

new/usr/src/common/smbios/smb_info.c

1

31886 Thu Mar 26 15:18:36 2015

new/usr/src/common/smbios/smb_info.c

SKU fix for 5094

_____unchanged_portion_omitted_____

```
411 int
412 smbios_info_chassis(smbios_hdl_t *shp, id_t id, smbios_chassis_t *chp)
413 {
414     const smb_struct_t *stp = smb_lookup_id(shp, id);
415     /* Length is measurable by one byte, so it'll be no more than 255. */
416     uint8_t buf[256]
417     smb_chassis_t *ch = (smb_chassis_t *)&(buf[0]);
418     smb_chassis_t ch;
```

```
419     if (stp == NULL)
420         return (-1); /* errno is set for us */
```

```
422     if (stp->smbst_hdr->smbh_type != SMB_TYPE_CHASSIS)
423         return (smb_set_errno(shp, ESMB_TYPE));
```

```
425     smb_info_bcopy(stp->smbst_hdr, ch, sizeof (buf));
426     smb_info_bcopy(stp->smbst_hdr, &ch, sizeof (ch));
427     bzero(chp, sizeof (smbios_chassis_t));
```

```
428     chp->smbc_oemdata = ch->smbch_oemdata;
429     chp->smbc_lock = (ch->smbch_type & SMB_CHT_LOCK) != 0;
430     chp->smbc_type = ch->smbch_type & ~SMB_CHT_LOCK;
431     chp->smbc_bustate = ch->smbch_bustate;
432     chp->smbc_psstate = ch->smbch_psstate;
433     chp->smbc_thstate = ch->smbch_thstate;
434     chp->smbc_security = ch->smbch_security;
435     chp->smbc_uheight = ch->smbch_uheight;
436     chp->smbc_cords = ch->smbch_cords;
437     chp->smbc_elems = ch->smbch_cn;
438     chp->smbc_elemilen = ch->smbch_cm;
439     chp->smbc_oemdata = ch.smbch_oemdata;
440     chp->smbc_lock = (ch.smbch_type & SMB_CHT_LOCK) != 0;
441     chp->smbc_type = ch.smbch_type & ~SMB_CHT_LOCK;
442     chp->smbc_bustate = ch.smbch_bustate;
443     chp->smbc_psstate = ch.smbch_psstate;
444     chp->smbc_thstate = ch.smbch_thstate;
445     chp->smbc_security = ch.smbch_security;
446     chp->smbc_uheight = ch.smbch_uheight;
447     chp->smbc_cords = ch.smbch_cords;
448     chp->smbc_elems = ch.smbch_cn;
449     chp->smbc_elemilen = ch.smbch_cm;
```

```
440     if (shp->sh_smbvers >= SMB_VERSION_27) {
441         (void) strncpy(chp->smbc_sku, SMB_CH_SKU(ch),
442             sizeof (chp->smbc_sku));
443     }
444     if (shp->sh_smbvers >= SMB_VERSION_27)
445         chp->smbc_sku = smb_strptr(stp, ch.smbch_sku);
```

```
445     return (0);
```

```
446 }
```

_____unchanged_portion_omitted_____

new/usr/src/uts/common/sys/smbios.h

1

67005 Thu Mar 26 15:18:36 2015

new/usr/src/uts/common/sys/smbios.h

SKU fix for 5094

unchanged portion omitted

```
280 #define SMB_BBFL_MOTHERBOARD 0x01 /* board is a motherboard */
281 #define SMB_BBFL_NEEDAUX 0x02 /* auxiliary card or daughter req'd */
282 #define SMB_BBFL_REMOVABLE 0x04 /* board is removable */
283 #define SMB_BBFL_REPLACABLE 0x08 /* board is field-replacable */
284 #define SMB_BBFL_HOTSWAP 0x10 /* board is hot-swappable */
```

```
286 #define SMB_BBT_UNKNOWN 0x1 /* unknown */
287 #define SMB_BBT_OTHER 0x2 /* other */
288 #define SMB_BBT_SBLADE 0x3 /* server blade */
289 #define SMB_BBT_CSWITCH 0x4 /* connectivity switch */
290 #define SMB_BBT_SMM 0x5 /* system management module */
291 #define SMB_BBT_PROC 0x6 /* processor module */
292 #define SMB_BBT_IO 0x7 /* i/o module */
293 #define SMB_BBT_MEM 0x8 /* memory module */
294 #define SMB_BBT_DAUGHTER 0x9 /* daughterboard */
295 #define SMB_BBT_MOTHER 0xA /* motherboard */
296 #define SMB_BBT_PROCMEM 0xB /* processor/memory module */
297 #define SMB_BBT_PROCIO 0xC /* processor/i/o module */
298 #define SMB_BBT_INTER 0xD /* interconnect board */
```

```
300 /*
301 * SMBIOS Chassis description. See DSP0134 Section 7.4 for more information.
302 * We move the lock bit of the type field into smbc_lock for easier processing.
303 */
```

```
304 typedef struct smbios_chassis {
305     uint32_t smbc_oemdata; /* OEM-specific data */
306     uint8_t smbc_lock; /* lock present? */
307     uint8_t smbc_type; /* type */
308     uint8_t smbc_bustate; /* boot-up state */
309     uint8_t smbc_psstate; /* power supply state */
310     uint8_t smbc_thstate; /* thermal state */
311     uint8_t smbc_security; /* security status */
312     uint8_t smbc_uheight; /* enclosure height in U's */
313     uint8_t smbc_cords; /* number of power cords */
314     uint8_t smbc_elems; /* number of element records (n) */
315     uint8_t smbc_elemlen; /* length of contained element (m) */
316     char smbc_sku[256]; /* SKU number (as a string) */
316     const char *smbc_sku; /* SKU number */
317 } smbios_chassis_t;
```

