACPI specification at http://uefi.org/acpi/specs)
ASL	ACPI Source Language
APIC	Advanced Programmable Interrupt Controller
CPU	Central Processing Unit
DSDT	Differentiated System Description Table
DWORD	Double word, a 32-bit number
FADT	Fixed ACPI Description Table (See [ACPI 5.0] for the definition)
GAS	Generic Address Structure (See [ACPI 5.0] for the definition)
GPE	ACPI General Purpose Event register block (See [ACPI 5.0] for definition)
GPEx_STS	ACPI GPE Status register (part of the GPE register block specific for the event denoted by ‘x’ (0 or 1)
ID	Identifier
I/O	Input/Output
LAML	Log Area Minimum Length
LASA	Log Area Start Address
MMIO	Memory Mapped IO
MSO	Most Significant Octet
NO-OP	No operation. A CPU operation that does nothing.
OEM	Original Equipment Manufacturer
OS	Operating System
OSPM	Operating System Power Management
PCI	Peripheral Component Interconnect
PNP	Plug and Play
SAPIC	Streamlined APIC
SCI	System Control Interrupt
TCG	Trusted Computing Group
TPM	Trusted Platform Module
TPM2_	Prefix for a command defined in the TPM 2.0 Library specification
QWORD	Quad Word, a 64-bit number
WORD	Word, a 16-bit number

5 Compliance

Systems providing an ACPI table for TPM 1.2 or TPM 2.0 have to adhere to the layout of the tables described in this specification. A TCG compliant platform that implements ACPI SHALL provide the ACPI tables appropriate for the platform type.

6 Conventions

Text in this specification on a white background is normative. Informative text is formatted with a grey background.

6.1 Bit and Octet Numbering and Order

An integer value is considered to be an array of one or more octets. The octet at offset zero within the array is the most significant octet (MSO) of the integer. Bit number 0 of that integer is its least significant bit in the last octet in the array.

Example: A 32-bit integer is an array of four octets; the MSO is at offset [0], and the most significant bit is bit number 31. Bit zero of this 32-bit integer is the least significant bit in the octet at offset [3] in the array.

Note: Array indexing is zero-based.

Note: This definition does not match the "network bit order" used in many IETF documents, such as RFC 4034. In those documents, the most significant bit of a datum has the lowest bit number. It is conventional practice to send that bit first when using a serial network protocol, and the bits are numbered in the order in which they are sent. This specification numbers bits according to the power of two to which they correspond within a datum. This numbering corresponds to the normal convention for bit numbering in hardware registers that hold integer values rather than fixed-point numbers.

The first listed member of a structure is at the lowest offset within the structure and the last listed member is at the highest offset within the structure.

For a character string (letters delimited by ""), the first character of the string contains the MSO.

6.2 Numbers

Numbers are decimal unless a different radix is indicated.

Unless the number appears in a table intended to be machine readable, the radix is a subscript following the digits of the number. Only radix values of 2 and 16 are used in this specification.

Radix 16 (hexadecimal) numbers have a space separator between groups of two hexadecimal digits.

Example: $40\ FF\ 12\ 34_{16}$

Radix 2 (binary) numbers use a space separator between groups of four binary digits.

Example: $0100\ 1110\ 0001_{2}$

For numbers using a binary radix, the number of digits indicates the number of bits in the representation.

Example: 20_{16} is a hexadecimal number that contains exactly 8 bits and has a decimal value of 32.

Example: $10\ 0000_2$ is a binary number that contains exactly 6 bits and has a decimal value of 32.

Example: $0\ 20_{16}$ is a hexadecimal number that contains exactly 12 bits and has a decimal value of 32.

A number in a machine-readable table may use the "0x" prefix to denote a base 16 number. In this format, the number of digits is not always indicative of the number of bits in the representation.

Example: 0x20 is a hexadecimal number with a value of 32, and the number of bits is determined by the context.

7 ACPI Table

All TCG platforms supporting ACPI utilize the same header section layout, which is separated from the rest of the table with a double line. Table 2 describes the client ACPI table for TPM 1.2. Table 4 describes the server ACPI table for TPM 1.2. Table 7 describes the ACPI table for TPM 2.0, which can be used for client or server platforms.

The value in the signature field for the TPM 2.0 table ('TPM2') differs from the value for the TPM 1.2 tables ('TCPA').

