ACPI _DSD DEVICE PROPERTY REGISTRY
==================================

This is an industry registry of device property bindings to be used in the configuration of devices with ACPI, and in particular when using the _DSD object as defined in the ACPI specifcation [link to spec] starting with version 5.1. In that version of the specification, _DSD in general is defined. This registry is only for device properties used with the _DSD data format associated with UUID daffd814-6eba-4d8c-8a91-bc9bbf4aa301 [link to doc defining this _DSD UUID], the "Device Properties UUID."

The Device Properties UUID defines the data format for a _DSD Package (i.e., a Data Structure) containing a list of Packages of length two (2) (which will be referred to as "Properties"). The first element of each Package in the list (that is, a Key) must be a String and the second element of it (the Value) must be one of:

- an Integer,
- a String,
- a Reference, or
- a Package consisting entirely of Integer, String, or Reference objects (but _not_ nested Packages).

The list of valid Keys, and the format and interpretation of the corresponding Values, depends on the PNP or ACPI device ID (e.g., _HID) of the Device containing the _DSD. For instance, the PNP device ID returned by _HID for the example Device object MDEV below will determine the list of valid Keys and the corresponding Value data formats for that Device object's _DSD.

```
   Device (MDEV) {
      Name (_HID, "PNP####")
      Name (_DSD, Package () {
         ToUUID("daffd814-6eba-4d8c-8a91-bc9bbf4aa301"),
         Package () {
            Package (2) {"length", 16},    // Property 1
            Package (2) {"speed", 100},    // Property 2
            ...
            Package (2) {...}     // Property n
         }
      })
      ...
   }
```

In the context of the Device Properties UUID each Property is a characteristic of the hardware itself, or the way it is used in the system in which it is incorporated, as opposed to software configuration data. As such, the list of valid property Keys and Value data formats associated with them must be defined by the device vendor in a way that is independent of the firmware interface to be used on any given platform.

Since multiple Properties with the same Key in a single Data Structure associated with the Device Properties UUID are not permitted, this registry provides a mechanism for making Keys and their associated Values known. The intent is to encourage the re-use of existing Properties, and the de facto standardization of Properties so that a single set of ACPI tables may be successfully used with multiple operating systems. The registry aims to ensure fair, orderly, consistent and conflict free naming of the Properties, their Keys and Values.

TO REGISTER A NEW DEVICE PROPERTY

Send a request to the ACPI Specification Workgroup (ASWG) Chairperson [link to email] specifying the Property to be registered. All requests are subject to ASWG approval based on the objectives noted above.

The list of currently registered Properties is here [link to current registry].

Properties are subject to uniqueness requirements and some requests may not be available.

Properties also fall into two groups:

1) Properties that apply to a class of devices, such as network interface cards, or SATA devices.

2) Properties that apply to a unique subset within a class, typically a device specific to a particular hardware vendor.

Requests for registering specific Properties will also be evaluated to determine which of the above groups are appropriate.

The registrar retains the right to make a Property apply to a class of devices instead of a unique subset when it makes sense to do so. Since it is not very practical to *a priori* name all possible device classes, it is expected that the registrar will add new classes as they become clear.

Should a Property apply to unique subset, the string for the Key must be prefixed by a Vendor Name, each name allocated upon request.

Please note that requests may not be granted for a number of reasons, from simple duplication of an existing Property, to an incomplete definition of the Property. If a request is not granted, the requestor will be notified of the reason.

An example request might look something like this:

```
   From: fred@yoydyne.com
   Subject: Register New Device Properties

   Please register the following new Property as part of the class of
   network interface devices:

         Property:      phy-channel
         Value:         Integer (ASL assumes hexadecimal)
         Description:
            If present, defines the PHY channel to be used by this device
         Example:
            Package (2) { "phy-channel", 3 }

   Please register the new Vendor Name "Yoyodyne."

   Please register the device specific Property:

         Property:      yoyodyne,secret-sauce
         Value:         Integer (ASL assumes hexadecimal)
         Description:
            If present, used to determine Flux Capacitor size.
         Example:
```

```
Package (2) { "yoydyne,secret-sauce", 42 }
```