

new/usr/src/cmd/Makefile.check

1

```

*****
4068 Wed Jan 2 21:49:14 2013
new/usr/src/cmd/Makefile.check
3442 sgs checks should be run in normal builds
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #

22 # Copyright (c) 2009, 2010, Oracle and/or its affiliates. All rights reserved.

24 include ../Makefile.master

26 #
27 # Commands providing manifests must offer a check target. A recursive check
28 # target across all commands directories is not currently provided.
29 #
30 MANIFEST_TOPDIRS=
31     acctadm //
32     auditd //
33     bnu //
34     consadm //
35     coreadm //
36     cron //
37     cvcd //
38     dispadmin //
39     drd //
40     dumpadm //
41     fcinfo //
42     fcoesvc //
43     fm //
44     ibd_upgrade //
45     intrd //
46     iscsid //
47     iscsitsvc //
48     kbd //
49     keyserv //
50     ldapcachemgr //
51     ldmad //
52     lms //
53     dlmgmt //
54     ndmpd //
55     nscd //
56     oplhpd //
57     pools //
58     power //
59     rexd //
60     rmvolmgr //
61     rpcbind //

```

new/usr/src/cmd/Makefile.check

2

```

62     rpcsvc //
63     sa //
64     saf //
65     sckmd //
66     sf880drd //
67     smserverd //
68     stmfproxy //
69     stmfsvc //
70     stmsboot //
71     syseventd //
72     syslogd //
73     utmpd //
74     vntsd //
75     ypcmd //
76     zoneadm //
77     zoneadmd //

79 MANIFEST_SUBDIRS=
80     boot/scripts //
81     cmd-crypto/scripts //
82     cmd-inet/usr.lib/ilbd //
83     cmd-inet/usr.lib/in.chargend //
84     cmd-inet/usr.lib/in.daytimed //
85     cmd-inet/usr.lib/in.dhcpd //
86     cmd-inet/usr.lib/in.discardd //
87     cmd-inet/usr.lib/in.echod //
88     cmd-inet/usr.lib/in.ndpd //
89     cmd-inet/usr.lib/in.ripngd //
90     cmd-inet/usr.lib/in.timed //
91     cmd-inet/usr.lib/inetd //
92     cmd-inet/usr.lib/mdnsd //
93     cmd-inet/usr.lib/slpd //
94     cmd-inet/usr.lib/vrrpd //
95     cmd-inet/usr.lib/wpad //
96     cmd-inet/usr.sbin //
97     cmd-inet/usr.sbin/in.ftpd //
98     cmd-inet/usr.sbin/in.rdisc //
99     cmd-inet/usr.sbin/in.routed //
100    cmd-inet/usr.sbin/in.talkd //
101    cmd-inet/usr.sbin/ipsecutils //
102    cmd-inet/usr.sbin/kssl/ksslcfg //
103    cmd-inet/usr.sbin/routeadm //
104    dcs/sparc/sun4u //
105    dfs.cmds/sharemgr //
106    fs.d/autofs //
107    fs.d/nfs/svc //
108    fs.d/smbclnt/svc //
109    gss/gssd //
110    hal/addons/network-devices //
111    hal/hald/solaris //
112    halt/smf.$(MACH) //
113    hostid/smf //
114    idmap/idmapd //
115    ipf/svc //
116    isns/isnsd //
117    krb5/kadmin/server //
118    krb5/krb5kdc //
119    krb5/kwarn //
120    krb5/slave //
121    lp/cmd/lpsched //
122    lvm/rpc.mdcommd //
123    lvm/rpc.metad //
124    lvm/rpc.metamedd //
125    lvm/rpc.metahd //
126    lvm/md_monitord //
127    lvm/util //

```

new/usr/src/cmd/Makefile.check

3

```
128     picl/picld           \
129     pools/poold         \
130     print/bsd-sysv-commands \
131     print/ppdmgr        \
132     rcap/rcapd          \
133     rpcsvc/rpc.bootparamd \
134     sendmail/lib        \
135     smbstrv/smbd        \
136     ssh/etc             \
137     svc/milestone       \
138     tsol/labeld        \
139     tsol/tnctl         \
140     tsol/tnid           \
141     tsol/tsol-zones     \
142     vscan/vscand       \
143     xvm/ipagent         \
144     ypcmd/yppasswd      \
145     ypcmd/ypupdated     \
146     zonestat/zonestatd
```

```
148 MISC_SUBDIRS= \
149     dtrace/test/tst \
150     sgs
148 $(CLOSED_BUILD)MANIFEST_SUBDIRS += \
149     $(CLOSED)/cmd/cmd-inet/usr.lib/in.iked
```

```
151 DTEST_SUBDIRS= \
152     dtrace/test/tst
```

152 .KEEP_STATE:

```
154 # Manifests cannot be checked in parallel, because we are using the global
155 # repository that is in $(SRC)/cmd/svc/seed/global.db. This is a
156 # repository that is built from the manifests in this workspace, whereas
157 # the build machine's repository may be out of sync with these manifests.
158 # Because we are using a private repository, svccfg-native must start up a
159 # private copy of configd-native. We cannot have multiple copies of
160 # configd-native trying to access global.db simultaneously.
```

162 .NO_PARALLEL:

```
164 check: svccfg_check $(MANIFEST_TOPDIRS) $(MANIFEST_SUBDIRS) $(MISC_SUBDIRS)
166 check: svccfg_check $(MANIFEST_TOPDIRS) $(MANIFEST_SUBDIRS) $(DTEST_SUBDIRS)
```

```
166 svccfg_check:
167     @$ (ECHO) "building requirements for svccfg check ..."; \
168     (cd $(SRC)/cmd/svc/seed && pwd && $(MAKE) $(MFLAGS) global.db)
```

```
170 $(MANIFEST_TOPDIRS) $(MANIFEST_SUBDIRS) $(MISC_SUBDIRS): FRC
172 $(MANIFEST_TOPDIRS) $(MANIFEST_SUBDIRS) $(DTEST_SUBDIRS): FRC
171     @cd $@; pwd; $(MAKE) check
```

173 FRC:

```

*****
4471 Wed Jan  2 21:49:15 2013
new/usr/src/cmd/sgs/Makefile
3442 sgs checks should be run in normal builds
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 # Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
23 #

25 include      $(SRC)/cmd/Makefile.cmd

27 # Note: Why SUBDIRS-common isn't sorted alphabetically
28 #
29 # The items under SGS are not independent of each other.
30 # They must be built in an order that ensures that
31 # all dependencies of an item have been built before the
32 # item itself.
33 #
34 SUBDIRS-common= libconv      \
35                 .WAIT       \
36                 libdl       \
37                 libelf      \
38                 liblddbg    \
39                 .WAIT       \
40                 libld        \
41                 libldmake   \
42                 libldstab   \
43                 librtld     \
44                 libcrle     \
45                 .WAIT       \
46                 0@0        \
47                 ld          \
48                 ldd         \
49                 lddstub     \
50                 rtld        \
51                 link_audit  \
52                 .WAIT       \
53                 librtld_db  \
54                 ldprof     \
55                 pvs         \
56                 crle       \
57                 ar          \
58                 dump        \
59                 elfdump    \
60                 elfedit    \
61                 elfwrap    \

```

```

62         error             \
63         gprof             \
64         lari              \
65         lex               \
66         lorder           \
67         m4                \
68         mcs               \
69         moe               \
70         nm                \
71         prof              \
72         ranlib            \
73         size              \
74         symorder         \
75         tsort             \
76         unifdef          \
77         yacc              \

79 SUBDIRS-i386=
80 SUBDIRS-sparc= rtld.4.x

82 SUBDIRS=          $(SUBDIRS-common) $(SUBDIRS-$(MACH))

84 # Messaging support
85 #
86 POSUBDIRS=        m4          nm          tsort          yacc
87 POFILE=           sgs.po
88 POFILES=          $(POSUBDIRS:%=%/.po)

90 MSGSUBDIRS=       ld          ldd          libld          liblddbg \
91                 libldstab   librtld     rtld          libelf   \
92                 ldprof     libcrle    pvs           elfdump  \
93                 elfedit    crle       moe           lari     \
94                 librtld_db  elfwrap    ar

96 MSGDIR=           messages

99 all :=            TARGET= all
100 install :=        TARGET= install
101 clean :=           TARGET= clean
102 clobber :=         TARGET= clobber
103 delete :=          TARGET= delete
104 lint :=            TARGET= lint
105 _msg :=            TARGET= catalog
106 _msg_gettext :=    TARGET= catalog
107 _msg_sgsmmsg :=    TARGET= catalog
108 chkmsg :=          TARGET= chkmsg

111 .KEEP_STATE:

113 .PARALLEL:        $(SUBDIRS)

115 all install:      native-add .WAIT $(SUBDIRS)

117 include           $(SRC)/cmd/Makefile.targ

119 # Messaging support
120 #
121 _msg: _msg_gettext _msg_sgsmmsg

123 _msg_gettext:     $(MSGDOMAIN)/$(POFILE)

125 _msg_sgsmmsg:     $(MSGDIR)

127 $(MSGDOMAIN)/$(POFILE): \

```

```

128             $(MSGDOMAIN) $(POFILE)
130 $(POFILE):   $(POSUBDIRS)
131             $(RM) $(POFILE)
132             cat $(POFILES) > $(POFILE)
134 $(MSGDIR):   $(MSGSUBDIRS) FRC
135             @ cd $@; pwd; $(MAKE) $(TARGET)
137 chkmsg:      libconv $(MSGSUBDIRS) FRC
139 check:       chkmsg
141 #endif /* ! codereview */
142 # built from lib/Makefile
143 install_lib:  FRC
144             @ cd lex; pwd; $(MAKE) $@
145             @ cd yacc; pwd; $(MAKE) $@
147 lint:        $(SUBDIRS)
149 delete \
150 clean clobber: native-clobber .WAIT $(SUBDIRS) $(MSGDIR)
152 $(SUBDIRS):  FRC
153             @ cd $@; pwd; $(MAKE) $(TARGET)
156 # Integration of ld and ld.so.1 in some development cycles requires that both
157 # of these modules be built using the new ld. This 'native' target allows us
158 # to build a local ld which will then be used to build the delivered version of
159 # itself and ld.so.1. Once this new functionality appears in the standard ld
160 # this target can be disabled.
162 native-add:   native-PROTO FRC
163             @ cd tools/$(MACH); pwd; $(MAKE) native
164             @ cd libconv/$(MACH); pwd; $(MAKE)
165             @ cd libelf/$(MACH); pwd; $(MAKE) native
166             @ cd liblddbg/$(MACH); pwd; $(MAKE) native
167             @ cd libldstab/$(MACH); pwd; $(MAKE) native
168             @ cd libld/$(MACH); pwd; $(MAKE) native
169             @ cd ld/$(MACH); pwd; $(MAKE) native
171 native-clobber:
172             @ cd tools; pwd; $(MAKE) $(TARGET)
173             $(RM) -r proto/$(MACH)
175 native-PROTO:
176             -@mkdir -p proto/$(MACH)
178 FRC:
180 #
181 # Cross-reference customization: ignore the directories named by XRPRUNE,
182 # and tweak the file globs slightly.
183 #
184 XRPRUNE=      rtd.4.x packages abi
185 XRADD=        *.msg mapfile* llib-[a-z]*
186 XRDEL=        Makefile* kobj_*
188 #
189 # Establish a set of directories for xref to search. As there are duplicates
190 # of things like headers, and only one file will be added to the xref database,
191 # we want xref to list the source file.
192 #
193 XRDIRS=       . \

```

```

194             ../../common/elfcap \
195             ../../head \
196             ../../uts/common/krtld \
197             ../../uts/common/sys \
198             ../../uts/sparc/sys \
199             ../../uts/sparc/krtld \
200             ../../uts/intel/ia32/krtld \
201             ../../uts/intel/amd64/krtld
203 xref:         FRC
204             @ $(RM) cscope.*
205             xref -p -x cscope

```



```

260 #
261 @ MSG_GRP_INVALIDDX "file %s: group section [%u]s: entry %d: \
262 invalid section index: %d"
263 @ MSG_GRP_INVALIDSYM "file %s: group section [%u]s: invalid group symbol %s"

265 # Relocation processing messages (some of these are required to satisfy
266 # do_reloc(), which is common code used by cmd/sgs/rtld - make sure both
267 # message files remain consistent).

269 @ MSG_REL_NOFIT "relocation error: %s: file %s: symbol %s: \
270 value 0x%llx does not fit"
271 @ MSG_REL_NONALIGN "relocation error: %s: file %s: symbol %s: \
272 offset 0x%llx is non-aligned"
273 @ MSG_REL_NULL "relocation error: file %s: section [%u]s: \
274 skipping null relocation record"
275 @ MSG_REL_NOTSUP "relocation error: %s: file %s: section [%u]s: \
276 relocation not currently supported"
277 @ MSG_REL_PICREDLOC "relocation error: %s: file %s: symbol %s: \
278 -z redlocsym may not be used for pic code"
279 @ MSG_REL_TLSLE "relocation error: %s: file %s: symbol %s: \
280 relocation illegal when building a shared object"
281 @ MSG_REL_TLSBND "relocation error: %s: file %s: symbol %s: \
282 bound to: %s: relocation illegal when not bound \
283 to object being created"
284 @ MSG_REL_TLSSTAT "relocation error: %s: file %s: symbol %s: \
285 relocation illegal when building a static object"
286 @ MSG_REL_TLSBADSYM "relocation error: %s: file %s: symbol %s: \
287 bad symbol type %s: symbol type must be TLS"
288 @ MSG_REL_BADTLS "relocation error: %s: file %s: symbol %s: \
289 relocation illegal for TLS symbol"
290 @ MSG_REL_BADGOTBASED "relocation error: %s: file %s: symbol %s: a GOT \
291 relative relocation must reference a local symbol"
292 @ MSG_REL_UNKNWSYM "relocation error: %s: file %s: section [%u]s: \
293 attempt to relocate with respect to unknown \
294 symbol %s: offset 0x%llx, symbol index %d"
295 @ MSG_REL_UNSUPSZ "relocation error: %s: file %s: symbol %s: \
296 offset size (%d bytes) is not supported"
297 @ MSG_REL_INVALIDOFFSET "relocation error: %s: file %s: section [%u]s: \
298 invalid offset symbol '%s': offset 0x%llx"
299 @ MSG_REL_INVALIDRELT "relocation error: file %s: section [%u]s: \
300 invalid relocation type: 0x%x"
301 @ MSG_REL_EMPTYSEC "relocation error: %s: file %s: symbol %s: \
302 attempted against empty section [%u]s"
303 @ MSG_REL_EXTERNSYM "relocation error: %s: file %s: symbol %s: \
304 external symbolic relocation against non-allocatable \
305 section %s; cannot be processed at runtime: \
306 relocation ignored"
307 @ MSG_REL_UNEXPREL "relocation error: %s: file %s: symbol %s: \
308 unexpected relocation; generic processing performed"
309 @ MSG_REL_UNEXPSYM "relocation error: %s: file %s: symbol %s: \
310 unexpected symbol referenced from file %s"
311 @ MSG_REL_SYMDISC "relocation error: %s: file %s: section [%u]s: \
312 symbol %s: symbol has been discarded with discarded \
313 section: [%u]s"
314 @ MSG_REL_NOSYMBOL "relocation error: %s: file %s: section: [%u]s: \
315 offset: 0x%llx: relocation requires reference symbol"
316 @ MSG_REL_DISPREL1 "relocation error: %s: file %s: symbol %s: \
317 displacement relocation applied to the symbol \
318 %s at 0x%llx: symbol %s is a copy relocated symbol"
319 @ MSG_REL_UNSUPSIZE "relocation error: %s: file %s: section [%u]s: \
320 relocation against section symbol unsupported"

322 @ MSG_REL_DISPREL2 "relocation warning: %s: file %s: symbol %s: \
323 may contain displacement relocation"
324 @ MSG_REL_DISPREL3 "relocation warning: %s: file %s: symbol %s: \
325 displacement relocation applied to the symbol \

```

```

326 %s: at 0x%llx: displacement relocation will not be \
327 visible in output image"
328 @ MSG_REL_DISPREL4 "relocation warning: %s: file %s: symbol %s: \
329 displacement relocation to be applied to the symbol \
330 %s: at 0x%llx: displacement relocation will be \
331 visible in output image"
332 @ MSG_REL_COPY "relocation warning: %s: file %s: symbol %s: \
333 relocation bound to a symbol with STV_PROTECTED \
334 visibility"
335 @ MSG_RELINVSEC "relocation warning: %s: file %s: section: [%u]s: \
336 against suspicious section [%u]s; relocation ignored"
337 @ MSG_REL_TLSIE "relocation warning: %s: file %s: symbol %s: \
338 relocation has restricted use when building a shared \
339 object"

341 @ MSG_REL_SLOPCDATNONAM "relocation warning: %s: file %s: section [%u]s: \
342 relocation against discarded COMDAT section [%u]s: \
343 redirected to file %s"
344 @ MSG_REL_SLOPCDATNAM "relocation warning: %s: file %s: section [%u]s: \
345 symbol %s: relocation against discarded COMDAT \
346 section [%u]s: redirected to file %s"
347 @ MSG_REL_SLOPCDATNOSYM "relocation warning: %s: file %s: section [%u]s: \
348 symbol %s: relocation against discarded COMDAT \
349 section [%u]s: symbol not found, relocation ignored"

351 @ MSG_REL_NOREG "relocation error: REGISTER relocation not supported \
352 on target architecture"

354 #
355 # TRANSLATION_NOTE
356 # The following 7 messages are the message to print the
357 # following example messages.
358 #
359 #Text relocation remains referenced
360 # against symbol offset in file
361 #str 0x14 main.o
362 #printf 0xc main.o
363 #
364 # The first two lines are the header, and the next msgid
365 # is the format string for the header.
366 # Tabs and spaces are used for alignment.
367 # The first and third %s are for: "Text relocation remains against symbol"
368 # The second %s and fourth %s are for: "referenced in file"
369 # The third %s is for: "offset"
370 #
371 @ MSG_REL_REMAIN_FMT_1 "%-40s\t%s\n %s\t\t %s\t%s"
372 #
373 # TRANSLATION_NOTE
374 # The next two msdid make a sentence. So translate:
375 # "Text relocation remain against symbol"
376 # And separate them into two msgstr considering the proper
377 # alignment.
378 @ MSG_REL_RMN_ITM_11 "Text relocation remains"
379 @ MSG_REL_RMN_ITM_12 "against symbol"
380 @ MSG_REL_RMN_ITM_13 "warning: Text relocation remains"

382 @ MSG_REL_RMN_ITM_2 "offset"

384 #
385 # TRANSLATION_NOTE
386 # The next two msdid make a sentence. So translate:
387 # "referenced in file"
388 # And separate them into two msgstr considering the proper
389 # alignment.
390 @ MSG_REL_RMN_ITM_31 "referenced"
391 @ MSG_REL_RMN_ITM_32 "in file"

```



```

524 @ MSG_PSYM_EXPReason1 "output file is static object"
525 @ MSG_PSYM_EXPReason2 "-z nopartial option in effect"
526 @ MSG_PSYM_EXPReason3 "move infrastructure size is greater than move data"

528 #
529 # Support library failures
530 #
531 @ MSG_SUP_NOLOAD "dlopen() of support library (%s) failed with \
532 error: %s"
533 @ MSG_SUP_BADVERSION "initialization of support library (%s) failed with \
534 bad version. supported: %d returned: %d"

537 #
538 # TRANSLATION_NOTE
539 # The following 7 messages are the message to print the
540 # following example messages.
541 #
542 #Undefined first referenced
543 # symbol in file
544 #inquire halt_hold.o
545 #
546 @ MSG_SYM_FMT_UNDEF "%s\t\t\t%s\
547 \n %s \t\t\t %s"

549 #
550 # TRANSLATION_NOTE
551 # The next two msdid make a sentence. So translate:
552 # "Undefined symbol"
553 # And separate them into two msgstr considering the proper
554 # alignment.
555 @ MSG_SYM_UNDEF_ITM_11 "Undefined"
556 @ MSG_SYM_UNDEF_ITM_12 "symbol"
557 #
558 # TRANSLATION_NOTE
559 # The next two msdid make a sentence. So translate:
560 # "first referenced in file"
561 # And separate them into two msgstr considering the proper
562 # alignment.
563 @ MSG_SYM_UNDEF_ITM_21 "first referenced"
564 @ MSG_SYM_UNDEF_ITM_22 "in file"
565 #

567 @ MSG_SYM_UND_UNDEF "%-35s %s"
568 @ MSG_SYM_UND_NOVER "%-35s %s (symbol has no version assigned)"
569 @ MSG_SYM_UND_IMPL "%-35s %s (symbol belongs to implicit dependency %s)"
570 @ MSG_SYM_UND_NOTA "%-35s %s (symbol belongs to unavailable version %s \
571 (%s))"
572 @ MSG_SYM_UND_BNDLOCAL "%-35s %s (symbol scope specifies local binding)"

574 @ MSG_SYM_ENTRY "entry point"
575 @ MSG_SYM_UNDEF "%s symbol '%s' is undefined"
576 @ MSG_SYM_EXTERN "%s symbol '%s' is undefined (symbol belongs to \
577 dependency %s)"
578 @ MSG_SYM_NOCRT "symbol '%s' not found, but %s section exists - \
579 possible link-edit without using the compiler driver"

581 # Output file update messages

583 @ MSG_UPD_NOREADSEG "No read-only segments found. Setting '_etext' to 0"
584 @ MSG_UPD_NORDWRSEG "No read-write segments found. Setting '_edata' to 0"
585 @ MSG_UPD_NOSEG "Setting 'end' and 'end' to 0"

587 @ MSG_UPD_SEGOVERLAP "%s: segment address overlap;\n\
588 \tprevious segment ending at address 0x%llx overlaps\n\
589 \tuser defined segment '%s' starting at address 0x%llx"

```

```

590 @ MSG_UPD_LARGSIZE "%s: segment %s calculated size 0x%llx\n\
591 \tis larger than user-defined size 0x%llx"

593 @ MSG_UPD_NOBITS "NOBITS section found before end of initialized data"
594 @ MSG_SEG_FIRNOTLOAD "First segment has type %s, PT_LOAD required: %s"
595 @ MSG_UPD_MULEHFRAME "file %s; section [%u]%s and file %s; section [%u]%s \
596 have incompatible attributes and cannot \
597 be merged into a single output section"

600 # Version processing messages

602 @ MSG_VER_HIGHER "file %s: version revision %d is higher than \
603 expected %d"
604 @ MSG_VER_NOEXIST "file %s: version '%s' does not exist:\n\
605 \trequired by file %s"
606 @ MSG_VER_UNDEF "version '%s' undefined, referenced by version '%s':\n\
607 \trequired by file %s"
608 @ MSG_VER_UNAVAIL "file %s: version '%s' is unavailable:\n\
609 \trequired by file %s"
610 @ MSG_VER_DEFINED "version symbol '%s' already defined in file %s"
611 @ MSG_VER_INVALID "version symbol '%s' from file %s has an invalid \
612 version index (%d)"
613 @ MSG_VER_ADDVERS "unused $ADDVERS specification from file '%s' \
614 for object '%s'\nversion(s):"
615 @ MSG_VER_ADDVER "\t%s"
616 @ MSG_VER_CYCLIC "following versions generate cyclic dependency:"

618 # Capabilities messages

620 @ MSG_CAP_MULDEF "capabilities symbol '%s' has multiply-defined members:"
621 @ MSG_CAP_MULDEFSYMS "\t(file %s symbol '%s'; file %s symbol '%s');"
622 @ MSG_CAP_REDUNDANT "file %s: section [%u]: symbol capabilities \
623 redundant, as object capabilities are more restrictive"
624 @ MSG_CAP_NOSYMSFOUND "no global symbols have been found that are associated \
625 with capabilities identified relocatable objects: \
626 -z symbolcap has no effect"

628 @ MSG_CAPINFO_INVALSYM "file %s: capabilities info section [%u]: index %d: \
629 family member symbol '%s': invalid"
630 @ MSG_CAPINFO_INVALIDLEAD "file %s: capabilities info section [%u]: index %d: \
631 family lead symbol '%s': invalid symbol index %d"

633 # Basic strings

635 @ MSG_STR_ALIGNMENTS "alignments"
636 @ MSG_STR_COMMAND "(command line)"
637 @ MSG_STR_TLSREL "(internal TLS relocation requirement)"
638 @ MSG_STR_SIZES "sizes"
639 @ MSG_STR_UNKNOWN "<unknown>"
640 @ MSG_STR_SECTION "%s (section)"
641 @ MSG_STR_SECTION_MSTR "%s (merged string section)"

643 #
644 # TRANSLATION_NOTE
645 # The elf_ function name represents a man page reference and should not
646 # be translated.
647 @ MSG_ELF_BEGIN "file %s: elf_begin"
648 @ MSG_ELF_CNTL "file %s: elf_cntl"
649 @ MSG_ELF_GETARHDR "file %s: elf_getarhdr"
650 @ MSG_ELF_GETARSYM "file %s: elf_getarsym"
651 @ MSG_ELF_GETDATA "file %s: elf_getdata"
652 @ MSG_ELF_GETEHDR "file %s: elf_getehdr"
653 @ MSG_ELF_GETPHDR "file %s: elf_getphdr"
654 @ MSG_ELF_GETSCN "file %s: elf_getscn: scnndx: %d"
655 @ MSG_ELF_GETSHDR "file %s: elf_getshdr"