ACPI tables have the little-endian byte format defined in the ACPI specification.

7.1 Client ACPI Table for TPM 1.2

7.1.1 Client Common Header Values

These are the specific values to be used in the client platform version of the ACPI table for TPM 1.2.

Table 1: Defined values for Client ACPI Table for TPM 1.2

Field	Value	Description
Length	50	Size of the table
Revision	2	Revision for PC Client Platform Class
Platform Class	0	PC Client Platform Class

7.1.2 ACPI Table Layout

Table 2: TCG Hardware Interface Description Table Format for TPM 1.2 Clients

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TCPA'. Signature for the TCG Hardware Interface Table.
Length	4	4	See section 7.1.1. The length of this table starting from the Signature field up to and including the LASA field. It does not include the size of the area storing events or other data that is referenced or pointed to by any of these fields.
Revision	1	8	See section 7.1.1. Revision of this table including the data and structures reference by it. E.g., if the event structures with the area reference by LASA change, this revision shall be incremented. **Note*:*** The purview of this revision is within platform class as indicated by the Platform Class field. This means that each platform class increments this field autonomously. Software referencing this table should interpret the Platform Class field prior to interpreting this Revision field.
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID. Per ACPI specification. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by "OEM ID").
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI, this is "An OEM-supplied revision number. Larger numbers are assumed to be newer revisions."

Field	Byte Length	Byte Offset	Description
Creator ID	4	28	Vendor ID of utility that created the table. For the tables containing Definition Blocks, this is the ID for the ASL Compiler.
Creator Revision	4	32	Revision of utility that created the table. For the tables containing Definition Blocks, this is the revision for the ASL Compiler.
Platform Class	2	36	See section 7.1.1.
Log Area Minimum Length (LAML)	4	38	Identifies the minimum length (in bytes) of the system's pre-boot TCG event log area. **Note**: For PC Client Implementation Specification up to and including 1.2 the minimum log size is 64KB.
Log Area Start Address (LASA)	8	42	Contains the 64-bit physical address of the start of the system's pre-boot TCG event log area, in QWORD format. **Note**: The log area ranges from address LASA to LASA+(LAML-1).

7.2 Server ACPI Table for TPM 1.2

7.2.1 Server Common Header Values

These are the specific values to be used in the server platform version of the ACPI table for TPM 1.2.

Table 3: Defined values for Server ACPI table for TPM 1.2

Field	Value	Description
Length	100	Size of the table
Revision	2	Revision for Server Platform Class
Platform Class	1	Server Platform Class

7.2.2 ACPI Table Layout

Table 4: TCG Hardware Interface Description Table Format for TPM 1.2 Servers

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TCPA'. Signature for the TCG Hardware Interface Table.
Length	4	4	See Section 7.2.1. The length of this table starting from the Signature field up to and including the PCI Function Number field. It does not include the size of the area storing events or other data that is referenced or pointed to by any of these fields.
Revision	1	8	See Section 7.2.1. Revision of this table including the data and structures reference by it. E.g., if the event structures with the area reference by LASA change, this revision shall be incremented. ***Note:*** The purview of this revision is within platform class as indicated by the Platform Class field. This means that each platform class increments this field autonomously. Software referencing this table should interpret the Platform Class field prior to interpreting this Revision field.
Checksum	1	9	Entire table must sum to zero.

