

sysctl(3) interface

Source: <http://unix.derkeiler.com/Mailing-Lists/FreeBSD/hackers/2007-01/msg00344.html>

- *From:* Daniel Rudy <dr2867@xxxxxxxxxxx>
 - *Date:* Sat, 27 Jan 2007 07:42:14 -0800
-

Hello List,

I've been taking apart and analyzing the sysctl(8) program to gain a better insight into how to use the sysctl(3) interface. Adding some debugging code to the program in strategic locations, this is what I have as an output:

```
debug: name: dev
debug: all: oid: 0 2 440
debug: name: dev.nexus.%parent
debug: oid: 440 912 913
debug: all: oid: 0 2 440 912 913
debug: name: dev.nexus.0.%desc
debug: oid: 440 912 914 915
debug: all: oid: 0 2 440 912 914 915
debug: name: dev.nexus.0.%driver
debug: oid: 440 912 914 916
debug: value: nexusdev.nexus.0.%driver: nexus
debug: all: oid: 0 2 440 912 914 916
debug: name: dev.nexus.0.%location
debug: oid: 440 912 914 917
debug: all: oid: 0 2 440 912 914 917
debug: name: dev.nexus.0.%pnpinfo
debug: oid: 440 912 914 918
debug: all: oid: 0 2 440 912 914 918
debug: name: dev.nexus.0.%parent
debug: oid: 440 912 914 919
debug: value: root0dev.nexus.0.%parent: root0
debug: all: oid: 0 2 440 912 914 919
debug: name: dev.acpi.%parent
debug: oid: 440 920 921
debug: all: oid: 0 2 440 920 921
debug: name: dev.acpi.0.%desc
debug: oid: 440 920 922 923
debug: value: AMIINT dev.acpi.0.%desc: AMIINT
debug: all: oid: 0 2 440 920 922 923
debug: name: dev.acpi.0.%driver
debug: oid: 440 920 922 924
```

sysctl(3) interface

```
debug: value: acpidev.acpi.0.%driver: acpi
debug: all: oid: 0 2 440 920 922 924
debug: name: dev.acpi.0.%location
debug: oid: 440 920 922 925
debug: all: oid: 0 2 440 920 922 925
debug: name: dev.acpi.0.%pnpinfo
debug: oid: 440 920 922 926
```

It's using an oid of 0 and 2 to get something, then it comes up with 440 and then a sequence of numbers that are incrementing in a peculiar pattern. I went looking and found that 0 is CTL_UNSPEC which according to the comment is unused, but I see it here in the program output.

I also noticed this little blurb in the source code too:

```
/*
 * These functions uses a presently undocumented interface to the kernel
 * to walk the tree and get the type so it can print the value.
 * This interface is under work and consideration, and should probably
 * be killed with a big axe by the first person who can find the time.
 * (be aware though, that the proper interface isn't as obvious as it
 * may seem, there are various conflicting requirements.
 */
```

But I figure it's for the actual display of the various variables and not for returning information about the dev tree.

So, my question is, how do I walk the tree to get the PnP info for all the devices in the system?

--

Daniel Rudy

freebsd-hackers@xxxxxxxxxxx mailing list

<http://lists.freebsd.org/mailman/listinfo/freebsd-hackers>

To unsubscribe, send any mail to "freebsd-hackers-unsubscribe@xxxxxxxxxxx"