```

```

656 @ MSG_ELF_MEMORY      "file %s: elf_memory"
657 @ MSG_ELF_NDXSCN      "file %s: elf_ndxscn"
658 @ MSG_ELF_NEWDATA      "file %s: elf_newdata"
659 @ MSG_ELF_NEWEHDR      "file %s: elf_newehdr"
660 @ MSG_ELF_NEWSCN       "file %s: elf_newscn"
661 @ MSG_ELF_NEWPHDR      "file %s: elf_newphdr"
662 @ MSG_ELF_STRPTR       "file %s: elf_strptr"
663 @ MSG_ELF_UPDATE       "file %s: elf_update"
664 @ MSG_ELF_SWAP_WRIMAGE "file %s: _elf_swap_wrimage"

667 @ MSG_REJ_MACH        "file %s: wrong ELF machine type: %s"
668 @ MSG_REJ_CLASS       "file %s: wrong ELF class: %s"
669 @ MSG_REJ_DATA        "file %s: wrong ELF data format: %s"
670 @ MSG_REJ_TYPE        "file %s: bad ELF type: %s"
671 @ MSG_REJ_BADFLAG     "file %s: bad ELF flags value: %s"
672 @ MSG_REJ_MISFLAG     "file %s: mismatched ELF flags value: %s"
673 @ MSG_REJ_VERSION     "file %s: mismatched ELF/lib version: %s"
674 @ MSG_REJ_HAL         "file %s: HAL Rl extensions required"
675 @ MSG_REJ_US3         "file %s: Sun UltraSPARC III extensions required"
676 @ MSG_REJ_STR         "file %s: %s"
677 @ MSG_REJ_UNKFILE     "file %s: unknown file type"
678 @ MSG_REJ_UNKCAP      "file=%s; unknown capability: %d"
679 @ MSG_REJ_HWCAP_1    "file %s: hardware capability (CA_SUNW_HW_1) \
680                          unsupported: %s"
681 @ MSG_REJ_SFCAP_1    "file %s: software capability (CA_SUNW_SF_1) \
682                          unsupported: %s"
683 @ MSG_REJ_MACHCAP    "file %s: machine capability (CA_SUNW_MACH) \
684                          unsupported: %s"
685 @ MSG_REJ_PLATCAP    "file %s: platform capability (CA_SUNW_PLAT) \
686                          unsupported: %s"
687 @ MSG_REJ_HWCAP_2    "file %s: hardware capability (CA_SUNW_HW_2) \
688                          unsupported: %s"
689 @ MSG_REJ_ARCHIVE    "file %s: invalid archive use"

691 # Guidance messages
692 @ MSG_GUIDE_SUMMARY   "see ld(1) -z guidance for more information"
693 @ MSG_GUIDE_DEFS      "-z defs option recommended for shared objects"
694 @ MSG_GUIDE_DIRECT    "-B direct or -z direct option recommended before \
695                          first dependency"
696 @ MSG_GUIDE_LAZYLOAD  "-z lazyload option recommended before \
697                          first dependency"
698 @ MSG_GUIDE_MAPFILE   "version 2 mapfile syntax recommended: %s"
699 @ MSG_GUIDE_TEXT      "position independent (PIC) code recommended for \
700                          shared objects"
701 @ MSG_GUIDE_UNUSED    "removal of unused dependency recommended: %s"

703 @ _END_

706 # The following strings represent reserved names. Reference to these strings
707 # is via the MSG_ORIG() macro, and thus translations are not required.

709 @ MSG_STR_EOF         "<eof>"
710 @ MSG_STR_ERROR       "<error>"
711 @ MSG_STR_EMPTY       ""
712 @ MSG_QSTR_BANG       "'!'"
713 @ MSG_STR_COLON       ":"
714 @ MSG_QSTR_COLON     "':'"
715 @ MSG_QSTR_SEMICOLON ";'"
716 @ MSG_QSTR_EQUAL     "'=''"
717 @ MSG_QSTR_PLUSEQ    "'+='"
718 @ MSG_QSTR_MINUSEQ   "'-='"
719 @ MSG_QSTR_ATSIGN    "'@'"
720 @ MSG_QSTR_DASH      "'-'"
721 @ MSG_QSTR_LEFTBKTT "'{'"

```

```

722 @ MSG_QSTR_RIGHTBKTT "'}'"
723 @ MSG_QSTR_PIPE      "'|'"
724 @ MSG_QSTR_STAR      "'*'"
725 @ MSG_STR_DOT        "."
726 @ MSG_STR_SLASH      "/"
727 @ MSG_STR_DYNAMIC    "(.dynamic)"
728 @ MSG_STR_ORIGIN     "$ORIGIN"
729 @ MSG_STR_MACHINE     "$MACHINE"
730 @ MSG_STR_PLATFORM   "$PLATFORM"
731 @ MSG_STR_ISALIST     "$ISALIST"
732 @ MSG_STR_OSNAME     "$OSNAME"
733 @ MSG_STR_OSREL      "$OSREL"
734 @ MSG_STR_UU_REAL_U  "__real_"
735 @ MSG_STR_UU_WRAP_U  "__wrap_"
736 @ MSG_STR_UELF32     "_ELF32"
737 @ MSG_STR_UELF64     "_ELF64"
738 @ MSG_STR_USPARC     "_sparc"
739 @ MSG_STR_UX86       "_x86"
740 @ MSG_STR_TRUE       "true"

742 @ MSG_STR_CDIR_ADD   "$add"
743 @ MSG_STR_CDIR_CLEAR "$clear"
744 @ MSG_STR_CDIR_ERROR "$error"
745 @ MSG_STR_CDIR_MFVER "$mapfile_version"
746 @ MSG_STR_CDIR_IF    "$if"
747 @ MSG_STR_CDIR_ELIF "$elif"
748 @ MSG_STR_CDIR_ELSE "$else"
749 @ MSG_STR_CDIR_ENDIF "$endif"

752 @ MSG_FMT_ARMEM      "%s(%s)"
753 @ MSG_FMT_COLPATH    "%s:%s"
754 @ MSG_FMT_SYMNAM     "%s'"
755 @ MSG_FMT_NULLSYMNAM "%s[%d]"
756 @ MSG_FMT_STRCAT     "%s%s"

758 @ MSG_PTH_RTLD       "/usr/lib/ld.so.1"

760 @ MSG_SUNW_OST_SGS  "SUNW_OST_SGS"

763 # Section strings

765 @ MSG_SCN_BSS        ".bss"
766 @ MSG_SCN_DATA       ".data"
767 @ MSG_SCN_COMMENT    ".comment"
768 @ MSG_SCN_DEBUG      ".debug"
769 @ MSG_SCN_DEBUG_INFO ".debug_info"
770 @ MSG_SCN_DYNAMIC    ".dynamic"
771 @ MSG_SCN_DYNSYMSORT ".SUNW_dynsymsort"
772 @ MSG_SCN_DYNTLSSORT ".SUNW_dyntlssort"
773 @ MSG_SCN_DYNSTR     ".dynstr"
774 @ MSG_SCN_DYNSYM     ".dynsym"
775 @ MSG_SCN_DYNSYM_SHNDX ".dynsym_shndx"
776 @ MSG_SCN_LDYNSYM    ".SUNW_ldynsym"
777 @ MSG_SCN_LDYNSYM_SHNDX ".SUNW_ldynsym_shndx"
778 @ MSG_SCN_EX_SHARED  ".ex_shared"
779 @ MSG_SCN_EX_RANGES  ".exception_ranges"
780 @ MSG_SCN_EXCL       ".excl"
781 @ MSG_SCN_FINI       ".fini"
782 @ MSG_SCN_FINIARRAY  ".fini_array"
783 @ MSG_SCN_GOT        ".got"
784 @ MSG_SCN_GNU_LINKONCE ".gnu.linkonce."
785 @ MSG_SCN_HASH       ".hash"
786 @ MSG_SCN_INDEX     ".index"
787 @ MSG_SCN_INIT       ".init"

```

```

788 @ MSG_SCN_INITARRAY      ".init_array"
789 @ MSG_SCN_INTERP         ".interp"
790 @ MSG_SCN_LBSS           ".lbss"
791 @ MSG_SCN_LDATA         ".ldata"
792 @ MSG_SCN_LINE          ".line"
793 @ MSG_SCN_LRODATA       ".lrodata"
794 @ MSG_SCN_PLT           ".plt"
795 @ MSG_SCN_PREINITARRAY   ".preinit_array"
796 @ MSG_SCN_REL           ".rel"
797 @ MSG_SCN_RELA          ".rela"
798 @ MSG_SCN_RODATA        ".rodata"
799 @ MSG_SCN_SBSS          ".sbss"
800 @ MSG_SCN_SBSS2         ".sbss2"
801 @ MSG_SCN_SDATA         ".sdata"
802 @ MSG_SCN_SDATA2        ".sdata2"
803 @ MSG_SCN_SHSTRTAB     ".shstrtab"
804 @ MSG_SCN_STAB          ".stab"
805 @ MSG_SCN_STABEXCL     ".stab.exclstr"
806 @ MSG_SCN_STRTAB        ".strtab"
807 @ MSG_SCN_SUNWMOVE      ".SUNW_move"
808 @ MSG_SCN_SUNWRELOC     ".SUNW_reloc"
809 @ MSG_SCN_SUNWSYMINFO   ".SUNW_syminfo"
810 @ MSG_SCN_SUNWVERSION   ".SUNW_version"
811 @ MSG_SCN_SUNWVERSYM    ".SUNW_versym"
812 @ MSG_SCN_SUNWCAP       ".SUNW_cap"
813 @ MSG_SCN_SUNWCAPINFO   ".SUNW_capinfo"
814 @ MSG_SCN_SUNWCAPCHAIN  ".SUNW_capchain"
815 @ MSG_SCN_SYMTAB        ".symtab"
816 @ MSG_SCN_SYMTAB_SHNDX  ".symtab_shndx"
817 @ MSG_SCN_TBSS          ".tbss"
818 @ MSG_SCN_TDATA         ".tdata"
819 @ MSG_SCN_TEXT           ".text"

821 @ MSG_SYM_FINIARRAY     "finiarray"
822 @ MSG_SYM_INITARRAY     "initarray"
823 @ MSG_SYM_PREINITARRAY  "preinitarray"

825 #
826 # GNU section names
827 #
828 @ MSG_SCN_CTORS          ".ctors"
829 @ MSG_SCN_DTORS          ".dtors"
830 @ MSG_SCN_EHFRAME        ".eh_frame"
831 @ MSG_SCN_EHFRAME_HDR    ".eh_frame_hdr"
832 @ MSG_SCN_GCC_X_TBL      ".gcc_except_table"
833 @ MSG_SCN_JCR            ".jcr"

835 # Segment names for segments referenced by entrance criteria

837 @ MSG_ENT_BSS           "bss"
838 @ MSG_ENT_DATA          "data"
839 @ MSG_ENT_EXTRA         "extra"
840 @ MSG_ENT_LDATA         "ldata"
841 @ MSG_ENT_LRODATA       "lrodata"
842 @ MSG_ENT_NOTE          "note"
843 @ MSG_ENT_TEXT          "text"

845 # Symbol names

847 @ MSG_SYM_START         "_start"
848 @ MSG_SYM_MAIN          "main"

850 @ MSG_SYM_FINI_U        "_fini"
851 @ MSG_SYM_INIT_U        "_init"
852 @ MSG_SYM_DYNAMIC       "DYNAMIC"
853 @ MSG_SYM_DYNAMIC_U     "_DYNAMIC"

```

```

854 @ MSG_SYM_EDATA         "edata"
855 @ MSG_SYM_EDATA_U       "_edata"
856 @ MSG_SYM_END           "end"
857 @ MSG_SYM_END_U         "_end"
858 @ MSG_SYM_ETEXT         "etext"
859 @ MSG_SYM_ETEXT_U       "_etext"
860 @ MSG_SYM_GOTBTL        "GLOBAL_OFFSET_TABLE_"
861 @ MSG_SYM_GOTBTL_U     "_GLOBAL_OFFSET_TABLE_"
862 @ MSG_SYM_PLKTBL        "PROCEDURE_LINKAGE_TABLE_"
863 @ MSG_SYM_PLKTBL_U     "_PROCEDURE_LINKAGE_TABLE_"
864 @ MSG_SYM_TLSETADDR_U  "__tls_get_addr"
865 @ MSG_SYM_TLSETADDR_UU "__tls_get_addr"

867 @ MSG_SYM_L_END         "END_"
868 @ MSG_SYM_L_END_U       "_END_"
869 @ MSG_SYM_L_START       "START_"
870 @ MSG_SYM_L_START_U     "_START_"

872 # Support functions

874 @ MSG_SUP_VERSION       "ld_version"
875 @ MSG_SUP_INPUT_DONE    "ld_input_done"

877 @ MSG_SUP_START_64     "ld_start64"
878 @ MSG_SUP_ATEXIT_64    "ld_atexit64"
879 @ MSG_SUP_OPEN_64      "ld_open64"
880 @ MSG_SUP_FILE_64      "ld_file64"
881 @ MSG_SUP_INSEC_64     "ld_input_section64"
882 @ MSG_SUP_SEC_64       "ld_section64"

884 @ MSG_SUP_START        "ld_start"
885 @ MSG_SUP_ATEXIT       "ld_atexit"
886 @ MSG_SUP_OPEN         "ld_open"
887 @ MSG_SUP_FILE         "ld_file"
888 @ MSG_SUP_INSEC        "ld_input_section"
889 @ MSG_SUP_SEC          "ld_section"

891 #
892 # Message previously in 'ld'
893 #
894 #
895 @ _START_

897 # System error messages

899 @ MSG_SYS_STAT          "file %s: stat failed: %s"
900 @ MSG_SYS_READ          "file %s: read failed: %s"
901 @ MSG_SYS_NOTREG       "file %s: is not a regular file"

903 # Argument processing messages

905 @ MSG_ARG_DY_INCOMP     "%s option is incompatible with building a dynamic \
906 executable"
907 @ MSG_MARG_DY_INCOMP    "%s is incompatible with building a dynamic \
908 executable"
909 @ MSG_ARG_ST_INCOMP     "%s option is incompatible with building a static \
910 object (-dn, -r, --relocatable)"
911 @ MSG_MARG_ST_INCOMP    "%s is incompatible with building a static \
912 object (-dn, -r, --relocatable)"
913 @ MSG_MARG_ST_ONLYAVL   "%s is only available when building a shared object"
914 @ MSG_ARG_INCOMP        "option %s and %s are incompatible"
915 @ MSG_MARG_INCOMP       "%s and %s are incompatible"
916 @ MSG_ARG_MTONCE        "option %s appears more than once, first setting taken"
917 @ MSG_MARG_MTONCE       "%s appears more than once, first setting taken"
918 @ MSG_ARG_ILLEGAL       "option %s has illegal argument '%s'"
919 @ MSG_ARG_YP            "option -YP and -Y%c may not be specified concurrently"

```



```

1052 @ MSG_MAP_NOSTACK1      "%s: %llu: %s not allowed on STACK segment"
1053 @ MSG_MAP_MOREONCE      "%s: %llu: %s set more than once on same line"
1054 @ MSG_MAP_NOTERM        "%s: %llu: unterminated quoted string: %s"
1055 @ MSG_MAP_SECINSEGE     "%s: %llu: section within segment ordering done on \
1056                          a non-existent segment '%s'"
1057 @ MSG_MAP_UNEXINHERIT    "%s: %llu: unnamed version cannot inherit from other \
1058                          versions: %s"
1059 @ MSG_MAP_UNEXTOK        "%s: %llu: unexpected occurrence of '%c' token"

1061 @ MSG_MAP_SEGEMPLOAD     "%s: %llu: empty segment must be of type LOAD or NULL"
1062 @ MSG_MAP_SEGEMPEXE     "%s: %llu: a LOAD empty segment definition is only \
1063                          allowed when creating a dynamic executable"
1064 @ MSG_MAP_SEGEMPATT      "%s: %llu: a LOAD empty segment must have an address \
1065                          and size"
1066 @ MSG_MAP_SEGEMPNOATT    "%s: %llu: a NULL empty segment must not have an \
1067                          address or size"
1068 @ MSG_MAP_SEGEMPSEC      "%s: %llu: empty segment can not have sections \
1069                          assigned to it"
1070 @ MSG_MAP_SEGEMNOPERM    "%s: %llu: empty segment must not have \
1071                          p_flags set: 0x%x"

1073 @ MSG_MAP_CNTADDRORDER  "%s: %llu: segment cannot have an explicit address \
1074                          and also be in the SEGMENT_ORDER list: %s"
1075 @ MSG_MAP_CNTDISSEG      "%s: %llu: segment cannot be disabled: %s"
1076 @ MSG_MAP_DUPNAMMENT     "%s: %llu: cannot redefine entrance criteria: %s"
1077 @ MSG_MAP_DUPORDSEG      "%s: %llu: segment is already in %s list: %s"
1078 @ MSG_MAP_DUP_OS_ORD     "%s: %llu: section is already in OS_ORDER list: %s"
1079 @ MSG_MAP_DUP_IS_ORD     "%s: %llu: entrance criteria is already in \
1080                          IS_ORDER list: %s"
1081 @ MSG_MAP_UNKENT         "%s: %llu: unknown entrance criteria \
1082                          (ASSIGN_SECTION): %s"
1083 @ MSG_MAP_UNKSEG         "%s: %llu: unknown segment: %s"
1084 @ MSG_MAP_UNKSYMDEF      "%s: %llu: unknown symbol definition: %s"
1085 @ MSG_MAP_UNKSEGTYPER    "%s: %llu: unknown internal segment type %d"
1086 @ MSG_MAP_UNKSOOTYP      "%s: %llu: unknown shared object type: %s"
1087 @ MSG_MAP_UNKSEGATT      "%s: %llu: unknown segment attribute: %s"
1088 @ MSG_MAP_UNKSEGFLG      "%s: %llu: unknown segment flag: ?%c"
1089 @ MSG_MAP_UNKSECTYP      "%s: %llu: unknown section type: %s"

1091 @ MSG_MAP_SEGSIZE       "%s: %lld: existing segment size symbols cannot \
1092                          be reset: %s"
1093 @ MSG_MAP_SEGADDR       "%s: %llu: segment address or length '%s' %s"
1094 @ MSG_MAP_BADCAPVAL     "%s: %llu: bad capability value: %s"
1095 @ MSG_MAP_UNKCAPATTR    "%s: %llu: unknown capability attribute '%s'"
1096 @ MSG_MAP_EMPTYCAP      "%s: %llu: empty capability definition; ignored"

1098 @ MSG_MAP_SYMDEF1       "%s: %llu: symbol '%s' is already defined in file: \
1099                          %s"
1100 @ MSG_MAP_SYMDEF2       "%s: %llu: symbol '%s': %s"

1102 @ MSG_MAP_EXPSCOL       "%s: %llu: expected a ';' "
1103 @ MSG_MAP_EXPEQU        "%s: %llu: expected a '=', ':', '|', or '@' "
1104 @ MSG_MAP_EXPSEGATT      "%s: %llu: expected one or more segment attributes \
1105                          after an '=' "
1106 @ MSG_MAP_EXPSEGNAM      "%s: %llu: expected a segment name at the beginning \
1107                          of a line"
1108 @ MSG_MAP_EXPSEGTYPER    "%s: %llu: %s segment cannot be used with %s \
1109                          directive: %s"
1110 @ MSG_MAP_EXPSYM_1       "%s: %llu: expected a symbol name after '@' "
1111 @ MSG_MAP_EXPSYM_2       "%s: %llu: expected a symbol name after '{' "
1112 @ MSG_MAP_EXPSEC         "%s: %llu: expected a section name after '|' "
1113 @ MSG_MAP_EXPSEO         "%s: %llu: expected a shared object definition \
1114                          after '-' "
1115 @ MSG_MAP_MULTFILTEE     "%s: %llu: multiple filtee definitions are unsupported"
1116 @ MSG_MAP_NOFILTER      "%s: %llu: filtee definition required"
1117 @ MSG_MAP_BADSF1        "%s: %llu: unknown software capabilities: 0x%llx; \

```

```

1118                          ignored"
1119 @ MSG_MAP_INADDR32SF1    "%s: %llu: software capability ADDR32: is ineffective \
1120                          when building 32-bit object: ignored"
1121 @ MSG_MAP_NOINTPOSE      "%s: %llu: interposition symbols can only be defined \
1122                          when building a dynamic executable"
1123 @ MSG_MAP_NOEXVLSZ      "%s: %llu: value and size attributes are incompatible \
1124                          with extern or parent symbols"
1125 @ MSG_MAP_FLTR_ONLYAVL   "%s: %llu: symbol filtering is only available when \
1126                          building a shared object"

1128 @ MSG_MAP_SEGSAME       "segments '%s' and '%s' have the same assigned \
1129                          virtual address"
1130 @ MSG_MAP_EXCLIMIT       "exceeds internal limit"
1131 @ MSG_MAP_NOBADFRM       "number is badly formed"

1133 @ MSG_MAP_SEGTYP        "segment type"
1134 @ MSG_MAP_SEGVADDR       "segment virtual address"
1135 @ MSG_MAP_SEGPHYS        "segment physical address"
1136 @ MSG_MAP_SEGLEN        "segment length"
1137 @ MSG_MAP_SEGFLAG        "segment flags"
1138 @ MSG_MAP_SEGALIGN       "segment alignment"
1139 @ MSG_MAP_SEGROUND       "segment rounding"

1141 @ MSG_MAP_SECTYP        "section type"
1142 @ MSG_MAP_SECFLAG        "section flags"
1143 @ MSG_MAP_SECNAM        "section name"

1145 @ MSG_MAP_SYMVAL        "symbol value"
1146 @ MSG_MAP_SYMSIZE       "symbol size"

1148 @ MSG_MAP_DIFF_SYMVAL   "symbol values differ"
1149 @ MSG_MAP_DIFF_SYMSZ    "symbol sizes differ"
1150 @ MSG_MAP_DIFF_SYMTYP   "symbol types differ"
1151 @ MSG_MAP_DIFF_SYMNDX   "symbol indexes differ"
1152 @ MSG_MAP_DIFF_SYMLCL   "symbol scope conflict against local and non-local"
1153 @ MSG_MAP_DIFF_SYMGLOB  "symbol scope conflict against singleton/exported"
1154 @ MSG_MAP_DIFF_SYMPROT  "symbol scope conflict against protected"
1155 @ MSG_MAP_DIFF_SYMVER   "symbol version conflict"
1156 @ MSG_MAP_DIFF_SYMMUL   "symbol multiple definition"
1157 @ MSG_MAP_DIFF_SNGLDIR  "singleton scope and direct declaration are \
1158                          incompatible"
1159 @ MSG_MAP_DIFF_PROTNDIR "protected scope and no-direct declaration \
1160                          are incompatible"

1163 @ MSG_MAP_SECCORDER     "section ordering requested, but no matching section \
1164                          found: segment: %s section: %s"

1167 # Mapfile Directives

1169 @ MSG_MAP_EXP_ATTR       "%s: %llu: expected attribute name (%s), or \
1170                          terminator (';', '|'): %s"
1171 @ MSG_MAP_EXP_CAPMASK    "%s: %llu: expected capability name, integer value, or \
1172                          terminator (';', '|'): %s"
1173 @ MSG_MAP_EXP_CAPNAME    "%s: %llu: expected name, or terminator (';', '|'): %s"
1174 @ MSG_MAP_EXP_CAPID      "%s: %llu: expected name, or '{' following %s: %s"
1175 @ MSG_MAP_EXP_CAPHW      "%s: %llu: expected hardware capability, or \
1176                          terminator (';', '|'): %s"
1177 @ MSG_MAP_EXP_CAPSF      "%s: %llu: expected software capability, or \
1178                          terminator (';', '|'): %s"
1179 @ MSG_MAP_EXP_EQ         "%s: %llu: expected '=' following %s: %s"
1180 @ MSG_MAP_EXP_EQ_ALL     "%s: %llu: expected '=', '+=' , or '-=' following %s: %s"
1181 @ MSG_MAP_EXP_EQ_PEQ     "%s: %llu: expected '=' following %s: %s"
1182 @ MSG_MAP_EXP_DIR        "%s: %llu: expected mapfile directive (%s): %s"
1183 @ MSG_MAP_SFLG_EXBANG    "%s: %llu: '!' appears without corresponding flag"