Field	Byte Length	Byte Offset	Description
OEMID	6	10	OEM ID. Per ACPI specification. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by "OEM ID").
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI, this is "An OEM-supplied revision number. Larger numbers are assumed to be newer revisions."
Creator ID	4	28	Vendor ID of utility that created the table. For the tables containing Definition Blocks, this is the ID for the ASL Compiler.
Creator Revision	4	32	Revision of utility that created the table. For the tables containing Definition Blocks, this is the revision for the ASL Compiler.
Platform Class	2	36	See Section 7.2.1.
Reserved	2	38	0 **Note:** This creates natural alignment for the fields that follow.
Log Area Minimum Length (LAML)	8	40	Identifies the minimum length (in bytes) of the system's pre-boot TCG event log area.
Log Area Start Address (LASA)	8	48	Contains the 64-bit physical address of the start of the system's pre-boot TCG event log area, in QWORD format. **Note**: The log area ranges from address LASA to LASA+(LAML-1).
Specification Revision	2	56	Identifies the TCG specification revision, in BCD format, to which the interface was designed. The first byte holds the most significant digits, while second byte holds the least significant digits of the revision, e.g. a value of 01 02_{16} indicates the interface is compatible with TCG specification v1.2.
Device Flags	1	58	See Table 5
Interrupt Flags	1	59	See Table 6
GPE	1	60	The bit assignment of the SCI within the GPEx_STS register of a GPE described in the FADT that the interface triggers. **Note*:*** This field is valid only if Bit[2] of the Interrupt Flags field is set.)
Reserved	3	61	0
Global System Interrupt	4	64	The I/O APIC or I/O SAPIC Global System Interrupt used by the interface. **Note*:*** This field is valid only if Bit[3] of the Interrupt Flags field is set.
Base Address	12	68	The base address of the hardware register set described using the GAS. The Address_Space_ID field in the GAS can only be of the value of 0 (System Memory) and 1 (System IO). All other values are not permitted. This address must be the Host Side address in the case of MMIO, and it must be the Host Side IO port address in the case of IO Port.
Reserved	4	80	Set to 0. This is to naturally align the data fields that follow.
Configuration Address	12	84	The configuration address of the TPM hardware device described using the GAS. The Address_Space_ID field in the GAS can only be of the value of 0 (System Memory) and 1 (System IO). All other values are not permitted. This is only valid if Bit[2] of the Device Flags field is set. This address must be the Host Side address in the case of MMIO, and it must be the Host Side IO port address in the case of IO Port.

Field	Byte Length	Byte Offset	Description
PCI Segment Group Number	1	96	PCI Segment Group Number, if the TPM device is a PCI device
PCI Bus Number	1	97	PCI Bus Number, if the TPM device is a PCI device
PCI Device Number	1	98	Bit 4:0 – PCI Device Number: The PCI device number if the TPM device is a PCI device. Bit 7:5 – Reserved
PCI Function Number	1	99	Bit 2:0 – PCI Function Number: The PCI function number if the TPM device is a PCI device. Bit 7:3 – Reserved

7.2.3 Device Flags

Table 5: Bit layout of Device Flags

Bit	Description
7..3	Reserved
2	TPM configuration address valid 0 = TPM configuration address is invalid 1 = TPM configuration address is valid
1	TPM Bus is PNP 0 = FALSE (the TPM address and interrupt must not be changed) 1 = TRUE (the TPM address and interrupt may be changed by PNP OS code)
0	PCI Device Flag. For PCI TCG devices, this bit is set. 0 = non-PCI device, the PCI Segment Group, Bus, Device and Function Number fields combined corresponds to the ACPI _UID value of the device whose _HID or _CID contains a TPM plug and play ID. 1 = PCI Device

7.2.4 Interrupt Flags

Table 6: Bit layout of Interrupt Flags

Bit	Description
7..4	Reserved
3	I/O APIC/SAPIC interrupt (Global System Interrupt) 0 = not supported 1 = supported
2	SCI triggered through GPE 0 = not supported 1 = supported
1	Interrupt Polarity, 0 = Active-High: This interrupt is sampled when the signal is high, or true. 1 = Active-Low: This interrupt is sampled when the signal is low, or false. **Note:** PCI devices are always active low, so this bit is set to 1 for PCI devices.
0	Interrupt Mode, 0 = Level-Triggered: This interrupt is triggered in response to the signal being in either a high or low state. 1 = Edge-Triggered: This interrupt is triggered in response to a change in signalstate, either high to low or low to high. **Note:** PCI devices are always level triggered, so this bit is set to 0 for PCI devices.

7.3 ACPI Table for TPM 2.0

This is the definition for the ACPI table for TPM 2.0.