unchanged portion omitted

19937 Thu Mar 26 15:18:37 2015

new/usr/src/uts/common/sys/smbios_impl.h

SKU fix for 5094

unchanged_portion_omitted

```
99 typedef struct smb_chassis {
100     smb_header_t smbch_hdr;          /* structure header */
101     uint8_t smbch_manufacturer;      /* manufacturer */
102     uint8_t smbch_type;              /* type */
103     uint8_t smbch_version;           /* version */
104     uint8_t smbch_serial;            /* serial number */
105     uint8_t smbch_asset;             /* asset tag */
106     uint8_t smbch_bustate;           /* boot-up state */
107     uint8_t smbch_psstate;           /* power supply state */
108     uint8_t smbch_thstate;           /* thermal state */
109     uint8_t smbch_security;          /* security state */
110     uint32_t smbch_oemdata;          /* OEM-specific data */
111     uint8_t smbch_uheight;           /* enclosure height */
112     uint8_t smbch_cords;             /* number of power cords */
113     uint8_t smbch_cn;                /* number of contained records */
114     uint8_t smbch_cm;                /* size of contained records */
115     uint8_t smbch_cv[1];             /* array of contained records */
116     uint8_t smbch_sku;               /* SKU number */
117 } smb_chassis_t;
```

```
118 /* WARNING: the argument is evaluated three times! */
119 #define SMB_CH_SKU(smbcp) ((char *) \
120     (smbcp)->smbch_cv + ((smbcp)->smbch_cn * (smbcp)->smbch_cm))
121 #define SMB_CHT_LOCK    0x80        /* lock bit within smbch_type */
```

```
123 typedef struct smb_processor {
124     smb_header_t smbpr_hdr;          /* structure header */
125     uint8_t smbpr_socket;            /* socket designation */
126     uint8_t smbpr_type;              /* processor type (see <smbios.h>) */
127     uint8_t smbpr_family;            /* processor family (see <smbios.h>) */
128     uint8_t smbpr_manufacturer;      /* manufacturer */
129     uint64_t smbpr_cpuid;            /* processor cpuid information */
130     uint8_t smbpr_version;           /* version */
131     uint8_t smbpr_voltage;           /* voltage */
132     uint16_t smbpr_clkspeed;         /* external clock speed in MHz */
133     uint16_t smbpr_maxspeed;         /* maximum speed in MHz */
134     uint16_t smbpr_curspeed;         /* current speed in MHz */
135     uint8_t smbpr_status;            /* status (see <smbios.h>) */
136     uint8_t smbpr_upgrade;          /* upgrade */
137     uint16_t smbpr_l1cache;          /* L1 cache handle (if any) */
138     uint16_t smbpr_l2cache;          /* L2 cache handle (if any) */
139     uint16_t smbpr_l3cache;          /* L3 cache handle (if any) */
140     uint8_t smbpr_serial;            /* serial number */
141     uint8_t smbpr_asset;             /* asset tag */
142     uint8_t smbpr_part;              /* part number */
143     uint8_t smbpr_corecount;         /* number of cores per socket */
144     uint8_t smbpr_coresenabled;      /* number of enabled cores per socket */
145     uint8_t smbpr_threadcount;       /* number of threads per socket */
146     uint16_t smbpr_cflags;           /* cpu characteristics (see <smbios.h>) */
147     uint16_t smbpr_family2;          /* processor family2 (see <smbios.h>) */
148 } smb_processor_t;
```

unchanged_portion_omitted