```

```

1184 @ MSG_MAP_EXP_FILNAM      "%s: %llu: expected file name following %s: %s"
1185 @ MSG_MAP_EXP_FILPATH     "%s: %llu: expected file path following %s: %s"
1186 @ MSG_MAP_EXP_INT         "%s: %llu: expected integer value following %s: %s"
1187 @ MSG_MAP_EXP_LBKT        "%s: %llu: expected '{' following %s: %s"
1188 @ MSG_MAP_EXP_OBJNAM      "%s: %llu: expected object name following %s: %s"
1189 @ MSG_MAP_SFLG_ONEBANG     "%s: %llu: '!' can only be specified once per flag"
1190 @ MSG_MAP_EXP_SECFLAG     "%s: %llu: expected section flag (%s), '!', or \
1191 terminator (';', ')': %s"
1192 @ MSG_MAP_EXP_SECNAM      "%s: %llu: expected section name following %s: %s"
1193 @ MSG_MAP_EXP_SEGFLAG     "%s: %llu: expected segment flag (%s), or \
1194 terminator (';', ')': %s"
1195 @ MSG_MAP_EXP_ECNAM      "%s: %llu: expected entrance criteria (ASSIGN_SECTION) \
1196 name, or terminator (';', ')': %s"
1197 @ MSG_MAP_EXP_SEGNAM      "%s: %llu: expected segment name following %s: %s"
1198 @ MSG_MAP_EXP_SEM         "%s: %llu: expected ';' to terminate %s: %s"
1199 @ MSG_MAP_EXP_SEMLBKT     "%s: %llu: expected ';' or '{' following %s: %s"
1200 @ MSG_MAP_EXP_SEMRBKT     "%s: %llu: expected ';' or '}' to terminate %s: %s"
1201 @ MSG_MAP_EXP_SHTYPE      "%s: %llu: expected section type: %s"
1202 @ MSG_MAP_EXP_SYM         "%s: %llu: expected symbol name, symbol scope, \
1203 or '*': %s"
1204 @ MSG_MAP_EXP_SYMEND      "%s: %llu: expected inherited version name, or \
1205 terminator (';'): %s"
1206 @ MSG_MAP_EXP_SYMDELIM    "%s: %llu: expected one of ':', ';', or '{': %s"
1207 @ MSG_MAP_EXP_SYMFLAG     "%s: %llu: expected symbol flag (%s), or \
1208 terminator (';', ')': %s"
1209 @ MSG_MAP_EXP_SYMNAM      "%s: %llu: expected symbol name following %s: %s"
1210 @ MSG_MAP_EXP_SYMSCOPE    "%s: %llu: expected symbol scope (%s): %s"
1211 @ MSG_MAP_EXP_SYMTYPE     "%s: %llu: expected symbol type (%s): %s"
1212 @ MSG_MAP_EXP_VERSION     "%s: %llu: expected version name following %s: %s"
1213 @ MSG_MAP_BADEXTRA        "%s: %llu: unexpected text found following %s directive"
1214 @ MSG_MAP_VALUELIMIT     "%s: %llu: numeric value exceeds word size: %s"
1215 @ MSG_MAP_MALVALUE        "%s: %llu: malformed numeric value: %s"
1216 @ MSG_MAP_BADVALUETAIL   "%s: %llu: unexpected characters following numeric \
1217 constant: %s"
1218 @ MSG_MAP_WSNEEDED        "%s: %llu: whitespace needed before token: %s"
1219 @ MSG_MAP_BADCHAR         "%s: %llu: unexpected text: %s"
1220 @ MSG_MAP_BADKWQUOTE     "%s: %llu: mapfile keywords should not be quoted: %s"
1221 @ MSG_MAP_CDIR_NOTBOL     "%s: %llu: mapfile control directive not at start of \
1222 line: %s"
1223 @ MSG_MAP_NOATTR         "%s: %llu: %s specified no attributes (empty {})"
1224 @ MSG_MAP_NOVALUES        "%s: %llu: %s specified without values"
1225 @ MSG_MAP_INTERR          "%<internal error>"
1226 @ MSG_MAP_ISORDVER        "%s: %llu: version 0 mapfile ?O flag and version 1 \
1227 segment IS_ORDER attribute are mutually exclusive: %s"
1228 @ MSG_MAP_SYMATTR        "%symbol attributes";

1230 # Mapfile Control Directives

1232 @ MSG_MAP_CDIR_BADVDIR    "%s: %llu: $mapfile_version directive must specify \
1233 version 2 or higher: %d"
1234 @ MSG_MAP_CDIR_BADVER     "%s: %llu: unknown mapfile version: %d"
1235 @ MSG_MAP_CDIR_REPVER     "%s: %llu: $mapfile_version must be first directive \
1236 in file"
1237 @ MSG_MAP_CDIR_REQARG     "%s: %llu: %s directive requires an argument"
1238 @ MSG_MAP_CDIR_REQNOARG   "%s: %llu: %s directive does not accept arguments"
1239 @ MSG_MAP_CDIR_BAD        "%s: %llu: unrecognized mapfile control directive"
1240 @ MSG_MAP_CDIR_NOIF       "%s: %llu: %s directive used without opening $if"
1241 @ MSG_MAP_CDIR_ELSE       "%s: %llu: %s directive preceded by $else on line %d"
1242 @ MSG_MAP_CDIR_NOEND      "%s: %llu: EOF encountered without closing $endif \
1243 for $if on line %d"
1244 @ MSG_MAP_CDIR_ERROR      "%s: %llu: error: %s"

1247 # Mapfile Conditional Expressions

1249 @ MSG_MAP_CEXP_TOKERR     "%s: %llu: syntax error in conditional expression at: %s"

```

```

1250 @ MSG_MAP_CEXP_SEMERR    "%s: %llu: malformed conditional expression"
1251 @ MSG_MAP_CEXP_BADOPUSE  "%s: %llu: invalid operator use in conditional \
1252 expression"
1253 @ MSG_MAP_CEXP_UNBALPAR  "%s: %llu: unbalanced parenthesis in conditional \
1254 expression"
1255 @ MSG_MAP_BADCESC        "%s: %llu: unrecognized escape in double quoted \
1256 token: \\%c\n"

1258 # Generic error diagnostic labels

1260 @ MSG_STR_NULL           "(null)"

1262 @ MSG_DBG_DFLT_FMT       "debug: "
1263 @ MSG_DBG_AOUT_FMT       "debug: a.out: "
1264 @ MSG_DBG_NAME_FMT       "debug: %s: "

1266 # -z assert-deflib strings

1268 @ MSG_ARG_ASSDEFLIB_MALFORMED "library name malformed: %s"
1269 @ MSG_ARG_ASSDEFLIB_FOUND    "dynamic library found on default search path \
1270 (%s): lib%s.so"
1271 #endif /* ! codereview */

1273 @ _END_

1276 # Software identification. Note, the SGU strings is historic, and has
1277 # little relevance. It is preserved as applications have used this
1278 # string to identify the Solaris link-editor.

1280 @ MSG_SGS_ID             "ld: Software Generation Utilities - \
1281 Solaris Link Editors: "

1283 # The following strings represent reserved words, files, pathnames and symbols.
1284 # Reference to this strings is via the MSG_ORIG() macro, and thus no message
1285 # translation is required.

1287 @ MSG_DBG_FOPEN_MODE     "w"

1289 @ MSG_DBG_CLS32_FMT      "32: "
1290 @ MSG_DBG_CLS64_FMT      "64: "

1292 @ MSG_STR_PATHTOK        ";"
1293 @ MSG_STR_AOUT           "a.out"

1295 @ MSG_STR_LIB_A          "%s/lib%s.a"
1296 @ MSG_STR_LIB_SO         "%s/lib%s.so"
1297 @ MSG_STR_PATH           "%s/%s"
1298 @ MSG_STR_STRLN         "%s\n"
1299 @ MSG_STR_NL             "\n"
1300 @ MSG_STR_CAPGROUPID     "CAP_GROUP_%d"

1302 @ MSG_STR_LD_DYNAMIC     "dynamic"
1303 @ MSG_STR_SYMBOLIC       "symbolic"
1304 @ MSG_STR_ELIMINATE     "eliminate"
1305 @ MSG_STR_LOCAL          "local"
1306 @ MSG_STR_PROGBITS       "progbits"
1307 @ MSG_STR_SYMTAB         "symtab"
1308 @ MSG_STR_DYNSYM         "dynsym"
1309 @ MSG_STR_REL            "rel"
1310 @ MSG_STR_RELA           "rela"
1311 @ MSG_STR_STRTAB         "strtab"
1312 @ MSG_STR_HASH           "hash"
1313 @ MSG_STR_LIB            "lib"
1314 @ MSG_STR_NOTE           "note"
1315 @ MSG_STR_NOBITS         "nobits"

```

```

1316 @ MSG_STR_HWCAP_1      "hwcap_1"
1317 @ MSG_STR_SFCAP_1      "sficap_1"
1318 @ MSG_STR_SOEXT         ".so"

1320 @ MSG_STR_OPTIONS       "3:6:abc:d:e:f:h:il:mo:p:rstu:z:B:CD:F:GI:L:M:N:P:Q:R:\
1321                          S:VW:Y:?"

1323 # Argument processing strings

1325 @ MSG_ARG_3             "-3"
1326 @ MSG_ARG_6             "-6"
1327 @ MSG_ARG_A             "-a"
1328 @ MSG_ARG_B             "-b"
1329 @ MSG_ARG_CB           "-B"
1330 @ MSG_ARG_BDIRECT       "-Bdirect"
1331 @ MSG_ARG_BDYNAMIC       "-Bdynamic"
1332 @ MSG_ARG_BELIMINATE    "-Beliminate"
1333 @ MSG_ARG_BGROUP        "-Bgroup"
1334 @ MSG_ARG_BLOCAL        "-Blocal"
1335 @ MSG_ARG_BNODIRECT     "-Bnodirect"
1336 @ MSG_ARG_BSYMBOLIC     "-Bsymbolic"
1337 @ MSG_ARG_BTRANSLATOR   "-Btranslator"
1338 @ MSG_ARG_C             "-c"
1339 @ MSG_ARG_D             "-d"
1340 @ MSG_ARG_DY            "-dy"
1341 @ MSG_ARG_CI            "-I"
1342 @ MSG_ARG_CN            "-N"
1343 @ MSG_ARG_P             "-p"
1344 @ MSG_ARG_CP            "-P"
1345 @ MSG_ARG_CQ            "-Q"
1346 @ MSG_ARG_CY            "-Y"
1347 @ MSG_ARG_CYL           "-YL"
1348 @ MSG_ARG_CYP           "-YP"
1349 @ MSG_ARG_CYU           "-YU"
1350 @ MSG_ARG_Z             "-z"
1351 @ MSG_ARG_ZDEFNODEF     "-z[defs|nodefs]"
1352 @ MSG_ARG_ZGUIDE        "-zguidance"
1353 @ MSG_ARG_ZNODEF        "-znodefs"
1354 @ MSG_ARG_ZNOINTERP     "-znointerp"
1355 @ MSG_ARG_ZRELAXRELOC   "-zrelaxreloc"
1356 @ MSG_ARG_ZNORELAXRELOC "-znorelaxreloc"
1357 @ MSG_ARG_ZTEXT         "-ztext"
1358 @ MSG_ARG_ZTEXTOFF      "-ztextoff"
1359 @ MSG_ARG_ZTEXTWARN     "-ztextwarn"
1360 @ MSG_ARG_ZTEXTALL      "-z[text|textwarn|textoff]"
1361 @ MSG_ARG_ZLOADFLTR     "-zloadfltr"
1362 @ MSG_ARG_ZCOMBRELOC    "-zcombreloc"
1363 @ MSG_ARG_ZSYMBOLCAP    "-zsymbolcap"
1364 @ MSG_ARG_ZFATWNOFATW   "-z[fatal-warnings|nofatalwarnings]"

1366 @ MSG_ARG_ABSEXEC       "absexec"
1367 @ MSG_ARG_ALTEXEC64     "altexec64"
1368 @ MSG_ARG_NOCOMPSTRTAB  "nocompstrtab"
1369 @ MSG_ARG_GROUPPERM     "groupperm"
1370 @ MSG_ARG_NOGROUPPERM  "nogroupperm"
1371 @ MSG_ARG_LAZYLOAD      "lazyload"
1372 @ MSG_ARG_NOLAZYLOAD    "nolazyload"
1373 @ MSG_ARG_INTERPOSE     "interpose"
1374 @ MSG_ARG_DIRECT        "direct"
1375 @ MSG_ARG_NODIRECT      "nodirect"
1376 @ MSG_ARG_IGNORE        "ignore"
1377 @ MSG_ARG_RECORD        "record"
1378 @ MSG_ARG_INITFIRST     "initfirst"
1379 @ MSG_ARG_INITARRAY     "initarray="
1380 @ MSG_ARG_FINIARRAY     "finiarray="
1381 @ MSG_ARG_PREINITARRAY   "preinitarray="

```

```

1382 @ MSG_ARG_RTLDINFO     "rtldinfo="
1383 @ MSG_ARG_DTRACE        "dtrace="
1384 @ MSG_ARG_TRANSLATOR    "translator"
1385 @ MSG_ARG_NOOPEN       "nodlopen"
1386 @ MSG_ARG_NOW          "now"
1387 @ MSG_ARG_ORIGIN       "origin"
1388 @ MSG_ARG_DEFS         "defs"
1389 @ MSG_ARG_NODEFS       "nodefs"
1390 @ MSG_ARG_NODUMP       "nodump"
1391 @ MSG_ARG_NOVERSION     "noversion"
1392 @ MSG_ARG_TEXT         "text"
1393 @ MSG_ARG_TEXTOFF      "textoff"
1394 @ MSG_ARG_TEXTWARN     "textwarn"
1395 @ MSG_ARG_MULDEFS      "muldefs"
1396 @ MSG_ARG_NODELETE     "nodelete"
1397 @ MSG_ARG_NOINTERP     "nointerp"
1398 @ MSG_ARG_NOPARTIAL    "nopartial"
1399 @ MSG_ARG_NORELOC      "noreloc"
1400 @ MSG_ARG_REDLCSYM     "redlocsymb"
1401 @ MSG_ARG_VERBOSE      "verbose"
1402 @ MSG_ARG_WEAKEXT      "weakextract"
1403 @ MSG_ARG_LOADFLTR     "loadfltr"
1404 @ MSG_ARG_ALLEXTRT     "allextract"
1405 @ MSG_ARG_DFLEXTRT     "defaultextract"
1406 @ MSG_ARG_COMBRELOC    "combreloc"
1407 @ MSG_ARG_NOCOMBRELOC  "nocombreloc"
1408 @ MSG_ARG_NODEFAULTLIB  "nodefaultlib"
1409 @ MSG_ARG_ENDFILTEE    "endfiltee"
1410 @ MSG_ARG_LD32         "ld32="
1411 @ MSG_ARG_LD64         "ld64="
1412 @ MSG_ARG_RESCAN       "rescan"
1413 @ MSG_ARG_RESCAN_NOW   "rescan-now"
1414 @ MSG_ARG_RESCAN_START "rescan-start"
1415 @ MSG_ARG_RESCAN_END   "rescan-end"
1416 @ MSG_ARG_GUIDE        "guidance"
1417 @ MSG_ARG_NOLDYNSYM    "noldynsym"
1418 @ MSG_ARG_RELAXRELOC   "relaxreloc"
1419 @ MSG_ARG_NORELAXRELOC "norelaxreloc"
1420 @ MSG_ARG_NOSIGHANDLER "nosighandler"
1421 @ MSG_ARG_GLOBAUDIT    "globalaudit"
1422 @ MSG_ARG_TARGET       "target="
1423 @ MSG_ARG_WRAP         "wrap="
1424 @ MSG_ARG_FATWARN      "fatal-warnings"
1425 @ MSG_ARG_NOFATWARN    "nofatal-warnings"
1426 @ MSG_ARG_HELP         "help"
1427 @ MSG_ARG_GROUP        "group"
1428 @ MSG_ARG_REDUCE       "reduce"
1429 @ MSG_ARG_STATIC       "static"
1430 @ MSG_ARG_SYMBOLCAP     "symbolcap"
1431 @ MSG_ARG_DEFERRED     "deferred"
1432 @ MSG_ARG_NODEFERRED   "nodeferred"
1433 @ MSG_ARG_ASSERTFLIB   "assert-deflib"

1435 @ MSG_ARG_LCOM         "L,"
1436 @ MSG_ARG_PCOM         "P,"
1437 @ MSG_ARG_UCOM         "U,"

1439 @ MSG_ARG_T_RPATH      "rpath"
1440 @ MSG_ARG_T_SHARED     "shared"
1441 @ MSG_ARG_T_SONAME     "soname"
1442 @ MSG_ARG_T_WL         "l,-"

1444 @ MSG_ARG_T_AUXFLTR    "-auxiliary"
1445 @ MSG_ARG_T_MULDEFS    "-allow-multiple-definition"
1446 @ MSG_ARG_T_INTERP     "-dynamic-linker"
1447 @ MSG_ARG_T_ENDGROUP   "-end-group"

```

```

1448 @ MSG_ARG_T_ENTRY      "--entry"
1449 @ MSG_ARG_T_STDFLTR    "--filter"
1450 @ MSG_ARG_T_FATWARN     "--fatal-warnings"
1451 @ MSG_ARG_T_NOFATWARN  "--no-fatal-warnings"
1452 @ MSG_ARG_T_HELP      "--help"
1453 @ MSG_ARG_T_LIBRARY    "--library"
1454 @ MSG_ARG_T_LIBPATH    "--library-path"
1455 @ MSG_ARG_T_NOUNDEF    "--no-undefined"
1456 @ MSG_ARG_T_NOWHOLEARC "--no-whole-archive"
1457 @ MSG_ARG_T_OUTPUT     "--output"
1458 @ MSG_ARG_T_RELOCATABLE "--relocatable"
1459 @ MSG_ARG_T_STARTGROUP "--start-group"
1460 @ MSG_ARG_T_STRIP      "--strip-all"
1461 @ MSG_ARG_T_UNDEF     "--undefined"
1462 @ MSG_ARG_T_VERSION   "--version"
1463 @ MSG_ARG_T_WHOLEARC  "--whole-archive"
1464 @ MSG_ARG_T_WRAP      "--wrap"
1465 @ MSG_ARG_T_OPAR      "("
1466 @ MSG_ARG_T_CPAR      ")"

```

1468 # -z guidance-item strings

```

1469 @ MSG_ARG_GUIDE_DELIM    ",: \t"
1470 @ MSG_ARG_GUIDE_NO_ALL  "noall"
1471 @ MSG_ARG_GUIDE_NO_DEFS "nodefs"
1472 @ MSG_ARG_GUIDE_NO_DIRECT "nodirect"
1473 @ MSG_ARG_GUIDE_NO_LAZYLOAD "nolazyload"
1474 @ MSG_ARG_GUIDE_NO_MAPFILE "nomapfile"
1475 @ MSG_ARG_GUIDE_NO_TEXT "notext"
1476 @ MSG_ARG_GUIDE_NO_UNUSED "nounused"

```

1266 # -z assert-deflib strings

```

1267 @ MSG_ARG_ASSERTDEFLIB_MALFORMED "library name malformed: %s"
1268 @ MSG_ARG_ASSERTDEFLIB_FOUND    "dynamic library found on default search path \
1269 (%s): lib%s.so"

```

1478 # Environment variable strings

```

1480 @ MSG_LD_RUN_PATH      "LD_RUN_PATH"
1481 @ MSG_LD_LIBPATH_32   "LD_LIBRARY_PATH_32"
1482 @ MSG_LD_LIBPATH_64   "LD_LIBRARY_PATH_64"
1483 @ MSG_LD_LIBPATH      "LD_LIBRARY_PATH"

1485 @ MSG_LD_NOVERSION_32 "LD_NOVERSION_32"
1486 @ MSG_LD_NOVERSION_64 "LD_NOVERSION_64"
1487 @ MSG_LD_NOVERSION    "LD_NOVERSION"

1489 @ MSG_SGS_SUPPORT_32  "SGS_SUPPORT_32"
1490 @ MSG_SGS_SUPPORT_64 "SGS_SUPPORT_64"
1491 @ MSG_SGS_SUPPORT    "SGS_SUPPORT"

```

1494 # Symbol names

```

1496 @ MSG_SYM_LIBVER_U    "_lib_version"

```

1499 # Mapfile tokens

```

1501 @ MSG_MAP_LOAD      "load"
1502 @ MSG_MAP_NOTE     "note"
1503 @ MSG_MAP_NULL     "null"
1504 @ MSG_MAP_STACK    "stack"
1505 @ MSG_MAP_ADDVERS  "addvers"
1506 @ MSG_MAP_FUNCTION "function"
1507 @ MSG_MAP_DATA     "data"
1508 @ MSG_MAP_COMMON   "common"

```

```

1509 @ MSG_MAP_PARENT     "parent"
1510 @ MSG_MAP_EXTERN     "extern"
1511 @ MSG_MAP_DIRECT     "direct"
1512 @ MSG_MAP_NODIRECT  "nodirect"
1513 @ MSG_MAP_FILTER     "filter"
1514 @ MSG_MAP_AUXILIARY "auxiliary"
1515 @ MSG_MAP_OVERRIDE  "override"
1516 @ MSG_MAP_INTERPOSE "interpose"
1517 @ MSG_MAP_DYNSORT  "dynsort"
1518 @ MSG_MAP_NODYNSORT "nodynsort"