Table 7: TCG Hardware Interface Description Table Format for TPM 2.0

Field	Byte Length	Byte Offset	Description
Header			
Signature	4	0	'TPM2'. Signature for the TCG Hardware Interface Table.
Length	4	4	The length of this table starting from the Signature field up to and including the platform specific parameters field. (52 + size of platform specific parameters)
Revision	1	8	4. The current revision of this table.
Checksum	1	9	Entire table must sum to zero.
OEMID	6	10	OEM ID. Per ACPI specification. An OEM-supplied string that identifies the OEM.
OEM Table ID	8	16	For the TPM Interface Table, the table ID is the manufacturer model ID (assigned by the OEM identified by "OEM ID").
OEM Revision	4	24	OEM revision of TPM Interface Table for the given OEM Table ID. Per ACPI, this is "An OEM-supplied revision number. Larger numbers are assumed to be newer revisions."
Creator ID	4	28	Vendor ID of utility that created the table. For the tables containing Definition Blocks, this is the ID for the ASL Compiler.
Creator Revision	4	32	Revision of utility that created the table. For the tables containing Definition Blocks, this is the revision for the ASL Compiler.
Platform Class	2	36	0 for client platforms. 1 for server platforms.
Reserved	2	38	0
Address of Control Area	8	40	Physical address of Control Area. The Control Area contains status registers and the location of the memory buffers for communicating with the device. The area may be in either TPM 2.0 device memory or in memory reserved by the system during boot. Interfaces that do not require the Control Area set this value to zero.
Start Method	4	48	The Start Method selector determines which mechanism the device driver uses to notify the TPM 2.0 device that a command is available for processing. This field may contain one of the values specified in Table 8.
Platform Specific Parameters	Variable	52	The content of the platform specific parameters is determined by the Start Method used by the system's TPM device interface. This field contains values that may be used to initiate command processing. This information may be vendor specific. If the Start Method value is 2 then this field is four bytes in size and must be all zero.

Table 8: Start Method values for ACPI table for TPM 2.0

Value	Description
0	Not allowed (indicates value has not been set).
1	Reserved for legacy use (vendor specific).
2	Uses the ACPI Start method.
3 – 5	Reserved for legacy use (vendor specific).
6	Reserved for the Memory mapped I/O Interface (TIS 1.2+Cancel).
7	Uses the Command Response Buffer Interface.
8	Uses the Command Response Buffer Interface with the ACPI Start Method.
9+	Reserved for future use

8 ACPI Device

Devices are not required to be exposed in ACPI namespace if the device exists on a bus that is enumerable by the Operating System, or on a bus which is plug-n-play capable.

A TCG platform class-specific ACPI Table may provide a mechanism that can be used before the ability to execute ACPI control methods in the OS is available. The Server ACPI Table in section 2.2 of this specification is one example of such a table. This table is not, however, intrinsically supported in the OSPM as a way of discovering and reporting system resources. Therefore, it is recommended that non-PCI TCG Hardware on the platform be described in the ACPI name space. This makes it possible for the OSPM to enumerate the TCG Hardware as a device. In addition, the ACPI name space description is more flexible and friendly in hot-plug scenarios.

Note that to be ACPI compatible, the fixed resources for TCG Hardware must still be accounted for in accordance with the ACPI specification. If the device is not formally described in the ACPI Name Space, its resources must be described as fixed system resources or the resources appended to some other fixed resource system device in order to ensure that the OSPM does not attempt to allocate those resources to some other device.

To formally describe the TCG Hardware System Interface in ACPI Name Space, a TCG hardware device is created using the named device object. Table 9 is a non-exhaustive list of ACPI control methods that may be used in a TCG hardware device object, along with a recommended support level for each method.

A TCG platform may provide an ACPI device object representing the TPM in the ACPI namespace, if the bus where the TPM is located is not PNP capable or the bus is not exposed to the OS for PNP operations.

It is recommended that TPM 2.0 device object ACPI table appears under the DSDT table in the ACPI namespace. The TPM 2.0 device object should be located under the system bus.

