

Hierarchical Data Extension UUID For _DSD

October 2016

Revision 1.1

The material contained herein is not a license, either expressly or impliedly, to any intellectual property owned or controlled by any of the authors or developers of this material or to any contribution thereto. The material contained herein is provided on an "AS IS" basis and, to the maximum extent permitted by applicable law, this information is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses and of lack of negligence, all with regard to this material and any contribution thereto. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." The Unified EFI Forum, Inc. reserves any features or instructions so marked for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. ALSO, THERE

IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION AND ANY CONTRIBUTION THERETO.

IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR ANY CONTRIBUTION THERETO BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Copyright 2016 Unified EFI, Inc. All Rights Reserved.

Contents

1 Introduction	5
1.1 Terms	5
1.2 Conventions used in this document	5
1.2.1 Typographic conventions	5
2 Hierarchical Data Extension UUID	6
2.1 Data Format Definition	6
2.2 Example	6

Revision history

Revision Number	Description	Revision Date
<1.0>	First revision	August 4, 2015
<1.1>	Allow child node to be encoded as a Package reference as well as a String. Add restrictions on use of Strings and References. Add limitations on how Strings and References can be used together within a parent _DSD Hierarchy, updated ASL example to match.	October 20, 2016

1 Introduction

This document specifies the data format associated with UUID

dbb8e3e6-5886-4ba6-8795-1319f52a966b

(Hierarchical Data Extension UUID) for the `_DSD` (Device Specific Data) ACPI device configuration object.

1.1 Terms

The following terms are used throughout this document to describe varying aspects of input localization:

ACPI

Advanced Configuration and Power Interface specification.

Device

Hardware component or set of interrelated hardware registers.

GUID

Globally Unique Identifier. A 128-bit value used to uniquely name entities. A unique GUID can be generated by an individual without the help of a centralized authority. This allows the generation of names that will never conflict, even among multiple, unrelated parties.

UUID

Universal Unique Identifier, GUID.

1.2 Conventions used in this document

1.2.1 Typographic conventions

This document uses the typographic and illustrative conventions described below:

Plain text The normal text typeface is used for the vast majority of the descriptive text in a specification.

BOLD Monospace Computer code, example code segments, and all prototype code segments use a **BOLD Monospace** typeface with a dark red color. These code listings normally appear in one or more separate paragraphs, though words or segments can also be embedded in a normal text paragraph.

2 Hierarchical Data Extension UUID

2.1 Data Format Definition

The hierarchical Data Extension UUID

dbb8e3e6-5886-4ba6-8795-1319f52a966b

defines the data format for the Package (Data Structure) immediately following it as a list of Packages of length 2 (Sub-node Links) where the first element of each contained Package (Key) must be a String and the second element of it (Target) must be either a String encoding the name of the referenced ACPI object or a reference to the ACPI object. That name can be a fully qualified path, a relative path, or a simple name segment utilizing the ACPI namespace search rules as defined by the ACPI specification (ACPI 6.0, Section 5.3 "ACPI Namespace", Section 19.2.2 "ASL Name and Pathname Terms" and Section 19.3.2.2 "Strings").

Moreover, the ACPI object pointed to by Target (the Target Object) must evaluate to a Package formatted in accordance with the `_DSD` return value format defined by the ACPI specification (ACPI 6.0, Section 6.2.5). Also, like `_DSD`, it must return the same data every time it is evaluated and the meaning of those data is the same as for analogous data returned by `_DSD`.

The Key of each Sub-node Link must be unique within the enclosing Data Structure. That is, it is invalid to put two Sub-node Links with identical Keys into one enclosing Package.

This allows hierarchical device configuration information to be represented as a hierarchy of ACPI objects returning Packages following the `_DSD` data Package formatting rules. Then, each of those objects may be regarded as a Data-only Subnode of the Device object holding the `_DSD` at the top of the hierarchy.

For maximum interoperability, it is recommended to put all Target Objects referenced as Strings into the same scope in which the objects that return data including their names are located. If using object references, the resolution of the referenced object is managed by the AML interpreter, and there is no limitation placed on the location of the referenced object.

If the Target of any property within a Package is a Reference, then all Targets within and beneath the Package must also be References, not Strings. Implementers are encouraged to use exclusively strings or references throughout the hierarchy.

2.2 Example

The following example illustrates the possible use of the Hierarchical Data Extension UUID and Sub-node Links. It contains a definition of a master Device (SWC0), three Data-only Sub-nodes (DP0P, DPNP, DP00) and one child Device object (SWD0) under it. In addition, the Data-only Sub-node DP00 is a Sub-node of DPNP (which is a direct

Sub-node of the master Device). Additionally, a common set of properties (COMN) is provided and referenced by DP0P, DPNP, and DP00. This allows the firmware developer to ensure that information common to multiple devices is identical.

```

Device(SWC0) {
  Name(_HID, "VEND0000") // sample Vendor ID - do not use
  Name(_DSD, Package() {
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {...}, // Property 1
      ...
      Package (2) {...}, // Property n
    },
    ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
    Package () {
      Package (2) {"Alice", "DP0P"} // String example
      Package (2) {"Frank", "DPNP"} // String example
    }
  })

  Name(DP0P, Package(){ // Data-only subnode of SWC0
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {...}, // Property 1
      ...
      Package (2) {...} // Property n
    },
    ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
    Package () {
      Package (2) {"common-properties", ^COMN} // Reference
    }
  })

  Name(DPNP, Package(){ // Data-only subnode of SWC0
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {...}, // Property 1
      Package (2) {...} // Property 2
    },
    ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
    Package () {
      Package (2) {"child-of-Frank", "DP00"},
      Package (2) {"common-properties", ^COMN} // Reference
    }
  })

  Name(DP00, Package(){ // Data-only subnode of DPNP
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {...}, // Property 1
      ...
      Package (2) {...} // Property n
    },
    ToUUID("dbb8e3e6-5886-4ba6-8795-1319f52a966b"),
    Package () {
      Package (2) {"common-properties", ^COMN} // Reference
    }
  })

```

```
    }
  })

Device (SWD0) {
  Name(_ADR, ...)
  Name(_DSD, Package() {
    ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
    Package () {
      Package (2) {...}, // Property 1
      ...
      Package (2) {...} // Property n
    }
  })
} // End SWD0

Name(COMN, Package() { // Common properties
  ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
  Package () {
    Package (2) {...}, // Property 1
    ...
    Package (2) {...} // Property n
  }
} // End COMN

} // End SWC0
```