```

```

1520 @ MSG_MAPKW_ALIGN    "ALIGN"
1521 @ MSG_MAPKW_ALLOC    "ALLOC"
1522 @ MSG_MAPKW_ALLOW    "ALLOW"
1523 @ MSG_MAPKW_AMD64_LARGE "AMD64_LARGE"
1524 @ MSG_MAPKW_ASSIGN_SECTION "ASSIGN_SECTION"
1525 @ MSG_MAPKW_AUX     "AUXILIARY"
1526 @ MSG_MAPKW_CAPABILITY "CAPABILITY"
1527 @ MSG_MAPKW_COMMON  "COMMON"
1528 @ MSG_MAPKW_DATA    "DATA"
1529 @ MSG_MAPKW_DEFAULT "DEFAULT"
1530 @ MSG_MAPKW_DEPEND_VERSIONS "DEPEND_VERSIONS"
1531 @ MSG_MAPKW_DIRECT  "DIRECT"
1532 @ MSG_MAPKW_DISABLE "DISABLE"
1533 @ MSG_MAPKW_DYNSORT "DYNSORT"
1534 @ MSG_MAPKW_ELIMINATE "ELIMINATE"
1535 @ MSG_MAPKW_EXECUTE "EXECUTE"
1536 @ MSG_MAPKW_EXPORTED "EXPORTED"
1537 @ MSG_MAPKW_EXTERN  "EXTERN"
1538 @ MSG_MAPKW_FILTER  "FILTER"
1539 @ MSG_MAPKW_FILE_BASENAME "FILE_BASENAME"
1540 @ MSG_MAPKW_FILE_PATH "FILE_PATH"
1541 @ MSG_MAPKW_FILE_OBJNAME "FILE_OBJNAME"
1542 @ MSG_MAPKW_FUNCTION "FUNCTION"
1543 @ MSG_MAPKW_FLAGS    "FLAGS"
1544 @ MSG_MAPKW_GLOBAL   "GLOBAL"
1545 @ MSG_MAPKW_INTERPOSE "INTERPOSE"
1546 @ MSG_MAPKW_HIDDEN   "HIDDEN"
1547 @ MSG_MAPKW_HDR_NOALLOC "HDR_NOALLOC"
1548 @ MSG_MAPKW_HW       "HW"
1549 @ MSG_MAPKW_HW_1     "HW_1"
1550 @ MSG_MAPKW_HW_2     "HW_2"
1551 @ MSG_MAPKW_IS_NAME  "IS_NAME"
1552 @ MSG_MAPKW_IS_ORDER "IS_ORDER"
1553 @ MSG_MAPKW_LOAD_SEGMENT "LOAD_SEGMENT"
1554 @ MSG_MAPKW_LOCAL    "LOCAL"
1555 @ MSG_MAPKW_MACHINE  "MACHINE"
1556 @ MSG_MAPKW_MAX_SIZE "MAX_SIZE"
1557 @ MSG_MAPKW_NOHDR    "NOHDR"
1558 @ MSG_MAPKW_NODIRECT "NODIRECT"
1559 @ MSG_MAPKW_NODYNSORT "NODYNSORT"
1560 @ MSG_MAPKW_NOTE_SEGMENT "NOTE_SEGMENT"
1561 @ MSG_MAPKW_NULL_SEGMENT "NULL_SEGMENT"
1562 @ MSG_MAPKW_OS_ORDER "OS_ORDER"
1563 @ MSG_MAPKW_PADDR    "PADDR"
1564 @ MSG_MAPKW_PARENT   "PARENT"
1565 @ MSG_MAPKW_PHDR_ADD_NULL "PHDR_ADD_NULL"
1566 @ MSG_MAPKW_PLATFORM "PLATFORM"
1567 @ MSG_MAPKW_PROTECTED "PROTECTED"
1568 @ MSG_MAPKW_READ     "READ"
1569 @ MSG_MAPKW_ROUND    "ROUND"
1570 @ MSG_MAPKW_REQUIRE  "REQUIRE"
1571 @ MSG_MAPKW_SEGMENT_ORDER "SEGMENT_ORDER"
1572 @ MSG_MAPKW_SF       "SF"
1573 @ MSG_MAPKW_SF_1     "SF_1"
1574 @ MSG_MAPKW_SINGLETON "SINGLETON"

```



```
1575 @ MSG_MAPKW_SIZE           "SIZE"  
1576 @ MSG_MAPKW_SIZE_SYMBOL    "SIZE_SYMBOL"  
1577 @ MSG_MAPKW_STACK          "STACK"  
1578 @ MSG_MAPKW_SYMBOL_SCOPE    "SYMBOL_SCOPE"  
1579 @ MSG_MAPKW_SYMBOL_VERSION  "SYMBOL_VERSION"  
1580 @ MSG_MAPKW_SYMBOLIC        "SYMBOLIC"  
1581 @ MSG_MAPKW_TYPE            "TYPE"  
1582 @ MSG_MAPKW_VADDR          "VADDR"  
1583 @ MSG_MAPKW_VALUE           "VALUE"  
1584 @ MSG_MAPKW_WRITE           "WRITE"  
  
1587 @ MSG_STR_DTRACE            "PT_SUNWDTRACE"
```

```

*****
38681 Wed Jan  2 21:49:16 2013
new/usr/src/cmd/sgs/libld/common/place.c
3436 relocatable objects also need sloppy relocation
3439 discarded sections shouldn't end up on output lists
*****
_____unchanged_portion_omitted_____

632 /*
633  * Place a section into the appropriate segment and output section.
634  *
635  * entry:
636  *   ofl - File descriptor
637  *   isp - Input section descriptor of section to be placed.
638  *   path_info - NULL, or pointer to Place_path_info buffer initialized
639  *             by ld_place_path_info_init() for the file associated to isp,
640  *             for use in processing entrance criteria with non-empty
641  *             file matching string list (ec_files)
642  *   ident - Section identifier, used to order sections relative to
643  *          others within the output segment.
644  *   alt_os_name - If non-NULL, the name of the output section to place
645  *                isp into. If NULL, input sections go to an output section
646  *                with the same name as the input section.
647  */
648 Os_desc *
649 ld_place_section(Ofl_desc *ofl, Is_desc *isp, Place_path_info *path_info,
650                 int ident, const char *alt_os_name)
651 {
652     Ent_desc     *enp;
653     Sg_desc      *sgp;
654     Os_desc      *osp;
655     Aliste       idxl, iidx;
656     int          os_ndx;
657     Shdr         *shdr = isp->is_shdr;
658     Xword        shflagmask, shflags = shdr->sh_flags;
659     Ifl_desc     *ifl = isp->is_file;
660     char         *oname, *sname;
661     uint_t       onamehash;
662     Boolean      is_ehframe = (isp->is_flags & FLG_IS_EHFRAME) != 0;

664     /*
665      * Define any sections that must be thought of as referenced. These
666      * sections may not be referenced externally in a manner ld(1) can
667      * discover, but they must be retained (ie. not removed by -zignore).
668      */
669     static const Msg RefSecs[] = {
670         MSG_SCN_INIT,          /* MSG_ORIG(MSG_SCN_INIT) */
671         MSG_SCN_FINI,         /* MSG_ORIG(MSG_SCN_FINI) */
672         MSG_SCN_EX_RANGES,    /* MSG_ORIG(MSG_SCN_EX_RANGES) */
673         MSG_SCN_EX_SHARED,    /* MSG_ORIG(MSG_SCN_EX_SHARED) */
674         MSG_SCN_CTORS,        /* MSG_ORIG(MSG_SCN_CTORS) */
675         MSG_SCN_DTORS,        /* MSG_ORIG(MSG_SCN_DTORS) */
676         MSG_SCN_EHFRAME,      /* MSG_ORIG(MSG_SCN_EHFRAME) */
677         MSG_SCN_EHFRAME_HDR,  /* MSG_ORIG(MSG_SCN_EHFRAME_HDR) */
678         MSG_SCN_JCR,          /* MSG_ORIG(MSG_SCN_JCR) */
679         0
680     };

682     DBG_CALL(DBG_sec_in(ofl->ofl_lml, isp));

684     /*
685      * If this section identifies group members, or this section indicates
686      * that it is a member of a group, determine whether the section is
687      * still required.
688      */
689     if ((shflags & SHF_GROUP) || (shdr->sh_type == SHT_GROUP)) {

```

```

690         Group_desc     *gdesc;

692         if ((gdesc = ld_get_group(ofl, isp)) != NULL) {
693             DBG_CALL(DBG_sec_group(ofl->ofl_lml, isp, gdesc));

695             /*
696              * If this group has been replaced by another group,
697              * then this section needs to be discarded.
698              */
699             if (gdesc->gd_oisc) {
700                 isp->is_flags |= FLG_IS_DISCARD;

702                 /*
703                  * Since we're discarding the section, we
704                  * can skip assigning it to an output section.
705                  * The exception is that if the user
706                  * specifies -z relaxreloc, then
707                  * we need to assign the output section so
708                  * that the sloppy relocation logic will have
709                  * the information necessary to do its work.
710                  */
711                 if (!(ofl->ofl_flags1 & FLG_OF1_RLXREL))
712                     return (NULL);
713             }
714         }

716         /*
717          * SHT_GROUP sections can only be included into relocatable
718          * objects.
719          */
720         if (shdr->sh_type == SHT_GROUP) {
721             if ((ofl->ofl_flags & FLG_OF_RELOBJ) == 0) {
722                 isp->is_flags |= FLG_IS_DISCARD;
723                 return (NULL);
724             }
725         }
726     }

728     /*
729      * Always assign SHF_TLS sections to the DATA segment (and then the
730      * PT_TLS embedded inside of there).
731      */
732     if (shflags & SHF_TLS)
733         shflags |= SHF_WRITE;

735     /*
736      * Traverse the entrance criteria list searching for a segment that
737      * matches the input section we have. If an entrance criterion is set
738      * then there must be an exact match. If we complete the loop without
739      * finding a segment, then sgp will be NULL.
740      */
741     sgp = NULL;
742     for (APLIST_TRAVERSE(ofl->ofl_ents, idxl, enp)) {

744         /* Disabled segments are not available for assignment */
745         if (enp->ec_segment->sg_flags & FLG_SG_DISABLED)
746             continue;

748         /*
749          * If an entrance criteria doesn't have any of its fields
750          * set, it will match any section it is tested against.
751          * We set the FLG_EC_CATCHALL flag on these, primarily because
752          * it helps readers of our debug output to understand what
753          * the criteria means --- otherwise the user would just see
754          * that every field is 0, but might not understand the
755          * significance of that.

```

```

756     *
757     * Given that we set this flag, we can use it here as an
758     * optimization to short circuit all of the tests in this
759     * loop. Note however, that if we did not do this, the end
760     * result would be the same --- the empty criteria will sail
761     * past the following tests and reach the end of the loop.
762     */
763     if (enp->ec_flags & FLG_EC_CATCHALL) {
764         sgp = enp->ec_segment;
765         break;
766     }
767
768     if (enp->ec_type && (enp->ec_type != shdr->sh_type))
769         continue;
770     if (enp->ec_attrmask &&
771         /* LINTED */
772         (enp->ec_attrmask & enp->ec_attrbits) !=
773         (enp->ec_attrmask & shflags))
774         continue;
775     if (enp->ec_is_name &&
776         (strcmp(enp->ec_is_name, isp->is_name) != 0))
777         continue;
778
779     if ((alist_nitems(enp->ec_files) > 0) &&
780         ((path_info == NULL) || !eval_ec_files(path_info, enp)))
781         continue;
782
783     /* All entrance criteria tests passed */
784     sgp = enp->ec_segment;
785     break;
786 }
787
788 /*
789  * The final entrance criteria record is a FLG_EC_CATCHALL that points
790  * at the final predefined segment "extra", and this final segment is
791  * tagged FLG_SG_NODISABLE. Therefore, the above loop must always find
792  * a segment.
793  */
794 assert(sgp != NULL);
795
796 /*
797  * Transfer the input section sorting key from the entrance criteria
798  * to the input section. A non-zero value means that the section
799  * will be sorted on this key among the other sections that have a
800  * non-zero key. These sorted sections are collectively placed at the
801  * head of the output section.
802  *
803  * If the sort key is 0, the section is placed after the sorted
804  * sections in the order they are encountered.
805  */
806 isp->is_ordndx = enp->ec_ordndx;
807
808 /* Remember that this entrance criteria has placed a section */
809 enp->ec_flags |= FLG_EC_USED;
810
811 /*
812  * If our caller has supplied an alternative name for the output
813  * section, then we defer to their request. Otherwise, the default
814  * is to use the same name as that of the input section being placed.
815  *
816  * The COMDAT, SHT_GROUP and GNU name translations that follow have
817  * the potential to alter this initial name.
818  */
819 oname = (char *)((alt_os_name == NULL) ? isp->is_name : alt_os_name);
820
821 /*

```

```

822     * Solaris section names may follow the convention:
823     *
824     *     section-name%symbol-name
825     *
826     * This convention has been used to order the layout of sections within
827     * segments for objects built with the compilers -xF option. However,
828     * the final object should not contain individual section headers for
829     * all such input sections, instead the symbol name is stripped from the
830     * name to establish the final output section name.
831     *
832     * This convention has also been followed for COMDAT and sections
833     * identified though SHT_GROUP data.
834     *
835     * Strip out the % from the section name for:
836     *     - Non-relocatable objects
837     *     - Relocatable objects if input section sorting is
838     *       in force for the segment in question.
839     */
840     if (((ofl->ofl_flags & FLG_OF_RELOBJ) == 0) ||
841         (sgp->sg_flags & FLG_SG_IS_ORDER)) {
842         if ((sname = strchr(isp->is_name, '%')) != NULL) {
843             size_t size = sname - isp->is_name;
844
845             if ((oname = libld_malloc(size + 1)) == NULL)
846                 return ((Os_desc *)S_ERROR);
847             (void) strncpy(oname, isp->is_name, size);
848             oname[size] = '\0';
849             DBG_CALL(DBG_sec_redirected(ofl->ofl_lml, isp, oname));
850         }
851     }
852
853     /*
854     * When building relocatable objects, we must not redirect COMDAT
855     * section names into their outputs, such that our output object may
856     * be successfully used as an input object also requiring COMDAT
857     * processing
858     */
859
860     /*
861     #endif /* ! codereview */
862     * GNU section names may follow the convention:
863     *
864     *     .gnu.linkonce.*
865     *
866     * The .gnu.linkonce is a section naming convention that indicates a
867     * COMDAT requirement. Determine whether this section follows the GNU
868     * pattern, and if so, determine whether this section should be
869     * discarded or retained. The comparison of is_name[1] with 'g'
870     * is an optimization to skip using strcmp() too much. This is safe,
871     * because we know the name is not NULL, and therefore must have
872     * at least one character plus a NULL termination.
873     */
874     if ((isp->is_name == oname) && (isp->is_name[1] == 'g') &&
875         if (((ofl->ofl_flags & FLG_OF_RELOBJ) == 0) &&
876             (isp->is_name == oname) && (isp->is_name[1] == 'g') &&
877             (strcmp(MSG_ORIG(MSG_SCN_GNU_LINKONCE), isp->is_name,
878                 MSG_SCN_GNU_LINKONCE_SIZE) == 0)) {
879                 if ((ofl->ofl_flags & FLG_OF_RELOBJ) == 0) {
880                     if ((oname = (char *)gnu_linkonce_sec(isp->is_name)) !=
881                         isp->is_name) {
882                         DBG_CALL(DBG_sec_redirected(ofl->ofl_lml, isp,
883                             oname));
884                     }
885                 }
886                 if ((oname =
887                     (char *)gnu_linkonce_sec(isp->is_name)) != isp->is_name) {
888                     DBG_CALL(DBG_sec_redirected(ofl->ofl_lml, isp, oname));
889                 }
890             }

```

```

883     }
884
885     /*
886     * Explicitly identify this section type as COMDAT. Also,
887     * enable relaxed relocation processing, as this is typically
888     * a requirement with .gnu.linkonce sections.
889     */
890     isp->is_flags |= FLG_IS_COMDAT;
891     if ((ofl->ofl_flags1 & FLG_OF1_NRLXREL) == 0)
892         ofl->ofl_flags1 |= FLG_OF1_RLXREL;
893     DBG_CALL(Dbg_sec_gnu_comdat(ofl->ofl_lml, isp, TRUE,
894         (ofl->ofl_flags1 & FLG_OF1_RLXREL) != 0));
895 }
896
897 /*
898 * GNU section names may also follow the convention:
899 *
900 *     section-name.symbol-name
901 *
902 * This convention is used when defining SHT_GROUP sections of type
903 * COMDAT. Thus, any group processing will have discovered any group
904 * sections, and this identification can be triggered by a pattern
905 * match section names.
906 */
907 if ((isp->is_name == oname) && (isp->is_flags & FLG_IS_COMDAT) &&
908     if (((ofl->ofl_flags & FLG_OF1_RELOBJ) == 0) &&
909         ((isp->is_name == oname) && (isp->is_flags & FLG_IS_COMDAT) &&
910          ((sname = gnu_comdat_sym(ifl, isp)) != NULL)) {
911         size_t size = sname - isp->is_name;
912
913         if ((ofl->ofl_flags & FLG_OF1_RELOBJ) == 0) {
914 #endif /* ! codereview */
915             if ((oname = libld_malloc(size + 1)) == NULL)
916                 return ((Os_desc *)S_ERROR);
917             (void) strncpy(oname, isp->is_name, size);
918             oname[size] = '\0';
919             DBG_CALL(Dbg_sec_redirected(ofl->ofl_lml, isp, oname));
920         }
921 #endif /* ! codereview */
922
923         /*
924         * Enable relaxed relocation processing, as this is
925         * typically a requirement with GNU COMDAT sections.
926         */
927         if ((ofl->ofl_flags1 & FLG_OF1_NRLXREL) == 0) {
928             ofl->ofl_flags1 |= FLG_OF1_RLXREL;
929             DBG_CALL(Dbg_sec_gnu_comdat(ofl->ofl_lml, isp,
930                 FALSE, TRUE));
931         }
932     }
933
934 /*
935 * Assign a hash value now that the output section name has been
936 * finalized.
937 */
938 onamehash = sgs_str_hash(oname);
939
940 /*
941 * Determine if output section ordering is turned on. If so, return
942 * the appropriate ordering index for the section. This information
943 * is derived from the Sg_desc->sg_os_order list that was built
944 * up from the Mapfile.
945 *
946 * A value of 0 for os_ndx means that the section is not sorted
947 * (i.e. is not found in the sg_os_order). The items in sg_os_order
948 * are in the desired sort order, so adding 1 to their alist index

```

```

947     * gives a suitable index for sorting.
948     */
949     os_ndx = 0;
950     if (alist_nitems(sgp->sg_os_order) > 0) {
951         Sec_order *scop;
952
953         for (ALIST_TRAVERSE(sgp->sg_os_order, idx1, scop)) {
954             if (strcmp(scop->sco_secname, oname) == 0) {
955                 scop->sco_flags |= FLG_SGO_USED;
956                 os_ndx = idx1 + 1;
957                 break;
958             }
959         }
960     }
961
962 /*
963 * Mask of section header flags to ignore when matching sections. We
964 * are more strict with relocatable objects, ignoring only the order
965 * flags, and keeping sections apart if they differ otherwise. This
966 * follows the policy that sections in a relative object should only
967 * be merged if their flags are the same, and avoids destroying
968 * information prematurely. For final products however, we ignore all
969 * flags that do not prevent a merge.
970 */
971 shflagmask =
972     (ofl->ofl_flags & FLG_OF1_RELOBJ) ? ALL_SHF_ORDER : ALL_SHF_IGNORE;
973
974 /*
975 * Traverse the input section list for the output section we have been
976 * assigned. If we find a matching section simply add this new section.
977 */
978 iidx = 0;
979 for (APLIST_TRAVERSE(sgp->sg_osdescs, idx1, osp)) {
980     Shdr *os_shdr = osp->os_shdr;
981
982     /*
983     * An input section matches an output section if:
984     * - The ident values match
985     * - The names match
986     * - Not a GROUP section
987     * - Not a DTrace dof section
988     * - Section types match
989     * - Matching section flags, after screening out the
990     *   shflagmask flags.
991     *
992     * Section types are considered to match if any one of
993     * the following are true:
994     * - The type codes are the same
995     * - Both are .eh_frame sections (regardless of type code)
996     * - The input section is COMDAT, and the output section
997     *   is SHT_PROGBITS.
998     */
999     if ((ident == osp->os_identndx) &&
1000         (ident != ld_targ.t_id.id_rel) &&
1001         (onamehash == osp->os_namehash) &&
1002         (shdr->sh_type != SHT_GROUP) &&
1003         (shdr->sh_type != SHT_SUNW_dof) &&
1004         ((shdr->sh_type == os_shdr->sh_type) ||
1005          (is_ehframe && (osp->os_flags & FLG_OS_EHFRAME)) ||
1006          ((shdr->sh_type == SHT_SUNW_COMDAT) &&
1007           (os_shdr->sh_type == SHT_PROGBITS))) &&
1008         ((shflags & ~shflagmask) ==
1009          (os_shdr->sh_flags & ~shflagmask)) &&
1010         (strcmp(oname, osp->os_name) == 0)) {
1011         uintptr_t err;

```

```

1013      /*
1014      * Process any COMDAT section, keeping the first and
1015      * discarding all others.
1016      */
1017      if ((isp->is_flags & FLG_IS_COMDAT) &&
1018          ((err = add_comdat(ofl, osp, isp)) != 1))
1019          return ((Os_desc *)err);

1021      /*
1022      * Set alignment
1023      */
1024      set_addralign(ofl, osp, isp);

1026      /*
1027      * If this section is a non-empty TLS section indicate
1028      * that a PT_TLS program header is required.
1029      */
1030      if ((shflags & SHF_TLS) && shdr->sh_size &&
1031          ((ofl->ofl_flags & FLG_OF_RELOBJ) == 0))
1032          ofl->ofl_flags |= FLG_OF_TLSPHDR;

1034      /*
1035      * Insert the input section descriptor on the proper
1036      * output section descriptor list.
1037      *
1038      * If this segment requires input section ordering,
1039      * honor any mapfile specified ordering for otherwise
1040      * unordered sections by setting the mapfile_sort
1041      * argument of os_attach_isp() to True.
1042      */

1044      if (os_attach_isp(ofl, osp, isp,
1045                      (sgp->sg_flags & FLG_SG_IS_ORDER) != 0) == 0)
1046          return ((Os_desc *)S_ERROR);

1048      /*
1049      * If this input section and file is associated to an
1050      * artificially referenced output section, make sure
1051      * they are marked as referenced also. This ensures
1052      * that this input section and file isn't eliminated
1053      * when -zignore is in effect.
1054      *
1055      * See -zignore comments when creating a new output
1056      * section below.
1057      */
1058      if (((ifl &&
1059           (ifl->ifl_flags & FLG_IF_IGNORE)) || (DBG_ENABLED) &&
1060          (osp->os_flags & FLG_OS_SECTREF)) {
1061          isp->is_flags |= FLG_IS_SECTREF;
1062          if (ifl)
1063              ifl->ifl_flags |= FLG_IF_FILEREF;
1064      }

1066      DBG_CALL(Debug_sec_added(ofl->ofl_lml, osp, sgp));
1067      return (osp);
1068  }

1070      /*
1071      * Do we need to worry about section ordering?
1072      */
1073      if (os_ndx) {
1074          if (osp->os_ordndx) {
1075              if (os_ndx < osp->os_ordndx)
1076                  /* insert section here. */
1077                  break;
1078              else {

```

```

1079          idx = idxl + 1;
1080          continue;
1081      } else {
1082          /* insert section here. */
1083          break;
1084      }
1085      } else if (osp->os_ordndx) {
1086          idx = idxl + 1;
1087          continue;
1088      }
1089  }

1091      /*
1092      * If the new sections identifier is less than that of the
1093      * present input section we need to insert the new section
1094      * at this point.
1095      */
1096      if (ident < osp->os_identndx)
1097          break;

1099      idx = idxl + 1;
1100  }

1102      /*
1103      * We are adding a new output section. Update the section header
1104      * count and associated string size.
1105      *
1106      * If the input section triggering this output section has been marked
1107      * for discard, and if no other non-discarded input section comes along
1108      * to join it, then we will over count. We cannot know if this will
1109      * happen or not until all input is seen. Set FLG_OF_AJDOSCNT to
1110      * trigger a final count readjustment.
1111      */
1112      if (isp->is_flags & FLG_IS_DISCARD)
1113          ofl->ofl_flags |= FLG_OF_AJDOSCNT;
1114      ofl->ofl_shdrcnt++;
1115      if (st_insert(ofl->ofl_shdrsttab, oname) == -1)
1116          return ((Os_desc *)S_ERROR);

1118      /*
1119      * Create a new output section descriptor.
1120      */
1121      if ((osp = libld_calloc(sizeof (Os_desc), 1)) == NULL)
1122          return ((Os_desc *)S_ERROR);
1123      if ((osp->os_shdr = libld_calloc(sizeof (Shdr), 1)) == NULL)
1124          return ((Os_desc *)S_ERROR);

1126      /*
1127      * Convert COMDAT section to PROGBITS as this the first section of the
1128      * output section. Save any COMDAT section for later processing, as
1129      * additional COMDAT sections that match this section need discarding.
1130      */
1131      if ((shdr->sh_type == SHT_SUNW_COMDAT) &&
1132          ((shdr = isp_convert_type(isp, SHT_PROGBITS)) == NULL))
1133          return ((Os_desc *)S_ERROR);
1134      if ((isp->is_flags & FLG_IS_COMDAT) &&
1135          (add_comdat(ofl, osp, isp) == S_ERROR))
1136          return ((Os_desc *)S_ERROR);

1138      if (is_ehframe) {
1139          /*
1140          * Executable or sharable objects can have at most a single
1141          * .eh_frame section. Detect attempts to create more than
1142          * one. This occurs if the input sections have incompatible
1143          * attributes.
1144          */