Table 9: TCG Hardware Device Object Control Methods

Object	Description	Support Level
_ADR	Named object that evaluates to the interface's address on its parent bus. _ADR is a standard device configuration control method defined in the ACPI Specification.	Required only for devices on a bus that has standard enumeration mechanism.
_HID	Named object that provides the interface's Plug and Play identifier. This value may be TPM vendor specific. _HID is a standard device configuration control method defined in the ACPI Specification.	Required only for devices that do not have standard enumeration mechanism.
_STR	Named object that evaluates to a Unicode string that may be used by an OS to provide information to an end user describing the device. __STR is a standard device configuration control method defined in the ACPI Specification.	Optional
_UID	Named object that specifies a device's unique persistent ID, or a control method that generates it. _UID is a standard device configuration control method defined in the ACPI Specification.	Optional
_CRS	Named object that returns the TPM interface's current resource settings. Security hardware Interfaces are considered static resources; hence only return their defined resources. The address region definition is interface type/subtype dependent. _CRS is a standard device configuration control method defined in the ACPI Specification.	Required
_STA	Object that returns the status of the device: enabled, disabled or removed, as defined in the ACPI Specification. If this method is not present, the device is assumed to be enabled.	Recommended
_DSM	Device Specific Method Function 0 – standard query function Function 1 – TCG Hardware Information Arguments: Arg0 (Buffer): UUID - {CF8E16A5-C1E8-4e25-B712-4F54A96702C8} Arg1 (Integer): Revision ID = 1 Arg2 (Integer): Function Index = 1 Arg3 (Package): Arguments = empty package Returns: ACPI Buffer type; the definition of the return a package of 2 items and the description is as follows. Package item 1: Type: Integer Purpose: status of operation Description: 0: Failure 1: Success Package item 2: Type: Package Purpose: TCG Revision implemented in security hardware Description: A package of 2 integers: Integer 1: (BCD format) – most significant digits of TCG version Integer 2: (BCD format) – least significant digits of TCG version For example: a value of 0x0110 indicates the interface is compatible with TCG specification v1.1.	Optional
_GPE	Named object that evaluates to either an integer or a package. If _GPE evaluates to an integer, the value is the bit assignment of the SCI within the GPEx_STS register of a GPE block described in the FADT that the Security hardware device will trigger.	Required if interrupt through GPE is supported

Object	Description	Support Level
	If _GPE evaluates to a package, then that package contains two elements. The first is an object reference to the GPE Block device that contains the GPE register that will be triggered by the interface. The second element is numeric (integer) that specifies the bit assignment of the SCI within the GPEx_STS register of the GPE Block device referenced by the first element in the package. **Note:** This object is only provided if the interface supports a GPE.	

8.1 Optional Start Method for TPM 2.0 devices

Some platforms may implement an optional ACPI Start Method to permit the OS to request the firmware to execute or cancel a TPM 2.0 command. The use of the ACPI Start method is determined by the StartMethod field of the static TPM2 ACPI table (see Section 7.2.3.) If the StartMethod field of the static ACPI table indicates the use of this method, the ACPI Start Method shall be implemented. The ACPI functions defined herein shall reside in the _DSM control method object.

The _DSM method defined herein shall be implemented as follows:

Function 0 – standard query function

Function 1 – ACPI Start Method

Arguments:

Arg0 (Buffer): UUID = 6bbf6cab-5463-4714-b7cd-f0203c0368d4

Arg1 (Integer): Revision ID = 0

Arg2 (Integer): Function Index = 1

Arg3 (Package): Arguments = Empty Package

Return Value:

Type: Integer

Description of the Return Value:

0: Success

1: General Failure

Functional Behavior:

This function tells the system to review the fields in the TPM 2.0 device control area and take the appropriate action, such as, but not limited to execute or cancel a TPM 2.0 command.

The function is non-blocking. The call will return immediately. When Return Value 0 is returned, the TPM inspected the control area and will take appropriate action. For instance, if a command has been submitted and *Start* is SET, the command was accepted and will be executed by the TPM. Whenever possible the system should return a TPM response instead of a General Failure from this call. For example, if a command cannot be processed when the method is called, a Response buffer of TPM2_RC_Retry could be written to the TPM's response area and the *Start* field in the Control Area could be CLEARed. If the command was cancelled because the *Cancel* field was set, this method may write a TPM2_RC_Cancelled return code in the Response buffer, clear the *Start* field in the Control Area and return a value of 0. Alternatively, if the *Cancel* field

is set the method may return a value of 0 and the TPM 2.0 device may later complete or cancel the command per the requirements for cancelling a command.

When 1 is returned, the firmware is unable to read or act upon the request. Other than the return value, the request is equivalent to a NO-OP. Examples are (a) a bad OS driver requests execution of additional commands before a prior command completed, (b) the control area is not in physical memory, or (c) the command or response physical addresses do not exist. A return value of 1 may cause software to stop using the TPM 2.0 device until the next full system boot (this excludes hibernation/resume cycles).