```
**********************************************************
   24725 Tue Dec 10 20:23:53 2013
new/usr/src/cmd/sgs/libelf/common/update.c
4383 libelf can't write extended sections when ELF_F_LAYOUT
**********************************************************
______unchanged_portion_omitted_
```

```
static size_t
_elf_upd_usr(Elf * elf)
{
        NOTE(ASSUMING_PROTECTED(*elf))
        Lword           hi;
        Elf_Scn *       s;
        register Lword  sz;
        Ehdr *          eh = elf->ed_ehdr;
        unsigned        ver = eh->e_version;
        register char   *p = (char *)eh->e_ident;
        size_t          scncnt;


        /*
         * Ehdr and Phdr table go first
         */
        p[EI_MAG0] = ELFMAG0;
        p[EI_MAG1] = ELFMAG1;
        p[EI_MAG2] = ELFMAG2;
        p[EI_MAG3] = ELFMAG3;
        p[EI_CLASS] = ELFCLASS;
        /* LINTED */
        p[EI_VERSION] = (Byte)ver;
        hi = elf_fsize(ELF_T_EHDR, 1, ver);
        /* LINTED */
        eh->e_ehsize = (Half)hi;

        /*
         * If phnum is zero, phoff "should" be zero too,
         * but the application is responsible for it.
         * Allow a non-zero value here and update the
         * hi water mark accordingly.
         */

        if (eh->e_phnum != 0)
                /* LINTED */
                eh->e_phentsize = (Half)elf_fsize(ELF_T_PHDR, 1, ver);
        else
                eh->e_phentsize = 0;
        if ((sz = eh->e_phoff + eh->e_phentsize * eh->e_phnum) > hi)
                hi = sz;

        /*
         * Loop through sections, skipping index zero.
         * Compute section size before changing hi.
         * Allow null buffers for NOBITS.
         */

        if ((s = elf->ed_hdscn) == 0) {
        if ((s = elf->ed_hdscn) == 0)
                eh->e_shnum = 0;
                scncnt = 0;
        } else {
                scncnt = 1;
        else {
                eh->e_shnum = 1;
                *(Shdr*)s->s_shdr = _elf_snode_init.sb_shdr;
```

```
                s = s->s_next;
        }
        for (; s != 0; s = s->s_next) {
                register Dnode  *d;
                register Lword  fsz, j;
                Shdr *sh = s->s_shdr;

                if ((s->s_myflags & SF_READY) == 0)
                        (void) _elfxx_cookscn(s);

                ++scncnt;
                ++eh->e_shnum;
                sz = 0;
                for (d = s->s_hdnode; d != 0; d = d->db_next) {
                        if ((fsz = elf_fsize(d->db_data.d_type, 1,
                            ver)) == 0)
                                return (0);
                        j = _elf_msize(d->db_data.d_type, ver);
                        fsz *= (d->db_data.d_size / j);
                        d->db_osz = (size_t)fsz;

                        if ((sh->sh_type != SHT_NOBITS) &&
                            ((j = (d->db_data.d_off + d->db_osz)) > sz))
                                sz = j;
                }
                if (sh->sh_size < sz) {
                        _elf_seterr(EFMT_SCNSZ, 0);
                        return (0);
                }
                if ((sh->sh_type != SHT_NOBITS) &&
                    (hi < sh->sh_offset + sh->sh_size))
                        hi = sh->sh_offset + sh->sh_size;
        }

        /*
         * Shdr table last.  Comment above for phnum/phoff applies here.
         */
        if (scncnt != 0) {
        if (eh->e_shnum != 0)
                /* LINTED */
                eh->e_shentsize = (Half)elf_fsize(ELF_T_SHDR, 1, ver);
                if (scncnt < SHN_LORESERVE) {
                        eh->e_shnum = scncnt;
                } else {
                        Shdr *sh;
                        sh = (Shdr *)elf->ed_hdscn->s_shdr;
                        sh->sh_size = scncnt;
                        eh->e_shnum = 0;
                }
        } else {
        else
                eh->e_shentsize = 0;
        }
#endif /* ! codereview */

        if ((sz = eh->e_shoff + eh->e_shentsize * scncnt) > hi)
        if ((sz = eh->e_shoff + eh->e_shentsize * eh->e_shnum) > hi)
                hi = sz;

#ifdef TEST_SIZE
        if (test_size(hi) == 0)
                return (0);
#endif

        return ((size_t)hi);
}
______unchanged_portion_omitted_
```