```

```

1145     if ((ofl->ofl_flags & FLG_OF_EHFRAME) &&
1146         !(ofl->ofl_flags & FLG_OF_RELOBJ)) {
1147         eh_frame_muldef(ofl, isp);
1148         return ((Os_desc *)S_ERROR);
1149     }
1150     ofl->ofl_flags |= FLG_OF_EHFRAME;
1151
1152     /*
1153     * For .eh_frame sections, we always set the type to be the
1154     * type specified by the ABI. This allows .eh_frame sections
1155     * of type SHT_PROGBITS to be correctly merged with .eh_frame
1156     * sections of the ABI-defined type (e.g. SHT_AMD64_UNWIND),
1157     * with the output being of the ABI-defined type.
1158     */
1159     osp->os_shdr->sh_type = ld_targ_t.m.m_sht_unwind;
1160 } else {
1161     osp->os_shdr->sh_type = shdr->sh_type;
1162 }
1163
1164 osp->os_shdr->sh_flags = shdr->sh_flags;
1165 osp->os_shdr->sh_entsize = shdr->sh_entsize;
1166 osp->os_name = oname;
1167 osp->os_namehash = onamehash;
1168 osp->os_ordndx = os_ndx;
1169 osp->os_sgdesc = sgp;
1170 if (is_ehframe)
1171     osp->os_flags |= FLG_OS_EHFRAME;
1172
1173 if (ifl && (shdr->sh_type == SHT_PROGBITS)) {
1174     /*
1175     * Try to preserve the intended meaning of sh_link/sh_info.
1176     * See the translate_link() in update.c.
1177     */
1178     osp->os_shdr->sh_link = shdr->sh_link;
1179     if (shdr->sh_flags & SHF_INFO_LINK)
1180         osp->os_shdr->sh_info = shdr->sh_info;
1181 }
1182
1183 /*
1184 * When -zignore is in effect, user supplied sections and files that are
1185 * not referenced from other sections, are eliminated from the object
1186 * being produced. Some sections, although unreferenced, are special,
1187 * and must not be eliminated. Determine if this new output section is
1188 * one of those special sections, and if so mark it artificially as
1189 * referenced. Any input section and file associated to this output
1190 * section is also be marked as referenced, and thus won't be eliminated
1191 * from the final output.
1192 */
1193 if (ifl && ((ofl->ofl_flags1 & FLG_OF1_IGNORE) || DBG_ENABLED)) {
1194     const Msg *refsec;
1195
1196     for (refsec = RefSecs; *refsec; refsec++) {
1197         if (strcmp(osp->os_name, MSG_ORIG(*refsec)) == 0) {
1198             osp->os_flags |= FLG_OS_SECTREF;
1199
1200             if ((ifl->ifl_flags & FLG_IF_IGNORE) ||
1201                 DBG_ENABLED) {
1202                 isp->is_flags |= FLG_IS_SECTREF;
1203                 ifl->ifl_flags |= FLG_IF_FILEREF;
1204             }
1205             break;
1206         }
1207     }
1208 }
1209
1210 /*

```

```

1211     * Sections of type SHT_GROUP are added to the ofl->ofl_osgroups list,
1212     * so that they can be updated as a group later.
1213     */
1214     if ((shdr->sh_type == SHT_GROUP) &&
1215         ((isp->is_flags & FLG_IS_DISCARD) == 0) &&
1216         #endif /* !codereview */
1217         (aplist_append(&ofl->ofl_osgroups, osp,
1218             AL_CNT_OF1_OSGROUPS) == NULL))
1219         return ((Os_desc *)S_ERROR);
1220
1221     /*
1222     * If this section is a non-empty TLS section indicate that a PT_TLS
1223     * program header is required.
1224     */
1225     if ((shflags & SHF_TLS) && shdr->sh_size &&
1226         ((ofl->ofl_flags & FLG_OF_RELOBJ) == 0))
1227         ofl->ofl_flags |= FLG_OF_TLSPHDR;
1228
1229     /*
1230     * If a non-allocatable section is going to be put into a loadable
1231     * segment then turn on the allocate bit for this section and warn the
1232     * user that we have done so. This could only happen through the use
1233     * of a mapfile.
1234     */
1235     if ((sgp->sg_phdr.p_type == PT_LOAD) &&
1236         ((osp->os_shdr->sh_flags & SHF_ALLOC) == 0)) {
1237         ld_printf(ofl, ERR_WARNING, MSG_INTL(MSG_SCN_NONALLOC),
1238             ofl->ofl_name, osp->os_name, sgp->sg_name);
1239         osp->os_shdr->sh_flags |= SHF_ALLOC;
1240     }
1241
1242     /*
1243     * Retain this sections identifier for future comparisons when placing
1244     * a section (after all sections have been processed this variable will
1245     * be used to hold the sections symbol index as we don't need to retain
1246     * the identifier any more).
1247     */
1248     osp->os_identndx = ident;
1249
1250     /*
1251     * Set alignment.
1252     */
1253     set_addralign(ofl, osp, isp);
1254
1255     if (os_attach_isp(ofl, osp, isp, 0) == 0)
1256         return ((Os_desc *)S_ERROR);
1257
1258     DBG_CALL(DBG_sec_created(ofl->ofl_lml, osp, sgp));
1259
1260     /*
1261     * Insert the new section at the offset given by iidx. If no position
1262     * for it was identified above, this will be index 0, causing the new
1263     * section to be prepended to the beginning of the section list.
1264     * Otherwise, it is the index following the section that was identified.
1265     */
1266     if (aplist_insert(&sgp->sg_osdescs, osp, AL_CNT_SG_OSDESC,
1267         iidx) == NULL)
1268         return ((Os_desc *)S_ERROR);
1269     return (osp);
1270 }

```

new/usr/src/cmd/sgs/librtld_db/demo/Makefile.com

1

```
*****
2434 Wed Jan 2 21:49:16 2013
new/usr/src/cmd/sgs/librtld_db/demo/Makefile.com
3440 librtld_db demos need warnings gagged
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 # Copyright (c) 1995, 2010, Oracle and/or its affiliates. All rights reserved.

23 PROG=          rdb

25 # DEMO DELETE START
26 include        ../../../../Makefile.cmd
27 # DEMO DELETE END

29 MACH:sh=       uname -p

31 CFLAGS +=      $(DEMOFLAGS)

33 COMSRC=        bpt.c dis.c main.c ps.c gram.c lex.c globals.c help.c \
34                utils.c maps.c syms.c callstack.c disasm.c
35 M_SRC=         regs.c m_utils.c

37 BLTSRC=        gram.c lex.c
38 BLTHDR=        gram.h

40 # DEMO DELETE START
41 ONDLIBDIR=     /opt/SUNWworld/lib

43 # DEMO DELETE END
44 OBJDIR=        objs
45 OBJS =         $(COMSRC:%.c=$(OBJDIR)/%.o) $(M_SRC:%.c=$(OBJDIR)/%.o)

47 SRCS =         $(COMSRC:%=./common/%) $(M_SRC)

49 MV =          mv

51 .PARALLEL:    $(OBJS)

53 CPPFLAGS=      -I../common -I. $(CPPFLAGS.master)
54 LDLIBS +=      -lrtld_db -lself -ll -ly

56 CLEANFILES += $(BLTSRC) $(BLTHDR) simp libsub.so.1

58 # DEMO DELETE START
59 # The following lint error suppression definitions are to remove lex errors
60 # we have no control over.
61 LINTERRS =     -erroff=E_NAME_DEF_NOT_USED2 \
```

new/usr/src/cmd/sgs/librtld_db/demo/Makefile.com

2

```
62                -erroff=E_FUNC_RET_ALWAYS_IGNORE2 \
63                -erroff=E_FUNC_RET_MAYBE_IGNORED2 \
64                -erroff=E_BLOCK_DECL_UNUSED \
65                -erroff=E_EQUALITY_NOT_ASSIGNMENT
66 LINTFLAGS +=   $(LDLIBS) -L../../${MACH} $(LINTERRS)
67 LINTFLAGS64 += $(LDLIBS) -L../../${MACH} $(LINTERRS)
68 CLEANFILES += $(LINTOUT)

70 CERRWARN +=    -_gcc=-Wno-parentheses
71 CERRWARN +=    -_gcc=-Wno-unused-label
72 CERRWARN +=    -_gcc=-Wno-unused-variable
73 #endif /* ! codereview */
74 # DEMO DELETE END

76 test-sparc=    test-sparc-regs
77 test-i386=
78 TESTS=         test-maps test-breaks test-steps test-plt_skip \
79                test-object-padding $(test-$(MACH))

81 # DEMO DELETE START
82 ROOTONLDBIN=   $(ROOT)/opt/SUNWworld/bin
83 ROOTONLDBINPROG= $(PROG:%=$(ROOTONLDBIN)/%)
84 ROOTONLDBINPROG64= $(PROG:%=$(ROOTONLDBIN)/$(MACH64)/%)

86 FILEMODE=     0755
87 # DEMO DELETE END
```

```

*****
4441 Wed Jan  2 21:49:17 2013
new/usr/src/cmd/sgs/librtld_db/demo/common/dis.c
3440 librtld_db demos need warnings gagged
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright (c) 1995, 2010, Oracle and/or its affiliates. All rights reserved.
24 */

26 #include <stdio.h>
27 #include <stdlib.h>
28 #include <unistd.h>
29 #include <fcntl.h>
30 #include <string.h>
31 #include <errno.h>
32 #include <sys/types.h>
33 #include <sys/signal.h>
34 #include <sys/fault.h>
35 #include <sys/syscall.h>
36 #include <procfs.h>
37 #include <sys/auxv.h>
38 #include <libelf.h>
39 #include <sys/param.h>
40 #include <stdarg.h>

42 #include "rdb.h"
43 #include "disasm.h"

45 /*
46  * I don't like this global but it's a work-around for the
47  * poor disassemble interface for now.
48  */
49 static struct ps_prochandle *cur_ph;

51 /*
52  * This routine converts 'address' into it's closest symbol
53  * representation.
54  *
55  * The following flags are used to effect the output:
56  *
57  *   FLG_PAP_SONAME
58  *       embed the SONAME in the symbol name
59  *   FLG_PAP_NOHEXNAME
60  *       if no symbol found return a null string
61  *       If this flag is not set return a string displaying

```

```

62  *       the 'hex' value of address.
63  *       FLG_PAP_PLTDECOM
64  *       decompose the PLT symbol if possible
65  */
66 char *
67 print_address_ps(struct ps_prochandle *ph, ulong_t address, unsigned flags)
68 {
69     static char    buf[256];
70     GElf_Sym      sym;
71     char          *str;
72     ulong_t       val;

74     if (addr_to_sym(ph, address, &sym, &str) == RET_OK) {
75         map_info_t *mip;
76         ulong_t    pltbase;

78         if (flags & FLG_PAP_SONAME) {
79             /*
80              * Embed SOName in symbol name
81              */
82             if ((mip = addr_to_map(ph, address)) != 0) {
83                 if (mip = addr_to_map(ph, address)) {
84                     (void) strcpy(buf, mip->mi_name);
85                     (void) strcat(buf, "-");
86                 } else
87                     (void) sprintf(buf, "0x%08lx:", address);
88             } else
89                 buf[0] = '\0';

90             if ((flags & FLG_PAP_PLTDECOM) &&
91                 (pltbase = is_plt(ph, address)) != 0) {
92                 rd_plt_info_t rp;
93                 pstatus_t    pstatus;

95                 if (pread(ph->pp_statusfd, &pstatus,
96                     sizeof (pstatus), 0) == -1)
97                     perr("pap: reading pstatus");

99                 if (rd_plt_resolution(ph->pp_rap, address,
100                    pstatus.pr_lwp.pr_lwpid, pltbase,
101                    &rp) == RD_OK) {
102                     if (rp.pi_flags & RD_FLG_PI_PLTBOUND) {
103                         GElf_Sym _sym;
104                         char      *_str;

106                         if (addr_to_sym(ph, rp.pi_baddr,
107                            &_sym, &_str) == RET_OK) {
108                             (void) snprintf(buf, 256,
109                                "%s0x%lx:plt(%s)",
110                                buf, address, _str);
111                             return (buf);
112                         }
113                     }
114                 }
115                 val = sym.st_value;
116                 (void) snprintf(buf, 256, "%s0x%lx:plt(unbound)+0x%lx",
117                    buf, address, address - val);
118                 return (buf);
119             } else {
121                 val = sym.st_value;

123                 if (val < address)
124                     (void) snprintf(buf, 256, "%s%s+0x%lx", buf,
125                        str, address - val);
126                 else

```



```
127             (void) snprintf(buf, 256, "%s%s", buf, str);
128             return (buf);
129         }
130     } else {
131         if (flags & FLG_PAP_NOHEXNAME)
132             buf[0] = '\0';
133         else
134             (void) sprintf(buf, "0x%x", address);
135         return (buf);
136     }
137 }
```

unchanged_portion_omitted_

```

*****
87457 Wed Jan  2 21:49:17 2013
new/usr/src/cmd/sgs/packages/common/SUNWorld-README
3436 relocatable objects also need sloppy relocation
3439 discarded sections shouldn't end up on output lists
*****
1 #
2 # Copyright (c) 1996, 2010, Oracle and/or its affiliates. All rights reserved.
3 #
4 # CDDL HEADER START
5 #
6 # The contents of this file are subject to the terms of the
7 # Common Development and Distribution License (the "License").
8 # You may not use this file except in compliance with the License.
9 #
10 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
11 # or http://www.opensolaris.org/os/licensing.
12 # See the License for the specific language governing permissions
13 # and limitations under the License.
14 #
15 # When distributing Covered Code, include this CDDL HEADER in each
16 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
17 # If applicable, add the following below this CDDL HEADER, with the
18 # fields enclosed by brackets "[]" replaced with your own identifying
19 # information: Portions Copyright [yyyy] [name of copyright owner]
20 #
21 # CDDL HEADER END
22 #
23 # Note: The contents of this file are used to determine the versioning
24 # information for the SGS toolset. The number of CRs listed in
25 # this file must grow monotonically, or the SGS version will
26 # move backwards, causing a great deal of confusion. As such,
27 # CRs must never be removed from this file. See
28 # libconv/common/bld_vernote.ksh, and bug#4519569 for more
29 # details on SGS versioning.
30 #
31 -----
32 SUNWorld - link-editors development package.
33 -----

35 The SUNWorld package is an internal development package containing the
36 link-editors and some related tools. All components live in the OSNET
37 source base, but not all components are delivered as part of the normal
38 OSNET consolidation. The intent of this package is to provide access
39 to new features/bugfixes before they become generally available.

41 General link-editor information can be found:

43 http://linkers.central/
44 http://linkers.sfbay/ (also known as linkers.eng)

46 Comments and Questions:

48 Contact Rod Evans, Ali Bahrami, and/or Seizo Sakurai.

50 Warnings:

52 The postremove script for this package employs /usr/sbin/static/mv,
53 and thus, besides the common core dependencies, this package also
54 has a dependency on the SUNWsutl package.

56 Patches:

58 If the patch has been made official, you'll find it in:

60 http://sunsolve.east/cgi/show.pl?target=patches/os-patches

```

```

62 If it hasn't been released, the patch will be in:

64 /net/sunsoftpatch/patches/temporary

66 Note, any patches logged here refer to the temporary ("T") name, as we
67 never know when they're made official, and although we try to keep all
68 patch information up-to-date the real status of any patch can be
69 determined from:

71 http://sunsoftpatch.eng

73 If it has been obsoleted, the patch will be in:
74
75 /net/on${RELEASE}-patch/on${RELEASE}/patches/${MACH}/obsolete

78 History:

80 Note, starting after Solaris 10, letter codes in parenthesis may
81 be found following the bug synopsis. Their meanings are as follows:

83 (D) A documentation change accompanies the implementation change.
84 (P) A packaging change accompanies the implementation change.

86 In all cases, see the implementation bug report for details.

88 The following bug fixes exist in the OSNET consolidation workspace
89 from which this package is created:

91 -----
92 Solaris 8
93 -----
94 Bugid Risk Synopsis
95 -----
96 4225937 i386 linker emits sparc specific warning messages
97 4215164 shf_order flag handling broken by fix for 4194028.
98 4215587 using ld and the -r option on solaris 7 with compiler option -xarch=v9
99 causes link errors.
100 4234657 103627-08 breaks purify 4.2 (plt padding should not be enabled for
101 32-bit)
102 4235241 dbx no longer gets dlclose notification.
103 -----
104 All the above changes are incorporated in the following patches:
105 Solaris/SunOS 5.7_sparc patch 106950-05 (never released)
106 Solaris/SunOS 5.7_x86 patch 106951-05 (never released)
107 Solaris/SunOS 5.6_sparc patch 107733-02 (never released)
108 Solaris/SunOS 5.6_x86 patch 107734-02
109 -----
110 4248290 inetd dumps core upon bootup - failure in dlclose() logic.
111 4238071 dlopen() leaks while descriptors under low memory conditions
112 -----
113 All the above changes are incorporated in the following patches:
114 Solaris/SunOS 5.7_sparc patch 106950-06
115 Solaris/SunOS 5.7_x86 patch 106951-06
116 Solaris/SunOS 5.6_sparc patch 107733-03 (never released)
117 Solaris/SunOS 5.6_x86 patch 107734-03
118 -----
119 4267980 INITFIRST flag of the shard object could be ignored.
120 -----
121 All the above changes plus:
122 4238973 fix for 4121152 affects linking of Ada objects
123 4158744 patch 103627-02 causes core when RPATH has blank entry and
124 dlopen/dlclose is used
125 are incorporated in the following patches:
126 Solaris/SunOS 5.5.1_sparc patch 103627-12 (never released)

```

```

127 Solaris/SunOS 5.5.1_x86 patch 103628-11
128 -----
129 4256518 miscalculated calloc() during dlclose/tsorting can result in segv
130 4254171 DT_SPARC_REGISTER has invalid value associated with it.
131 -----
132 All the above changes are incorporated in the following patches:
133 Solaris/SunOS 5.7_sparc patch 106950-07
134 Solaris/SunOS 5.7_x86 patch 106951-07
135 Solaris/SunOS 5.6_sparc patch 107733-04 (never released)
136 Solaris/SunOS 5.6_x86 patch 107734-04
137 -----
138 4293159 ld needs to combine sections with and without SHF_ORDERED flag(comdat)
139 4292238 linking a library which has a static char ptr invokes mprotect() call
140 -----
141 All the above changes except for:
142 4256518 miscalculated calloc() during dlclose/tsorting can result in segv
143 4254171 DT_SPARC_REGISTER has invalid value associated with it.
144 plus:
145 4238973 fix for 4121152 affects linking of Ada objects
146 4158744 patch 103627-02 causes core when RPATH has blank entry and
147 dlopen/dlclose is used
148 are incorporated in the following patches:
149 Solaris/SunOS 5.5.1_sparc patch 103627-13
150 Solaris/SunOS 5.5.1_x86 patch 103628-12
151 -----
152 All the above changes are incorporated in the following patches:
153 Solaris/SunOS 5.7_sparc patch 106950-08
154 Solaris/SunOS 5.7_x86 patch 106951-08
155 Solaris/SunOS 5.6_sparc patch 107733-05
156 Solaris/SunOS 5.6_x86 patch 107734-05
157 -----
158 4295613 COMMON symbol resolution can be incorrect
159 -----
160 All the above changes plus:
161 4238973 fix for 4121152 affects linking of Ada objects
162 4158744 patch 103627-02 causes core when RPATH has blank entry and
163 dlopen/dlclose is used
164 are incorporated in the following patches:
165 Solaris/SunOS 5.5.1_sparc patch 103627-14
166 Solaris/SunOS 5.5.1_x86 patch 103628-13
167 -----
168 All the above changes plus:
169 4351197 nfs performance problem by 103627-13
170 are incorporated in the following patches:
171 Solaris/SunOS 5.5.1_sparc patch 103627-15
172 Solaris/SunOS 5.5.1_x86 patch 103628-14
173 -----
174 All the above changes are incorporated in the following patches:
175 Solaris/SunOS 5.7_sparc patch 106950-09
176 Solaris/SunOS 5.7_x86 patch 106951-09
177 Solaris/SunOS 5.6_sparc patch 107733-06
178 Solaris/SunOS 5.6_x86 patch 107734-06
179 -----
180 4158971 increase the default segment alignment for i386 to 64k
181 4064994 Add an $ISALIST token to those understood by the dynamic linker
182 xxxxxxxx ia64 common code putback
183 4239308 LD_DEBUG busted for sparc machines
184 4239008 Support MAP_ANON
185 4238494 link-auditing extensions required
186 4232239 R_SPARC_LOX10 truncates field
187 4231722 R_SPARC_UA* relocations are busted
188 4235514 R_SPARC_OLO10 relocation fails
189 4244025 sgsmsg update
190 4239281 need to support SECREL relocations for ia64
191 4253751 ia64 linker must support PT_IA_64_UNWIND tables
192 4259254 dllopen mistakenly closes fd 0 (stdin) under certain error conditions

```

```

193 4260872 libelf hangs when libthread present
194 4224569 linker core dumping when profiling specified
195 4270937 need mechanism to suppress ld.so.1's use of a default search path.
196 1050476 ld.so to permit configuration of search path
197 4273654 filtee processing using $ISALIST could be optimized
198 4271860 get MERCED cruft out of elf.h
199 4248991 Dynamic loader (via PLT) corrupts register G4
200 4275754 cannot mmap file: Resource temporarily unavailable
201 4277689 The linker can not handle relocation against MOVE tabl
202 4270766 atexit processing required on dlclose().
203 4279229 Add a "release" token to those understood by the dynamic linker
204 4215433 ld can bus error when insufficient disc space exists for output file
205 4285571 Pssst, want some free disk space? ld's miscalculating.
206 4286236 ar gives confusing "bad format" error with a null .stab section
207 4286838 ld.so.1 can't handle a no-bits segment
208 4287364 ld.so.1 runtime configuration cleanup
209 4289573 disable linking of ia64 binaries for Solaris8
210 4293966 crle(1)'s default directories should be supplied
211 -----
213 -----
214 Solaris 8 600 (1st Q-update - s28u1)
215 -----
216 Bugid Risk Synopsis
217 =====
218 4309212 dlsym can't find symbol
219 4311226 rejection of preloading in secure apps is inconsistent
220 4312449 dlclose: invalid deletion of dependency can occur using RTLD_GLOBAL
221 -----
222 All the above changes are incorporated in the following patches:
223 Solaris/SunOS 5.8_sparc patch 109147-01
224 Solaris/SunOS 5.8_x86 patch 109148-01
225 Solaris/SunOS 5.7_sparc patch 106950-10
226 Solaris/SunOS 5.7_x86 patch 106951-10
227 Solaris/SunOS 5.6_sparc patch 107733-07
228 Solaris/SunOS 5.6_x86 patch 107734-07
229 -----
231 -----
232 Solaris 8 900 (2nd Q-update - s28u2)
233 -----
234 Bugid Risk Synopsis
235 =====
236 4324775 non-PIC code & -zcombreloc don't mix very well...
237 4327653 run-time linker should preload tables it will process (madvise)
238 4324324 shared object code can be referenced before .init has fired
239 4321634 .init firing of multiple INITFIRST objects can fail
240 -----
241 All the above changes are incorporated in the following patches:
242 Solaris/SunOS 5.8_sparc patch 109147-03
243 Solaris/SunOS 5.8_x86 patch 109148-03
244 Solaris/SunOS 5.7_sparc patch 106950-11
245 Solaris/SunOS 5.7_x86 patch 106951-11
246 Solaris/SunOS 5.6_sparc patch 107733-08
247 Solaris/SunOS 5.6_x86 patch 107734-08
248 -----
249 4338812 crle(1) omits entries in the directory cache
250 4341496 RFE: provide a static version of /usr/bin/crle
251 4340878 rtdld should treat $ORIGIN like LD_LIBRARY_PATH in security issues
252 -----
253 All the above changes are incorporated in the following patches:
254 Solaris/SunOS 5.8_sparc patch 109147-04
255 Solaris/SunOS 5.8_x86 patch 109148-04
256 Solaris/SunOS 5.7_sparc patch 106950-12
257 Solaris/SunOS 5.7_x86 patch 106951-12
258 -----

```

```

259 4349563 auxiliary filter error handling regression introduced in 4165487
260 4355795 ldd -r now gives "displacement relocated" warnings
261 -----
262 All the above changes are incorporated in the following patches:
263 Solaris/SunOS 5.7_sparc patch 106950-13
264 Solaris/SunOS 5.7_x86 patch 106951-13
265 Solaris/SunOS 5.6_sparc patch 107733-09
266 Solaris/SunOS 5.6_x86 patch 107734-09
267 -----
268 4210412 versioning a static executable causes ld to core dump
269 4219652 Linker gives misleading error about not finding main (xarch=v9)
270 4103449 ld command needs a command line flag to force 64-bits
271 4187211 problem with RDISP32 linking in copy-relocated objects
272 4287274 dladdr, dlinfo do not provide the full path name of a shared object
273 4297563 dlclose still does not remove all objects.
274 4250694 rtdld_db needs a new auxvec entry
275 4235315 new features for rtdld_db (DT_CHECKSUM, dynamic linked .o files)
276 4303609 64bit libelf.so.1 does not properly implement elf_hash()
277 4310901 su.static fails when OSNet build with lazy-loading
278 4310324 elf_errno() causes Bus Error(coredump) in 64-bit multithreaded programs
279 4306415 ld core dump
280 4316531 BCP: possible failure with dlclose/_preexec_exit_handlers
281 4313765 LD_BREADTH should be shot
282 4318162 crle uses automatic strings in putenv.
283 4255943 Description of -t option incomplete.
284 4322528 sgs message test infrastructure needs improvement
285 4239213 Want an API to obtain linker's search path
286 4324134 use of extern mapfile directives can contribute unused symbols
287 4322581 ELF data structures could be layed out more efficiently...
288 4040628 Unnecessary section header symbols should be removed from .dynsym
289 4300018 rtdld: bindlock should be freed before calling call_fini()
290 4336102 dlclose with non-deletable objects can mishandle dependencies
291 4329785 mixing of SHT_SUNW_COMDAT & SHF_ORDERED causes ld to seg fault
292 4334617 COPY relocations should be produces for references to .bss symbols
293 4248250 Relcoation of local ABS symbols incorrect
294 4335801 For complimentary alignments eliminate ld: warning: symbol 'll'
295 has differing a
296 4336980 ld.so.1 relative path processing revisited
297 4243097 dlerror(3DL) is not affected by setlocale(3C).
298 4344528 dump should remove -D and -l usage message
299 xxxxxxx enable LD_ALTEXEC to access alternate link-editor
300 -----
301 All the above changes are incorporated in the following patches:
302 Solaris/SunOS 5.8_sparc patch 109147-06
303 Solaris/SunOS 5.8_x86 patch 109148-06
304 -----
306 -----
307 Solaris 8 101 (3rd Q-update - s28u3)
308 -----
309 Bugid Risk Synopsis
310 =====
311 4346144 link-auditing: plt_tracing fails if LA_SYMB_NOPLTENTER given after
312 being bound
313 4346001 The ld should support mapfile syntax to generate PT_SUNWSTACK segment
314 4349137 rtdld_db: A third fallback method for locating the linkmap
315 4343417 dladdr interface information inadequate
316 4343801 RFE: crle(1): provide option for updating configuration files
317 4346615 ld.so.1 attempting to open a directory gives: No such device
318 4352233 crle should not honor umask
319 4352330 LD_PRELOAD cannot use absolute path for privileged program
320 4357805 RFE: man page for ld(1) does not document all -z or -B options in
321 Solaris 8 9/00
322 4358751 ld.so.1: LD_XXX environ variables and LD_FLAGS should be synchronized.
323 4358862 link editors should reference "64" symlinks instead of sparcv9 (ia64).
324 4356879 PLTs could use faster code sequences in some cases

```

```

325 4367118 new fast baplt's fail when traversed twice in threaded application
326 4366905 Need a way to determine path to a shared library
327 4351197 nfs performance problem by 103627-13
328 4367405 LD_LIBRARY_PATH_64 not being used
329 4354500 SHF_ORDERED ordered sections does not properly sort sections
330 4369068 ld(1)'s weak symbol processing is inefficient (slow and doesn't scale).
331 -----
332 All the above changes are incorporated in the following patches:
333 Solaris/SunOS 5.8_sparc patch 109147-07
334 Solaris/SunOS 5.8_x86 patch 109148-07
335 Solaris/SunOS 5.7_sparc patch 106950-14
336 Solaris/SunOS 5.7_x86 patch 106951-14
337 -----
339 -----
340 Solaris 8 701 (5th Q-update - s28u5)
341 -----
342 Bugid Risk Synopsis
343 =====
344 4368846 ld(1) fails to version some interfaces given in a mapfile
345 4077245 dump core dump on null pointer.
346 4372554 elfdump should demangle symbols (like nm, dump)
347 4371114 dlclose may unmap a promiscuous object while it's still in use.
348 4204447 elfdump should understand SHN_AFTER/SHN_BEGIN macro
349 4377941 initialization of interposers may not occur
350 4381116 ldd/ld.so.1 could aid in detecting unused dependencies
351 4381783 dlopen/dlclose of a libCrun+libthread can dump core
352 4385402 linker & run-time linker must support GABI ELF updates
353 4394698 ld.so.1 does not process DF_SYMBOLIC - not GABI conforming
354 4394212 the link editor quietly ignores missing support libraries
355 4390308 ld.so.1 should provide more flexibility LD_PRELOAD'ing 32-bit/64-bit
356 objects
357 4401232 crle(1) could provide better flexibility for alternatives
358 4401815 fix misc nits in debugging output...
359 4402861 cleanup /usr/demo/link_audit & /usr/tmp/librtdld_db demo source code...
360 4393044 elfdump should allow raw dumping of sections
361 4413168 SHF_ORDERED bit causes linker to generate a separate section
362 -----
363 All the above changes are incorporated in the following patches:
364 Solaris/SunOS 5.8_sparc patch 109147-08
365 Solaris/SunOS 5.8_x86 patch 109148-08
366 -----
367 4452202 Typos in <sys/link.h>
368 4452220 dump doesn't support RUNPATH
369 -----
370 All the above changes are incorporated in the following patches:
371 Solaris/SunOS 5.8_sparc patch 109147-09
372 Solaris/SunOS 5.8_x86 patch 109148-09
373 -----
375 -----
376 Solaris 8 1001 (6th Q-update - s28u6)
377 -----
378 Bugid Risk Synopsis
379 =====
380 4421842 fixups in SHT_GROUP processing required...
381 4450433 problem with liblddbg output on -Dsection,detail when
382 processing SHF_LINK_ORDER
383 -----
384 All the above changes are incorporated in the following patches:
385 Solaris/SunOS 5.8_sparc patch 109147-10
386 Solaris/SunOS 5.8_x86 patch 109148-10
387 Solaris/SunOS 5.7_sparc patch 106950-15
388 Solaris/SunOS 5.7_x86 patch 106951-15
389 -----
390 4463473 pldd showing wrong output

```

```

391 -----
392 All the above changes are incorporated in the following patches:
393     Solaris/SunOS 5.8_sparc      patch 109147-11
394     Solaris/SunOS 5.8_x86       patch 109148-11
395 -----
397 -----
398 Solaris 8 202 (7th Q-update - s28u7)
399 -----
400 Bugid   Risk Synopsis
401 -----
402 4488954 ld.so.1 reuses same buffer to send unmapping range to
403     _preexec_exit_handlers()
404 -----
405 All the above changes are incorporated in the following patches:
406     Solaris/SunOS 5.8_sparc      patch 109147-12
407     Solaris/SunOS 5.8_x86       patch 109148-12
408 -----
410 -----
411 Solaris 9
412 -----
413 Bugid   Risk Synopsis
414 -----
415 4505289 incorrect handling of _START_ and _END_
416 4506164 mcs does not recognize #linkbefore or #linkafter qualifiers
417 4447560 strip is creating unexecutable files...
418 4513842 library names not in ld.so string pool cause corefile bugs
419 -----
420 All the above changes are incorporated in the following patches:
421     Solaris/SunOS 5.8_sparc      patch 109147-13
422     Solaris/SunOS 5.8_x86       patch 109148-13
423     Solaris/SunOS 5.7_sparc     patch 106950-16
424     Solaris/SunOS 5.7_x86       patch 106951-16
425 -----
426 4291384 ld -M with a mapfile does not properly align Fortran REAL*8 data
427 4413322 SunOS 5.9 librtld_db doesn't show dlopened ".o" files anymore?
428 4429371 librtld_db busted on ia32 with SC6.x compilers...
429 4418274 elfdump dumps core on invalid input
430 4432224 libelf xlate routines are out of date
431 4433643 Memory leak using dlopen()/dlclose() in Solaris 8
432 4446564 ldd/lddstub - core dump conditions
433 4446115 translating SUNW_move sections is broken
434 4450225 The rdb command can fall into an infinite loop
435 4448531 Linker Causes Segmentation Fault
436 4453241 Regression in 4291384 can result in empty symbol table.
437 4453398 invalid runpath token can cause ld to spin.
438 4460230 ld (for OS 5.8 and 5.9) loses error message
439 4462245 ld.so.1 core dumps when executed directly...
440 4455802 need more flexibility in establishing a support library for ld
441 4467068 dyn_plt_entsize not properly initialized in ld.so.1
442 4468779 elf_plt_trace_write() broken on i386 (link-auditing)
443 4465871 -zld32 and -zld64 does not work the way it should
444 4461890 bad shared object created with -zredlocsym
445 4469400 ld.so.1: is_so_loaded isn't as efficient as we thought...
446 4469566 lazy loading fallback can reference un-relocated objects
447 4470493 libelf incorrectly translates NOTE sections across architectures...
448 4469684 rtdld leaks dl_handles and permits on dlopen/dlclose
449 4475174 ld.so.1 prematurely reports the failure to load a object...
450 4475514 ld.so.1 can core dump in memory allocation fails (no swap)
451 4481851 Setting ld.so.1 environment variables globally would be useful
452 4482035 setting LD_PROFILE & LD_AUDIT causes ping command to issue warnings
453     on 5.8
454 4377735 segment reservations cause sbrk() to fail
455 4491434 ld.so.1 can leak file-descriptors when loading same named objects
456 4289232 some of warning/error/debugging messages from libld.so can be revised

```

```

457 4462748 Linker Portion of TLS Support
458 4496718 run-time linkers mutex_locks not working with ld_libc interface
459 4497270 The -zredlocsym option should not eliminate partially initialized local
460     symbols
461 4496963 dumping an object with crle(1) that uses $ORIGIN can loose its
462     dependencies
463 4499413 Sun linker orders of magnitude slower than gnu linker
464 4461760 lazy loading libXm and libXt can fail.
465 4469031 The partial initialized (local) symbols for intel platform is not
466     working.
467 4492883 Add link-editor option to multi-pass archives to resolve unsatisfied
468     symbols
469 4503731 linker-related commands misspell "argument"
470 4503768 whocalls(1) should output messages to stderr, not stdout
471 4503748 whocalls(1) usage message and manpage could be improved
472 4503625 nm should be taught about TLS symbols - that they aren't allowed that is
473 4300120 segment address validation is too simplistic to handle segment
474     reservations
475 4404547 krtld/reloc.h could have better error message, has typos
476 4270931 R_SPARC_HIX22 relocation is not handled properly
477 4485320 ld needs to support more the 32768 PLTs
478 4516434 sotruss can not watch libc_psr.so.1
479 4213100 sotruss could use more flexible pattern matching
480 4503457 ld seg fault with comdat
481 4510264 sections with SHF_TLS can come in different orders...
482 4518079 link-editor support library unable to modify section header flags
483 4515913 ld.so.1 can incorrectly decrement external reference counts on dlclose()
484 4519569 ld -V does not return an interesting value...
485 4524512 ld.so.1 should allow alternate termination signals
486 4524767 elfdump dies on bogus sh_name fields...
487 4524735 ld getopt processing of '-' changed
488 4521931 subroutine in a shared object as LOCL instead of GLOB
489 -----
490 All the above changes are incorporated in the following patches:
491     Solaris/SunOS 5.8_sparc      patch 109147-14
492     Solaris/SunOS 5.8_x86       patch 109148-14
493     Solaris/SunOS 5.7_sparc     patch 106950-17
494     Solaris/SunOS 5.7_x86       patch 106951-17
495 -----
496 4532729 tentative definition of TLS variable causes linker to dump core
497 4526745 fixup ld error message about duplicate dependencies/needed names
498 4522999 Solaris linker one order of magnitude slower than GNU linker
499 4518966 dldump undoes existing relocations with no thought of alignment or size.
500 4587441 Certain libraries have race conditions when setting error codes
501 4523798 linker option to align bss to large pagesize alignments.
502 4524008 ld can improperly set st_size of symbols named "_init" or "_fini"
503 4619282 ld cannot link a program with the option -sb
504 4620846 Perl Configure probing broken by ld changes
505 4621122 multiple ld '-zinitarray=' on a commandline fails
506 -----
507     Solaris/SunOS 5.8_sparc      patch 109147-15
508     Solaris/SunOS 5.8_x86       patch 109148-15
509     Solaris/SunOS 5.7_sparc     patch 106950-18
510     Solaris/SunOS 5.7_x86       patch 106951-18
511     Solaris/SunOS 5.6_sparc     patch 107733-10
512     Solaris/SunOS 5.6_x86       patch 107734-10
513 -----
514 All the above changes plus:
515     4616944 ar seg faults when order of object file is reversed.
516 are incorporated in the following patches:
517     Solaris/SunOS 5.8_sparc      patch 109147-16
518     Solaris/SunOS 5.8_x86       patch 109148-16
519 -----
520 All the above changes plus:
521     4872634 Large LD_PRELOAD values can cause SEGV of process
522 are incorporated in the following patches:

```

```

523 Solaris/SunOS 5.6_sparc patch T107733-11
524 Solaris/SunOS 5.6_x86 patch T107734-11
525 -----

527 -----
528 Solaris 9 1202 (2nd Q-update - s9u2)
529 -----

530 Bugid Risk Synopsis
531 =====
532 4546416 add help messages to ld.so mdbmodule
533 4526752 we should build and ship ld.so's mdb module
534 4624658 update 386 TLS relocation values
535 4622472 LA_SYMB_DLSYM not set for la_symbind() invocations
536 4638070 ldd/ld.so.1 could aid in detecting unreferenced dependencies
537 PSARC/2002/096 Detecting unreferenced dependencies with ldd(1)
538 4633860 Optimization for unused static global variables
539 PSARC/2002/113 ld -zignore - section elimination
540 4642829 ld.so.1 mprotect()'s text segment for weak relocations (it shouldn't)
541 4621479 'make' in $SRC/cmd/sgs/tools tries to install things in the proto area
542 4529912 purge ia64 source from sgs
543 4651709 dlopen(RTLD_NOLOAD) can disable lazy loading
544 4655066 crle: -u with nonexistent config file doesn't work
545 4654406 string tables created by the link-editor could be smaller...
546 PSARC/2002/160 ld -znoompstrtab - disable string-table compression
547 4651493 RTLD_NOW can result in binding to an object prior to its init being run.
548 4662575 linker displacement relocation checking introduces significant
549 linker overhead
550 4533195 ld interposes on malloc()/free() preventing support library from freeing
551 memory
552 4630224 crle get's confused about memory layout of objects...
553 4664855 crle on application failed with ld.so.1 encountering mmap() returning
554 ENOMEM err
555 4669582 latest dynamic linker causes libthread _init to get skipped
556 4671493 ld.so.1 inconsistently assigns PATHNAME() on primary objects
557 4668517 compile with map.bssalign doesn't copy _iob to bss
558 -----

559 All the above changes are incorporated in the following patches:
560 Solaris/SunOS 5.9_sparc patch T112963-01
561 Solaris/SunOS 5.8_sparc patch T109147-17
562 Solaris/SunOS 5.8_x86 patch T109148-17
563 -----

564 4701749 On Solaris 8 + 109147-16 ld crashes when building a dynamic library.
565 4707808 The ldd command is broken in the latest 2.8 linker patch.
566 -----

567 All the above changes are incorporated in the following patches:
568 Solaris/SunOS 5.9_sparc patch T112963-02
569 Solaris/SunOS 5.8_sparc patch T109147-18
570 Solaris/SunOS 5.8_x86 patch T109148-18
571 -----

572 4696204 enable extended section indexes in relocatable objects
573 PSARC/2001/332 ELF gABI updates - round II
574 PSARC/2002/369 libelf interfaces to support ELF Extended Sections
575 4706503 linkers need to cope with EF_SPARCV9_PSO/EF_SPARCV9_RMO
576 4716929 updating of local register symbols in dynamic sytab busted...
577 4710814 add "official" support for the "symbolic" keyword in linker map-file
578 PSARC/2002/439 linker mapfile visibility declarations
579 -----

580 All the above changes are incorporated in the following patches:
581 Solaris/SunOS 5.9_sparc patch T112963-03
582 Solaris/SunOS 5.8_sparc patch T109147-19
583 Solaris/SunOS 5.8_x86 patch T109148-19
584 Solaris/SunOS 5.7_sparc patch T106950-19
585 Solaris/SunOS 5.7_x86 patch T106951-19
586 -----

588 -----

```

```

589 Solaris 9 403 (3rd Q-update - s9u3)
590 -----

591 Bugid Risk Synopsis
592 =====
593 4731174 strip(1) does not fixup SHT_GROUP data
594 4733697 -zignore with gcc may exclude C++ exception sections
595 4733317 R_SPARC_*_HIX22 calculations are wrong with 32bit LD building
596 ELF64 binaries
597 4735165 fatal linker error when compiling C++ programs with -xlinkopt
598 4736951 The mcs broken when the target file is an archive file
599 -----

600 All the above changes are incorporated in the following patches:
601 Solaris/SunOS 5.8_sparc patch T109147-20
602 Solaris/SunOS 5.8_x86 patch T109148-20
603 Solaris/SunOS 5.7_sparc patch T106950-20
604 Solaris/SunOS 5.7_x86 patch T106951-20
605 -----

606 4739660 Threads deadlock in schedlock and dynamic linker lock.
607 4653148 ld.so.1/libc should unregister its dlclose() exit handler via a fini.
608 4743413 ld.so.1 doesn't terminate argv with NULL pointer when invoked directly
609 4746231 linker core-dumps when SECTION relocations are made against discarded
610 sections
611 4730433 ld.so.1 wastes time repeatedly opening dependencies
612 4744337 missing RD_CONSISTENT event with dllopen(LD_ID_NEWLWM, ...)
613 4670835 rd_load_objiter can ignore callback's return value
614 4745932 strip utility doesn't strip out Dwarf2 debug section
615 4754751 "strip" command doesn't remove comdat stab sections.
616 4755674 Patch 109147-18 results in coredump.
617 -----

618 All the above changes are incorporated in the following patches:
619 Solaris/SunOS 5.9_sparc patch T112963-04
620 Solaris/SunOS 5.7_sparc patch T106950-21
621 Solaris/SunOS 5.7_x86 patch T106951-21
622 -----

623 4772927 strip core dumps on an archive library
624 4774727 direct-bindings can fail against copy-reloc symbols
625 -----

626 All the above changes are incorporated in the following patches:
627 Solaris/SunOS 5.9_sparc patch T112963-05
628 Solaris/SunOS 5.9_x86 patch T113986-01
629 Solaris/SunOS 5.8_sparc patch T109147-21
630 Solaris/SunOS 5.8_x86 patch T109148-21
631 Solaris/SunOS 5.7_sparc patch T106950-22
632 Solaris/SunOS 5.7_x86 patch T106951-22
633 -----

635 -----
636 Solaris 9 803 (4th Q-update - s9u4)
637 -----

638 Bugid Risk Synopsis
639 =====
640 4730110 ld.so.1 list implementation could scale better
641 4728822 restrict the objects dlsym() searches.
642 PSARC/2002/478 New dlopen(3dl) flag - RTLD_FIRST
643 4714146 crle: 64-bit secure pathname is incorrect.
644 4504895 dlclose() does not remove all objects
645 4698800 Wrong comments in /usr/lib/ld/sparcv9/map.*
646 4745129 dldump is inconsistent with .dynamic processing errors.
647 4753066 LD_SIGNAL isn't very useful in a threaded environment
648 PSARC/2002/569 New dlinfo(3dl) flag - RTLD_DI_SIGNAL
649 4765536 crle: symbolic links can confuse alternative object configuration info
650 4766815 ld -r of object the TLS data fails
651 4770484 elfdump can not handle stripped archive file
652 4770494 The ld command gives improper error message handling broken archive
653 4775738 overwriting output relocation table when 'ld -zignore' is used
654 4778247 elfdump -e of core files fails

```

```

655 4779976 elfdump dies on bad relocation entries
656 4787579 invalid SHT_GROUP entries can cause linker to seg fault
657 4783869 dlclose: filter closure exhibits hang/failure - introduced with 4504895
658 4778418 ld.so.1: there be nits out there
659 4792461 Thread-Local Storage - x86 instruction sequence updates
660 PSARC/2002/746 Thread-Local Storage - x86 instruction sequence updates
661 4461340 sgs: ugly build output while suppressing ia64 (64-bit) build on Intel
662 4790194 dlopen(..., RTLD_GROUP) has an odd interaction with interposition
663 4804328 auditing of threaded applications results in deadlock
664 4806476 building relocatable objects with SHF_EXCLUDE loses relocation
665 information
666 -----
667 All the above changes are incorporated in the following patches:
668 Solaris/SunOS 5.9_sparc patch T112963-06
669 Solaris/SunOS 5.9_x86 patch T113986-02
670 Solaris/SunOS 5.8_sparc patch T109147-22
671 Solaris/SunOS 5.8_x86 patch T109148-22
672 -----
673 4731183 compiler creates .tlsbss section instead of .tbss as documented
674 4816378 TLS: a tls test case dumps core with C and C++ compilers
675 4817314 TLS_GD relocations against local symbols do not reference symbol...
676 4811951 non-default symbol visibility overridden by definition in shared object
677 4802194 relocation error of mozilla built by K2 compiler
678 4715815 ld should allow linking with no output file (or /dev/null)
679 4793721 Need a way to null all code in ISV objects enabling ld performance
680 tuning
681 -----
682 All the above changes plus:
683 4796237 RFE: link-editor became extremely slow with patch 109147-20 and
684 static libraries
685 are incorporated in the following patches:
686 Solaris/SunOS 5.9_sparc patch T112963-07
687 Solaris/SunOS 5.9_x86 patch T113986-03
688 Solaris/SunOS 5.8_sparc patch T109147-23
689 Solaris/SunOS 5.8_x86 patch T109148-23
690 -----
692 -----
693 Solaris 9 1203 (5th Q-update - s9u5)
694 -----
695 Bugid Risk Synopsis
696 =====
697 4830584 mmap for the padding region doesn't get freed after dlclose
698 4831650 ld.so.1 can walk off the end of it's call_init() array...
699 4831544 ldd using .so modules compiled with FD7 compiler caused a core dump
700 4834784 Accessing members in a TLS structure causes a core dump in Oracle
701 4824026 segv when -z combrelloc is used with -xlinkopt
702 4825296 typo in elfdump
703 -----
704 All the above changes are incorporated in the following patches:
705 Solaris/SunOS 5.9_sparc patch T112963-08
706 Solaris/SunOS 5.9_x86 patch T113986-04
707 Solaris/SunOS 5.8_sparc patch T109147-24
708 Solaris/SunOS 5.8_x86 patch T109148-24
709 -----
710 4470917 Solaris Process Model Unification (link-editor components only)
711 PSARC/2002/117 Solaris Process Model Unification
712 4744411 Bloomberg wants a faster linker.
713 4811969 64-bit links can be much slower than 32-bit.
714 4825065 ld(1) should ignore consecutive empty sections.
715 4838226 unrelocated shared objects may be erroneously collected for init firing
716 4830889 TLS: testcase core dumps with -xarch=v9 and -g
717 4845764 filter removal can leave dangling filtee pointer
718 4811093 appttrace -F libc date core dumps
719 4826315 Link editors need to be pre- and post- Unified Process Model aware
720 4868300 interposing on direct bindings can fail

```

```

721 4872634 Large LD_PRELOAD values can cause SEGV of process
722 -----
723 All the above changes are incorporated in the following patches:
724 Solaris/SunOS 5.9_sparc patch T112963-09
725 Solaris/SunOS 5.9_x86 patch T113986-05
726 Solaris/SunOS 5.8_sparc patch T109147-25
727 Solaris/SunOS 5.8_x86 patch T109148-25
728 -----
730 -----
731 Solaris 9 404 (6th Q-update - s9u6)
732 -----
733 Bugid Risk Synopsis
734 =====
735 4870260 The elfdump command should produce more warning message on invalid move
736 entries.
737 4865418 empty PT_TLS program headers cause problems in TLS enabled applications
738 4825151 compiler core dumped with a -mt -xF=%all test
739 4845829 The runtime linker fails to dlopen() long path name.
740 4900684 shared libraries with more than 32768 plt's fail for sparc ELF64
741 4906062 Makefiles under usr/src/cmd/sgs needs to be updated
742 -----
743 All the above changes are incorporated in the following patches:
744 Solaris/SunOS 5.9_sparc patch T112963-10
745 Solaris/SunOS 5.9_x86 patch T113986-06
746 Solaris/SunOS 5.8_sparc patch T109147-26
747 Solaris/SunOS 5.8_x86 patch T109148-26
748 Solaris/SunOS 5.7_sparc patch T106950-24
749 Solaris/SunOS 5.7_x86 patch T106951-24
750 -----
751 4900320 rtdld library mapping could be faster
752 4911775 implement GOTDATA proposal in ld
753 PSARC/2003/477 SPARC GOTDATA instruction sequences
754 4904565 Functionality to ignore relocations against external symbols
755 4764817 add section types SHT_DEBUG and SHT_DEBUGSTR
756 PSARC/2003/510 New ELF DEBUG and ANNOTATE sections
757 4850703 enable per-symbol direct bindings
758 4716275 Help required in the link analysis of runtime interfaces
759 PSARC/2003/519 Link-editors: Direct Binding Updates
760 4904573 elfdump may hang when processing archive files
761 4918310 direct binding from an executable can't be interposed on
762 4918938 ld.so.1 has become SPARC32PLUS - breaks 4.x binary compatibility
763 4911796 SIS8 C++: ld dump core when compiled and linked with xlinkopt=1.
764 4889914 ld crashes with SEGV using -M mapfile under certain conditions
765 4911936 exception are not catch from shared library with -zignore
766 -----
767 All the above changes are incorporated in the following patches:
768 Solaris/SunOS 5.9_sparc patch T112963-11
769 Solaris/SunOS 5.9_x86 patch T113986-07
770 Solaris/SunOS 5.8_sparc patch T109147-27
771 Solaris/SunOS 5.8_x86 patch T109148-27
772 Solaris/SunOS 5.7_sparc patch T106950-25
773 Solaris/SunOS 5.7_x86 patch T106951-25
774 -----
775 4946992 ld crashes due to huge number of sections (>65,000)
776 4951840 mcs -c goes into a loop on executable program
777 4939869 Need additional relocation types for abs34 code model
778 PSARC/2003/684 abs34 ELF relocations
779 -----
780 All the above changes are incorporated in the following patches:
781 Solaris/SunOS 5.9_sparc patch T112963-12
782 Solaris/SunOS 5.9_x86 patch T113986-08
783 Solaris/SunOS 5.8_sparc patch T109147-28
784 Solaris/SunOS 5.8_x86 patch T109148-28
785 -----

```

```

787 -----
788 Solaris 9 904 (7th Q-update - s9u7)
789 -----
790 Bugid Risk Synopsis
791 -----
792 4912214 Having multiple of libc.so.1 in a link map causes malloc() to fail
793 4526878 ld.so.1 should pass MAP_ALIGN flag to give kernel more flexibility
794 4930997 sgs bld_vernote.ksh script needs to be hardend...
795 4796286 ld.so.1: scenario for trouble?
796 4930985 clean up cruft under usr/src/cmd/sgs/tools
797 4933300 remove references to Ultra-1 in librtld_db demo
798 4936305 string table compression is much too slow...
799 4939626 SUNWorld internal package must be updated...
800 4939565 per-symbol filtering required
801 4948119 ld(1) -z loadfltr fails with per-symbol filtering
802 4948427 ld.so.1 gives fatal error when multiple RTLDINFO objects are loaded
803 4940894 ld core dumps using "-xldscope=symbolic
804 4955373 per-symbol filtering refinements
805 4878827 crle(1M) - display post-UPM search paths, and compensate for pre-UPM.
806 4955802 /usr/ccs/bin/ld dumps core in process_reld()
807 4964415 elfdump issues wrong relocation error message
808 4966465 LD_NOAUXFLTR fails when object is both a standard and auxiliary filter
809 4973865 the link-editor does not scale properly when linking objects with
810 lots of syms
811 4975598 SHT_SUNW_ANNOTATE section relocation not resolved
812 4974828 nss_files nss_compat r_mt tests randomly segfaulting
813 -----
814 All the above changes are incorporated in the following patches:
815 Solaris/SunOS 5.9_sparc patch T112963-13
816 Solaris/SunOS 5.9_x86 patch T113986-09
817 -----
818 4860508 link-editors should create/promote/verify hardware capabilities
819 5002160 crle: reservation for dumped objects gets confused by mmaped object
820 4967869 linking stripped library causes segv in linker
821 5006657 link-editor doesn't always handle nodirect binding syminfo information
822 4915901 no way to see ELF information
823 5021773 ld.so.1 has trouble with objects having more than 2 segments.
824 -----
825 All the above changes are incorporated in the following patches:
826 Solaris/SunOS 5.9_sparc patch T112963-14
827 Solaris/SunOS 5.9_x86 patch T113986-10
828 Solaris/SunOS 5.8_sparc patch T109147-29
829 Solaris/SunOS 5.8_x86 patch T109148-29
830 -----
831 All the above changes plus:
832 6850124 dlopen reports "No such file or directory" in spite of ENOMEM
833 when mmap fails in anon_map()
834 are incorporated in the following patches:
835 Solaris/SunOS 5.9_sparc patch TXXXXXX-XX
836 Solaris/SunOS 5.9_x86 patch TXXXXXX-XX
837 -----
839 -----
840 Solaris 10
841 -----
842 Bugid Risk Synopsis
843 -----
844 5044797 ld.so.1: secure directory testing is being skipped during filtee
845 processing
846 4963676 Remove remaining static libraries
847 5021541 unnecessary PT_SUNWBSS segment may be created
848 5031495 elfdump complains about bad symbol entries in core files
849 5012172 Need error when creating shared object with .o compiled
850 -xarch=v9 -xcode=abs44
851 4994738 rd_plt_resolution() resolves ebx-relative PLT entries incorrectly
852 5023493 ld -m output with patch 109147-25 missing .o information

```

```

853 -----
854 All the above changes are incorporated in the following patches:
855 Solaris/SunOS 5.9_sparc patch T112963-15
856 Solaris/SunOS 5.9_x86 patch T113986-11
857 Solaris/SunOS 5.8_sparc patch T109147-30
858 Solaris/SunOS 5.8_x86 patch T109148-30
859 -----
860 5071614 109147-29 & -30 break the build of on28-patch on Solaris 8 2/04
861 5029830 crle: provide for optional alternative dependencies.
862 5034652 ld.so.1 should save, and print, more error messages
863 5036561 ld.so.1 outputs non-fatal fatal message about auxiliary filter libraries
864 5042713 4866170 broke ld.so's '::setenv
865 5047082 ld can core dump on bad gcc objects
866 5047612 ld.so.1: secure pathname verification is flawed with filter use
867 5047235 elfdump can core dump printing PT_INTERP section
868 4798376 nits in demo code
869 5041446 gelf_update_*() functions inconsistently return NULL or 0
870 5032364 M_ID_TLSBSS and M_ID_UNKNOWN have the same value
871 4707030 Empty LD_PRELOAD_64 doesn't override LD_PRELOAD
872 4968618 symbolic linkage causes core dump
873 5062313 dladdr() can cause deadlock in MT apps.
874 5056867 $ISALIST/$HWCAP expansion should be more flexible.
875 4918303 0@0.so.1 should not use compiler-supplied crt*.o files
876 5058415 whocalls cannot take more than 10 arguments
877 5067518 The fix for 4918303 breaks the build if a new work space is used.
878 -----
879 All the above changes are incorporated in the following patches:
880 Solaris/SunOS 5.9_sparc patch T112963-16
881 Solaris/SunOS 5.9_x86 patch T113986-12
882 Solaris/SunOS 5.8_sparc patch T109147-31
883 Solaris/SunOS 5.8_x86 patch T109148-31
884 -----
885 5013759 *file* should report hardware/software capabilities (link-editor
886 components only)
887 5063580 libldstab: file /tmp/posto...: .stab[.index|.sbfocus] found with no
888 matching stri
889 5076838 elfdump(1) is built with a CTF section (the wrong one)
890 5080344 Hardware capabilities are not enforced for a.out
891 5079061 RTLD_DEFAULT can be expensive
892 PSARC/2004/747 New dlsym(3c) Handle - RTLD_PROBE
893 5064973 allow normal relocs against TLS symbols for some sections
894 5085792 LD_XXXX_64 should override LD_XXXX
895 5096272 every executable or library has a .SUNW_dof section
896 5094135 Bloomberg wants a faster ldd.
897 5086352 libld.so.3 should be built with a .SUNW_ctf ELF section, ready for CR
898 5098205 elfdump gives wrong section name for the global offset table
899 5092414 Linker patch 109147-29 makes Broadvison One-To-One server v4.1
900 installation fail
901 5080256 dump(1) doesn't list ELF hardware capabilities
902 5097347 recursive read lock in gelf_getsym()
903 -----
904 All the above changes are incorporated in the following patches:
905 Solaris/SunOS 5.9_sparc patch T112963-17
906 Solaris/SunOS 5.9_x86 patch T113986-13
907 Solaris/SunOS 5.8_sparc patch T109147-32
908 Solaris/SunOS 5.8_x86 patch T109148-32
909 -----
910 5106206 ld.so.1 fail to run a Solaris9 program that has libc linked with
911 -z lazyload
912 5102601 ON should deliver a 64-bit operating system for Opteron systems
913 (link-editor components only)
914 6173852 enable link_auditing technology for amd64
915 6174599 linker does not create .eh_frame_hdr sections for eh_frame sections
916 with SHF_LINK_ORDER
917 6175609 amd64 run-time linker has a corrupted note section
918 6175843 amd64 rdb_demo files not installed

```



```

919 6182293 ld.so.1 can repeatedly relocate object .plt (RTL_NOW).
920 6183645 ld core dumps when automounter fails
921 6178667 ldd list unexpected (file not found) in x86 environment.
922 6181928 Need new reloc types R_AMD64_GOTOFF64 and R_AMD64_GOTPC32
923 6182884 AMD64: ld core dumps when building a shared library
924 6173559 The ld may set incorrect value for sh_addralign under some conditions.
925 5105601 ld.so.1 gets a little too enthusiastic with interposition
926 6189384 ld.so.1 should accommodate a files dev/inode change (libc loopback mnt)
927 6177838 AMD64: linker cannot resolve PLT for 32-bit a.out(s) on amd64-S2 kernel
928 6190863 sparc disassembly code should be removed from rdb_demo
929 6191488 unwind eh_frame_hdr needs corrected encoding value
930 6192490 moe(1) returns /lib/libc.so.1 for optimal expansion of libc HWCAP
931      libraries
932 6192164 AMD64: introduce dlamd64getunwind interface
933      PSARC/2004/747 libc::dlamd64getunwind()
934 6195030 libld has bad version name
935 6195521 64-bit moe(1) missed the train
936 6198358 AMD64: bad eh_frame_hdr data when C and C++ mixed in a.out
937 6204123 ld.so.1: symbol lookup fails even after lazy loading fallback
938 6207495 UNIX98/UNIX03 vsx namespace violation DYNL_hdr/misc/dlfcn/T.dlfcn
939      14 Failed
940 6217285 ctfmerge crashed during full onmv build
941 -----
943 -----
944 Solaris 10 106 (1st Q-update - s10u1)
945 -----
946 Bugid      Risk Synopsis
947 -----
948 6209350 Do not include signature section from dynamic dependency library into
949      relocatable object
950 6212797 The binary compiled on SunOS4.x doesn't run on Solaris8 with Patch
951      109147-31
952 -----
953 All the above changes are incorporated in the following patches:
954      Solaris/SunOS 5.9_sparc      patch T112963-18
955      Solaris/SunOS 5.9_x86        patch T113986-14
956      Solaris/SunOS 5.8_sparc      patch T109147-33
957      Solaris/SunOS 5.8_x86        patch T109148-33
958 -----
959 6219538 112963-17: linker patch causes binary to dump core
960 -----
961 All the above changes are incorporated in the following patches:
962      Solaris/SunOS 5.10_sparc     patch T117461-01
963      Solaris/SunOS 5.10_x86       patch T118345-01
964      Solaris/SunOS 5.9_sparc      patch T112963-19
965      Solaris/SunOS 5.9_x86        patch T113986-15
966      Solaris/SunOS 5.8_sparc      patch T109147-34
967      Solaris/SunOS 5.8_x86        patch T109148-34
968 -----
969 6257177 incremental builds of usr/src/cmd/sgs can fail...
970 6219651 AMD64: Linker does not issue error for out of range R_AMD64_PC32
971 -----
972 All the above changes are incorporated in the following patches:
973      Solaris/SunOS 5.10_sparc     patch T117461-02
974      Solaris/SunOS 5.10_x86       patch T118345-02
975      Solaris/SunOS 5.9_sparc      patch T112963-20
976      Solaris/SunOS 5.9_x86        patch T113986-16
977      Solaris/SunOS 5.8_sparc      patch T109147-35
978      Solaris/SunOS 5.8_x86        patch T109148-35
979 NOTE: The fix for 6219651 is only applicable for 5.10_x86 platform.
980 -----
981 5080443 lazy loading failure doesn't clean up after itself (D)
982 6226206 ld.so.1 failure when processing single segment hwcap filtee
983 6228472 ld.so.1: link-map control list stacking can loose objects
984 6235000 random packages not getting installed in snv_09 and snv_10 -

```

```

985      rtdl/common/malloc.c Assertion
986 6219317 Large page support is needed for mapping executables, libraries and
987      files (link-editor components only)
988 6244897 ld.so.1 can't run apps from commandline
989 6251798 moe(1) returns an internal assertion failure message in some
990      circumstances
991 6251722 ld fails silently with exit 1 status when -z ignore passed
992 6254364 ld won't build libgenunix.so with absolute relocations
993 6215444 ld.so.1 caches "not there" lazy libraries, foils svc.startd(1M)'s logic
994 6222525 dlsym(3C) trusts caller(), which may return wrong results with tail call
995      optimization
996 6241995 warnings in sgs should be fixed (link-editor components only)
997 6258834 direct binding availability should be verified at runtime
998 6260361 lari shouldn't count a.out non-zero undefined entries as interesting
999 6260780 ldd doesn't recognize LD_NOAUXFLTR
1000 6266261 Add ld(1) -Bnoirect support (D)
1001 6261990 invalid e_flags error could be a little more friendly
1002 6261800 lari(1) should find more events uninteresting (D)
1003 6267352 libld_malloc provides inadequate alignment
1004 6268693 SHN_SUNW_IGNORE symbols should be allowed to be multiply defined
1005 6262789 Infosys wants a faster linker
1006 -----
1007 All the above changes are incorporated in the following patches:
1008      Solaris/SunOS 5.10_sparc     patch T117461-03
1009      Solaris/SunOS 5.10_x86       patch T118345-03
1010      Solaris/SunOS 5.9_sparc      patch T112963-21
1011      Solaris/SunOS 5.9_x86        patch T113986-17
1012      Solaris/SunOS 5.8_sparc      patch T109147-36
1013      Solaris/SunOS 5.8_x86        patch T109148-36
1014 -----
1015 6283601 The usr/src/cmd/sgs/packages/common/copyright contains old information
1016      legally problematic
1017 6276905 dlinfo gives inconsistent results (relative vs absolute linkname) (D)
1018      PSARC/2005/357 dlinfo(3c) RTLD_DI_ARGINFO
1019 6284941 excessive link times with many groups/sections
1020 6280467 dlclose() unmaps shared library before library's _fini() has finished
1021 6291547 ld.so mishandles LD_AUDIT causing security problems.
1022 -----
1023 All the above changes are incorporated in the following patches:
1024      Solaris/SunOS 5.10_sparc     patch T117461-04
1025      Solaris/SunOS 5.10_x86       patch T118345-04
1026      Solaris/SunOS 5.9_sparc      patch T112963-22
1027      Solaris/SunOS 5.9_x86        patch T113986-18
1028      Solaris/SunOS 5.8_sparc      patch T109147-37
1029      Solaris/SunOS 5.8_x86        patch T109148-37
1030 -----
1031 6295971 UNIX98/UNIX03 *vsx* DYNL_hdr/misc/dlfcn/T.dlfcn 14 fails, auxv.h syntax
1032      error
1033 6299525 .init order failure when processing cycles
1034 6273855 gcc and sgs/crle don't get along
1035 6273864 gcc and sgs/libld don't get along
1036 6273875 gcc and sgs/rtdl don't get along
1037 6272563 gcc and amd64/krtld/doreloc.c don't get along
1038 6290157 gcc and sgs/librtld_db/rdb_demo don't get along
1039 6301218 Matlab dumps core on startup when running on 112963-22 (D)
1040 -----
1041 All the above changes are incorporated in the following patches:
1042      Solaris/SunOS 5.10_sparc     patch T117461-06
1043      Solaris/SunOS 5.10_x86       patch T118345-08
1044      Solaris/SunOS 5.9_sparc      patch T112963-23
1045      Solaris/SunOS 5.9_x86        patch T113986-19
1046      Solaris/SunOS 5.8_sparc      patch T109147-38
1047      Solaris/SunOS 5.8_x86        patch T109148-38
1048 -----
1049 6314115 Checkpoint refuses to start, crashes on start, after application of
1050      linker patch 112963-22

```

```

1051 -----
1052 All the above changes are incorporated in the following patches:
1053 Solaris/SunOS 5.9_sparc patch T112963-24
1054 Solaris/SunOS 5.9_x86 patch T113986-20
1055 Solaris/SunOS 5.8_sparc patch T109147-39
1056 Solaris/SunOS 5.8_x86 patch T109148-39
1057 -----
1058 6318306 a dlsym() from a filter should be redirected to an associated filtee
1059 6318401 mis-aligned TLS variable
1060 6324019 ld.so.1: malloc alignment is insufficient for new compilers
1061 6324589 psh core dumps on x86 machines on snv_23
1062 6236594 AMD64: Linker needs to handle the new .lbss section (D)
1063 PSARC 2005/514 AMD64 - large section support
1064 6314743 Linker: incorrect resolution for R_AMD64_GOTPC32
1065 6311865 Linker: x86 medium model; invalid ELF program header
1066 -----
1067 All the above changes are incorporated in the following patches:
1068 Solaris/SunOS 5.10_sparc patch T117461-07
1069 Solaris/SunOS 5.10_x86 patch T118345-12
1070 -----
1071 6309061 link_audit should use __asm__ with gcc
1072 6310736 gcc and sgs/libld don't get along on SPARC
1073 6329796 Memory leak with iconv_open/iconv_close with patch 109147-33
1074 6332983 s9 linker patches 112963-24/113986-20 causing cluster machines not
1075 to boot
1076 -----
1077 All the above changes are incorporated in the following patches:
1078 Solaris/SunOS 5.10_sparc patch T117461-08
1079 Solaris/SunOS 5.10_x86 patch T121208-02
1080 Solaris/SunOS 5.9_sparc patch T112963-25
1081 Solaris/SunOS 5.9_x86 patch T113986-21
1082 Solaris/SunOS 5.8_sparc patch T109147-40
1083 Solaris/SunOS 5.8_x86 patch T109148-40
1084 -----
1085 6445311 The sparc S8/S9/S10 linker patches which include the fix for the
1086 CR6222525 are hit by the CR6439613.
1087 -----
1088 All the above changes are incorporated in the following patches:
1089 Solaris/SunOS 5.9_sparc patch T112963-26
1090 Solaris/SunOS 5.8_sparc patch T109147-41
1091 -----
1093 -----
1094 Solaris 10 807 (4th Q-update - s10u4)
1095 -----
1096 Bugid Risk Synopsis
1097 =====
1098 6487273 ld.so.1 may open arbitrary locale files when relative path is built
1099 from locale environment vars
1100 6487284 ld.so.1: buffer overflow in doprf() function
1101 -----
1102 All the above changes are incorporated in the following patches:
1103 Solaris/SunOS 5.10_sparc patch T124922-01
1104 Solaris/SunOS 5.10_x86 patch T124923-01
1105 Solaris/SunOS 5.9_sparc patch T112963-27
1106 Solaris/SunOS 5.9_x86 patch T113986-22
1107 Solaris/SunOS 5.8_sparc patch T109147-42
1108 Solaris/SunOS 5.8_x86 patch T109148-41
1109 -----
1110 6477132 ld.so.1: memory leak when running set*id application
1111 -----
1112 All the above changes are incorporated in the following patches:
1113 Solaris/SunOS 5.10_sparc patch T124922-02
1114 Solaris/SunOS 5.10_x86 patch T124923-02
1115 Solaris/SunOS 5.9_sparc patch T112963-30
1116 Solaris/SunOS 5.9_x86 patch T113986-24

```

```

1117 -----
1118 6340814 ld.so.1 core dump with HWCAP relocatable object + updated statistics
1119 6307274 crle bug with LD_LIBRARY_PATH
1120 6317969 elfheader limited to 65535 segments (link-editor components only)
1121 6350027 ld.so.1 aborts with assertion failed on amd64
1122 6362044 ld(1) inconsistencies with LD_DEBUG=-Dunused and -zignore
1123 6362047 ld.so.1 dumps core when combining HWCAP and LD_PROFILE
1124 6304206 runtime linker may respect LANG and LC_MESSAGE more than LC_ALL
1125 6363495 Catchup required with Intel relocations
1126 6326497 ld.so not properly processing LD_LIBRARY_PATH ending in :
1127 6307146 mcs dumps core when appending null string to comment section
1128 6371877 LD_PROFILE_64 with gprof does not produce correct results on amd64
1129 6372082 ld -r erroneously creates .got section on i386
1130 6201866 amd64: linker symbol elimination is broken
1131 6372620 printstack() segfaults when called from static function (D)
1132 6380470 32-bit ld(1) incorrectly builds 64-bit relocatable objects
1133 6391407 Insufficient alignment of 32-bit object in archive makes ld segfault
1134 (libelf component only) (D)
1135 6316708 LD_DEBUG should provide a means of identifying/isolating individual
1136 link-map lists (P)
1137 6280209 elfdump cores on memory model 0x3
1138 6197234 elfdump and dump don't handle 64-bit symbols correctly
1139 6398893 Extended section processing needs some work
1140 6397256 ldd dumps core in elf_fix_name
1141 6327926 ld does not set etext symbol correctly for AMD64 medium model (D)
1142 6390410 64-bit LD_PROFILE can fail: relocation error when binding profile plt
1143 6382945 AMD64-GCC: dbx: internal error: dwarf reference attribute out of bounds
1144 6262333 init section of .so dlopened from audit interface not being called
1145 6409613 elf_outsync() should fsync()
1146 6426048 C++ exceptions broken in Nevada for amd64
1147 6429418 ld.so.1: need work-around for Nvidia drivers use of static TLS
1148 6429504 crle(1) shows wrong defaults for non-existent 64-bit config file
1149 6431835 data corruption on x64 in 64-bit mode while LD_PROFILE is in effect
1150 6423051 static TLS support within the link-editors needs a major face lift (D)
1151 6388946 attempting to dlopen a .so file mislabeled as .so fails
1152 6446740 allow mapfile symbol definitions to create backing storage (D)
1153 4986360 linker crash on exec of .so (as opposed to a.out) -- error preferred
1154 instead
1155 6229145 ld: initarray/finiarray processing occurs after got size is determined
1156 6324924 the linker should warn if there's a .init section but not _init
1157 6424132 elfdump inserts extra whitespace in bitmap value display
1158 6449485 ld(1) creates misaligned TLS in binary compiled with -xpg
1159 6424550 Write to unallocated (wua) errors when libraries are built with
1160 -z lazyload
1161 6464235 executing the 64-bit ld(1) should be easy (D)
1162 6465623 need a way of building unix without an interpreter
1163 6467925 ld: section deletion (-z ignore) requires improvement
1164 6357230 specfiles should be nuked (link-editor components only)
1165 -----
1166 All the above changes are incorporated in the following patches:
1167 Solaris/SunOS 5.10_sparc patch T124922-03
1168 Solaris/SunOS 5.10_x86 patch T124923-03
1169 -----
1170 These patches also include the framework changes for the following bug fixes.
1171 However, the associated feature has not been enabled in Solaris 10 or earlier
1172 releases:
1173 -----
1174 6174390 crle configuration files are inconsistent across platforms (D, P)
1175 6432984 ld(1) output file removal - change default behavior (D)
1176 PSARC/2006/353 ld(1) output file removal - change default behavior
1177 -----
1179 -----
1180 Solaris 10 508 (5th Q-update - s10u5)
1181 -----
1182 Bugid Risk Synopsis

```

```

1183 =====
1184 6561987 data vac_conflict faults on liphread libthread libs in s10.
1185 -----
1186 All the above changes are incorporated in the following patches:
1187 Solaris/SunOS 5.10_sparc patch T127111-01
1188 Solaris/SunOS 5.10_x86 patch T127112-01
1189 -----
1190 6501793 GOTOP relocation transition (optimization) fails with offsets > 2^32
1191 6532924 AMD64: Solaris 5.11 55b: SEGV after whocatches
1192 6551627 OGL: SIGSEGV when trying to use OpenGL pipeline with splash screen,
1193 Solaris/Nvidia only
1194 -----
1195 All the above changes are incorporated in the following patches:
1196 Solaris/SunOS 5.10_sparc patch T127111-04
1197 Solaris/SunOS 5.10_x86 patch T127112-04
1198 -----
1199 6479848 Enhancements to the linker support interface needed. (D)
1200 PSARC/2006/595 link-editor support library interface - ld_open()
1201 6521608 assertion failure in runtime linker related to auditing
1202 6494228 pclose() error when an audit library calls popen() and the main target
1203 is being run under ldd (D)
1204 6568745 segfault when using LD_DEBUG with bit_audit library when instrumenting
1205 mozilla (D)
1206 PSARC/2007/413 Add -zglobalaudit option to ld
1207 6602294 ps_pbrandname breaks apps linked directly against librtld_db
1208 -----
1209 All the above changes are incorporated in the following patches:
1210 Solaris/SunOS 5.10_sparc patch T127111-07
1211 Solaris/SunOS 5.10_x86 patch T127112-07
1212 -----
1214 -----
1215 Solaris 10 908 (6th Q-update - s10u6)
1216 -----
1217 Bugid Risk Synopsis
1218 =====
1219 6672544 elf_rtbnldr must support non-ABI aligned stacks on amd64
1220 6668050 First trip through PLT does not preserve args in xmm registers
1221 -----
1222 All the above changes are incorporated in the following patch:
1223 Solaris/SunOS 5.10_x86 patch T137138-01
1224 -----
1226 -----
1227 Solaris 10 409 (7th Q-update - s10u7)
1228 -----
1229 Bugid Risk Synopsis
1230 =====
1231 6629404 ld with -z ignore doesn't scale
1232 6606203 link editor ought to allow creation of >2gb sized objects (P)
1233 -----
1234 All the above changes are incorporated in the following patches:
1235 Solaris/SunOS 5.10_sparc patch T139574-01
1236 Solaris/SunOS 5.10_x86 patch T139575-01
1237 -----
1238 6746674 setuid applications do not find libraries any more because trusted
1239 directories behavior changed (D)
1240 -----
1241 All the above changes are incorporated in the following patches:
1242 Solaris/SunOS 5.10_sparc patch T139574-02
1243 Solaris/SunOS 5.10_x86 patch T139575-02
1244 -----
1245 6703683 Can't build VirtualBox on Build 88 or 89
1246 6737579 process_req_lib() in libld consumes file descriptors
1247 6685125 ld/elfdump do not handle ZERO terminator .eh_frame amd64 unwind entry
1248 -----

```

```

1249 All the above changes are incorporated in the following patches:
1250 Solaris/SunOS 5.10_sparc patch T139574-03
1251 Solaris/SunOS 5.10_x86 patch T139575-03
1252 -----
1254 -----
1255 Solaris 10 1009 (8th Q-update - s10u8)
1256 -----
1257 Bugid Risk Synopsis
1258 =====
1259 6782597 32-bit ld.so.1 needs to accept objects with large inode number
1260 6805502 The addition of "inline" keywords to sgs code broke the lint
1261 verification in S10
1262 6807864 ld.so.1 is susceptible to a fatal dlsym()/setlocale() race
1263 -----
1264 All the above changes are incorporated in the following patches:
1265 Solaris/SunOS 5.10_sparc patch T141692-01
1266 Solaris/SunOS 5.10_x86 patch T141693-01
1267 NOTE: The fix for 6805502 is only applicable to s10.
1268 -----
1269 6826410 ld needs to sort sections using 32-bit sort keys
1270 -----
1271 All the above changes are incorporated in the following patches:
1272 Solaris/SunOS 5.10_sparc patch T141771-01
1273 Solaris/SunOS 5.10_x86 patch T141772-01
1274 NOTE: The fix for 6826410 is also available for s9 in the following patches:
1275 Solaris/SunOS 5.9_sparc patch T112963-33
1276 Solaris/SunOS 5.9_x86 patch T113986-27
1277 -----
1278 6568447 bcp is broken by 6551627
1279 6599700 librtld_db needs better plugin support
1280 6713830 mdb dumped core reading a gcore
1281 6756048 rd_loadobj_iter() should always invoke brand plugin callback
1282 6786744 32-bit dbx failed with unknown rtld_db.so error on snv_104
1283 -----
1284 All the above changes are incorporated in the following patches:
1285 Solaris/SunOS 5.10_sparc patch T141444-06
1286 Solaris/SunOS 5.10_x86 patch T141445-06
1287 -----
1289 -----
1290 Solaris 10 1005 (9th Q-update - s10u9)
1291 -----
1292 Bugid Risk Synopsis
1293 =====
1294 6850124 dlopen reports "No such file or directory" in spite of ENOMEM
1295 when mmap fails in anon_map()
1296 6826513 ldd gets confused by a crle(1) LD_PRELOAD setting
1297 6684577 ld should propagate SHF_LINK_ORDER flag to ET_REL objects
1298 6524709 executables using /usr/lib/libc.so.1 as the ELF interpreter dump core
1299 (link-editor components only)
1300 -----
1301 All the above changes are incorporated in the following patches:
1302 Solaris/SunOS 5.10_sparc patch T143895-01
1303 Solaris/SunOS 5.10_x86 patch T143896-01
1304 -----
1306 -----
1307 Solaris 10 XXXX (10th Q-update - s10u10)
1308 -----
1309 Bugid Risk Synopsis
1310 =====
1311 6478684 isainfo/cpuid reports pause instruction not supported on amd64
1312 PSARC/2010/089 Removal of AV_386_PAUSE and AV_386_MON
1313 -----
1314 All the above changes are incorporated in the following patches:

```

```

1315 Solaris/SunOS 5.10_sparc patch TXXXXXX-XX
1316 Solaris/SunOS 5.10_x86 patch TXXXXXX-XX
1317 -----
1319 -----
1320 Solaris Nevada (OpenSolaris 2008.05, snv_86)
1321 -----
1322 Bugid Risk Synopsis
1323 =====
1324 6409350 BrandZ project integration into Solaris (link-editor components only)
1325 6459189 UNIX03: *VSC* c99 compiler overwrites non-writable file
1326 6423746 add an option to relax the resolution of COMDAT relocs (D)
1327 4934427 runtime linker should load up static symbol names visible to
1328 dladdr() (D)
1329 PSARC/2006/526 SHT_SUNW_LDYNSYM - default local symbol addition
1330 6448719 sys/elf.h could be updated with additional machine and ABI types
1331 6336605 link-editors need to support R_*_SIZE relocations
1332 PSARC/2006/558 R_*_SIZE relocation support
1333 6475375 symbol search optimization to reduce rescans
1334 6475497 elfdump(1) is misreporting sh_link
1335 6482058 lari(1) could be faster, and handle per-symbol filters better
1336 6482974 defining virtual address of text segment can result in an invalid data
1337 segment
1338 6476734 crle(1m) "-l" as described fails system, crle cores trying to fix
1339 /a/var/ld/ld.config in failsafe
1340 6487499 link_audit "make clobber" creates and populates proto area
1341 6488141 ld(1) should detect attempt to reference 0-length .bss section
1342 6496718 restricted visibility symbol references should trigger archive
1343 extraction
1344 6515970 HWCAP processing doesn't clean up fmap structure - browser fails to
1345 run java applet
1346 6494214 Refinements to symbolic binding, symbol declarations and
1347 interposition (D)
1348 PSARC/2006/714 ld(1) mapfile: symbol interpose definition
1349 6475344 DTrace needs ELF function and data symbols sorted by address (D)
1350 PSARC/2007/026 ELF symbol sort sections
1351 6518480 ld -melf_i386 doesn't complain (D)
1352 6519951 bfu is just another word for exit today (RPATH -> RUNPATH conversion
1353 bites us) (D)
1354 6521504 ld: hardware capabilities processing from relocatables objects needs
1355 hardening.
1356 6518322 Some ELF utilities need updating for .SUNW_ldynsym section (D)
1357 PSARC/2007/074 -L option for nm(1) to display SHT_SUNW_LDYNSYM symbols
1358 6523787 dlopen() handle gets mistakenly orphaned - results in access to freed
1359 memory
1360 6531189 SEGV in dladdr()
1361 6527318 dlopen(name, RTLD_NOLOAD) returns handle for unloaded library
1362 6518359 extern mapfiles references to _init/_fini can create INIT/FINI
1363 addresses of 0
1364 6533587 ld.so.1: init/fini processing needs to compensate for interposer
1365 expectations
1366 6516118 Reserved space needed in ELF dynamic section and string table (D)
1367 PSARC/2007/127 Reserved space for editing ELF dynamic sections
1368 6535688 elfdump could be more robust in the face of Purify (D)
1369 6516665 The link-editors should be more resilient against gcc's symbol
1370 versioning
1371 6541004 hwcap filter processing can leak memory
1372 5108874 elfdump SEGVs on bad object file
1373 6547441 Uninitialized variable causes ld.so.1 to crash on object cleanup
1374 6341667 elfdump should check alignments of ELF header elements
1375 6387860 elfdump cores, when processing linux built ELF file
1376 6198202 mcs -d dumps core
1377 6246083 elfdump should allow section index specification
1378 (numeric -N equivalent) (D)
1379 PSARC/2007/247 Add -I option to elfdump
1380 6556563 elfdump section overlap checking is too slow for large files

```

```

1381 5006034 need ?E mapfile feature extension (D)
1382 6565476 rldld symbol version check prevents GNU ld binary from running
1383 6567670 ld(1) symbol size/section size verification uncovers Haskell
1384 compiler inconsistency
1385 6530249 elfdump should handle ELF files with no section header table (D)
1386 PSARC/2007/395 Add -P option to elfdump
1387 6573641 ld.so.1 does not maintain parent relationship to a dlopen() caller.
1388 6577462 Additional improvements needed to handling of gcc's symbol versioning
1389 6583742 ELF string conversion library needs to lose static writable buffers
1390 6589819 ld generated reference to __tls_get_addr() fails when resolving to a
1391 shared object reference
1392 6595139 various applications should export yy* global variables for libl
1393 PSARC/2007/474 new ldd(1) -w option
1394 6597841 gelf_getdyn() reads one too many dynamic entries
1395 6603313 dlclose() can fail to unload objects after fix for 6573641
1396 6234471 need a way to edit ELF objects (D)
1397 PSARC/2007/509 elfedit
1398 5035454 mixing -Kpic and -KPIC may cause SIGSEGV with -xarch=v9
1399 6473571 strip and mcs get confused and corrupt files when passed
1400 non-ELF arguments
1401 6253589 mcs has problems handling multiple SHT_NOTE sections
1402 6610591 do_reloc() should not require unused arguments
1403 6602451 new symbol visibilities required: EXPORTED, SINGLETON and ELIMINATE (D)
1404 PSARC/2007/559 new symbol visibilities - EXPORTED, SINGLETON, and
1405 ELIMINATE
1406 6570616 elfdump should display incorrectly aligned note section
1407 6614968 elfedit needs string table module (D)
1408 6620533 HWCAP filtering can leave uninitialized data behind - results in
1409 "rejected: Invalid argument"
1410 6617855 nodirect tag can be ignored when other syminfo tags are available
1411 (link-editor components only)
1412 6621066 Reduce need for new elfdump options with every section type (D)
1413 PSARC/2007/620 elfdump -T, and simplified matching
1414 6627765 soffice failure after integration of 6603313 - dangling GROUP pointer.
1415 6319025 SUNWbtool packaging issues in Nevada and S10ul.
1416 6626135 elfedit capabilities str->value mapping should come from
1417 usr/src/common/elfcap
1418 6642769 ld(1) -z combreloc should become default behavior (D)
1419 PSARC/2008/006 make ld(1) -z combreloc become default behavior
1420 6634436 XFFLAG should be updated. (link-editor components only)
1421 6492726 Merge SHF_MERGE|SHF_STRINGS input sections (D)
1422 4947191 OSNet should use direct bindings (link-editor components only)
1423 6654381 lazy loading fall-back needs optimizing
1424 6658385 ld core dumps when building Xorg on nv_82
1425 6516808 ld.so.1's token expansion provides no escape for platforms that don't
1426 report HWCAP
1427 6668534 Direct bindings can compromise function address comparisons from
1428 executables
1429 6667661 Direct bindings can compromise executables with insufficient copy
1430 relocation information
1431 6357282 ldd should recognize PARENT and EXTERN symbols (D)
1432 PSARC/2008/148 new ldd(1) -p option
1433 6672394 ldd(1) unused dependency processing is tricked by relocations errors
1434 -----
1436 -----
1437 Solaris Nevada (OpenSolaris 2008.11, snv_101)
1438 -----
1439 Bugid Risk Synopsis
1440 =====
1441 6671255 link-editor should support cross linking (D)
1442 PSARC/2008/179 cross link-editor
1443 6674666 elfedit dyn:posflag1 needs option to locate element via NEEDED item
1444 6675591 elfwrap - wrap data in an ELF file (D,P)
1445 PSARC/2008/198 elfwrap - wrap data in an ELF file
1446 6678244 elfdump dynamic section sanity checking needs refinement

```

```

1447 6679212 sgs use of SCCS id for versioning is obstacle to mercurial migration
1448 6681761 lies, darn lies, and linker README files
1449 6509323 Need to disable the Multiple Files loading - same name, different
1450 directories (or its stat() use)
1451 6686889 ld.so.1 regression - bad pointer created with 6509323 integration
1452 6695681 ldd(1) crashes when run from a chrooted environment
1453 6516212 usr/src/cmd/sgs/libelf warlock targets should be fixed or abandoned
1454 6678310 using LD_AUDIT, ld.so.1 calls shared library's .init before library is
1455 fully relocated (link-editor components only)
1456 6699594 The ld command has a problem handling 'protected' mapfile keyword.
1457 6699131 elfdump should display core file notes (D)
1458 6702260 single threading .init/.fini sections breaks staroffice
1459 6703919 boot hangs intermittently on x86 with onnv daily.0430 and on
1460 6701798 ld can enter infinite loop processing bad mapfile
1461 6706401 direct binding copy relocation fallback is insufficient for ild
1462 generated objects
1463 6705846 multithreaded C++ application seems to get deadlocked in the dynamic
1464 linker code
1465 6686343 ldd(1) - unused search path diagnosis should be enabled
1466 6712292 ld.so.1 should fall back to an interposer for failed direct bindings
1467 6716350 usr/src/cmd/sgs should be linted by nightly builds
1468 6720509 usr/src/cmd/sgs/sgsdemangler should be removed
1469 6617475 gas creates erroneous FILE symbols [was: ld.so.1 is reported as
1470 false positive by wsdiff]
1471 6724311 didump() mishandles R_AMD64_JUMP_SLOT relocations
1472 6724774 elfdump -n doesn't print siginfo structure
1473 6728555 Fix for amd64 aw (6617475) breaks pure gcc builds
1474 6734598 ld(1) archive processing failure due to mismatched file descriptors (D)
1475 6735939 ld(1) discarded symbol relocations errors (Studio and GNU).
1476 6354160 Solaris linker includes more than one copy of code in binary when
1477 linking gnu object code
1478 6744003 ld(1) could provide better argument processing diagnostics (D)
1479 PSARC 2008/583 add gld options to ld(1)
1480 6749055 ld should generate GNU style VERSYM indexes for VERNEED records (D)
1481 PSARC/2008/603 ELF objects to adopt GNU-style Versym indexes
1482 6752728 link-editor can enter UNDEF symbols in symbol sort sections
1483 6756472 AOUT search path pruning (D)
1484 -----
1486 -----
1487 Solaris Nevada (OpenSolaris 2009.06, snv_111)
1488 -----
1489 Bugid Risk Synopsis
1490 =====
1492 6754965 introduce the SF1_SUNW_ADDR32 bit in software capabilities (D)
1493 (link-editor components only)
1494 PSARC/2008/622 32-bit Address Restriction Software Capabilities Flag
1495 customer requests that DT_CONFIG strings be honored for secure apps (D)
1496 6765299 ld --version-script option not compatible with GNU ld (D)
1497 6748160 problem with -zrescan (D)
1498 PSARC/2008/651 New ld archive rescan options
1499 6763342 sloppy relocations need to get sloppier
1500 6736890 PT_SUNWBSS should be disabled (D)
1501 PSARC/2008/715 PT_SUNWBSS removal
1502 6772661 ldd/lddstub/ld.so.1 dump core in current nightly while processing
1503 libsoftcrypto_hwcaps.so.1
1504 6765931 mcs generates unlink(NULL) system calls
1505 6775062 remove /usr/lib/libldstab.so (D)
1506 6782977 ld segfaults after support lib version error sends bad args to vprintf()
1507 6773695 ld -z nopartial can break non-pic objects
1508 6778453 RTLD_GROUP prevents use of application defined malloc
1509 6789925 64-bit applications with SF1_SUNW_ADDR32 require non-default starting
1510 address
1511 6792906 ld -z nopartial fix breaks TLS
1512 6686372 ld.so.1 should use mmapobj(2)

```

```

1513 6726108 dlopen() performance could be improved.
1514 6792836 ld is slow when processing GNU linkonce sections
1515 6797468 ld.so.1: orphaned handles aren't processed correctly
1516 6798676 ld.so.1: enters infinite loop with realloc/defragmentation logic
1517 6237063 request extension to dl* family to provide segment bounds
1518 information (D)
1519 PSARC/2009/054 dlinfo(3c) - segment mapping retrieval
1520 6800388 shstrtab can be sized incorrectly when -z ignore is used
1521 6805009 ld.so.1: link map control list tear down leaves dangling pointer -
1522 pfinstall does it again.
1523 6807050 GNU linkonce sections can create duplicate and incompatible
1524 eh_frame FDE entries
1525 -----
1527 -----
1528 Solaris Nevada
1529 -----
1530 Bugid Risk Synopsis
1531 =====
1532 6813909 generalize eh_frame support to non-amd64 platforms
1533 6801536 ld: mapfile processing oddities unveiled through mmapobj(2) observations
1534 6802452 libelf shouldn't use MS_SYNC
1535 6818012 nm tries to modify readonly segment and dumps core
1536 6821646 xvm dom0 doesn't boot on daily.0324 and beyond
1537 6822828 librtld_db can return RD_ERR before RD_NOMAPS, which compromises dbx
1538 expectations.
1539 6821619 Solaris linkers need systematic approach to ELF OSABI (D)
1540 PSARC/2009/196 ELF objects to set OSABI / elfdump -O option
1541 6827468 6801536 breaks 'ld -s' if there are weak/strong symbol pairs
1542 6715578 AOUT (BCP) symbol lookup can be compromised with lazy loading.
1543 6752883 ld.so.1 error message should be buffered (not sent to stderr).
1544 6577982 ld.so.1 calls getpid() before it should when any LD_* are set
1545 6831285 linker LD_DEBUG support needs improvements (D)
1546 6806791 filter builds could be optimized (link-editor components only)
1547 6823371 calloc() uses suboptimal memset() causing 15% regression in SpecCPU2006
1548 gcc code (link-editor components only)
1549 6831308 ld.so.1: symbol rescanning does a little too much work
1550 6837777 ld ordered section code uses too much memory and works too hard
1551 6841199 Undo 10 year old workaround and use 64-bit ld on 32-bit objects
1552 6784790 ld should examine archives to determine output object class/machine (D)
1553 PSARC/2009/305 ld -32 option
1554 6849998 remove undocumented mapfile $SPECVERS and $NEED options
1555 6851224 elf_getshnum() and elf_getshstrndx() incompatible with 2002 ELF gABI
1556 agreement (D)
1557 PSARC/2009/363 replace elf_getphnum, elf_getshnum, and elf_getshstrndx
1558 6853809 ld.so.1: rescan fallback optimization is invalid
1559 6854158 ld.so.1: interposition can be skipped because of incorrect
1560 caller/destination validation
1561 6862967 rd_loadobj_iter() failing for core files
1562 6856173 streams core dumps when compiled in 64bit with a very large static
1563 array size
1564 6834197 ld pukes when given an empty plate
1565 6516644 per-symbol filtering shouldn't be allowed in executables
1566 6878605 ld should accept '%' syntax when matching input SHT_PROGBITS sections
1567 6850768 ld option to autogenerate wrappers/interposers similar to GNU ld
1568 --wrap (D)
1569 PSARC/2009/493 ld -z wrap option
1570 6888489 Null environment variables are not overriding crle(1) replaceable
1571 environment variables.
1572 6885456 Need to implement GNU-ld behavior in construction of .init/.fini
1573 sections
1574 6900241 ld should track SHT_GROUP sections by symbol name, not section name
1575 6901773 Special handling of STT_SECTION group signature symbol for GNU objects
1576 6901895 Failing asserts in ld update_osym() trying to build gcc 4.5 development
1577 head
1578 6909523 core dump when run "LD_DEBUG=help ls" in non-English locale

```

```

1579 6903688 mdb(1) can't resolve certain symbols in solaris10-branded processes
1580         from the global zone
1581 6923449 elfdump misinterprets _init/_fini symbols in dynamic section test
1582 6914728 Add dl_iterate_phdr() function to ld.so.1 (D)
1583         PSARC/2010/015 dl_iterate_phdr
1584 6916788 ld version 2 mapfile syntax (D)
1585         PSARC/2009/688 Human readable and extensible ld mapfile syntax
1586 6929607 ld generates incorrect VERDEF entries for ET_REL output objects
1587 6924224 linker should ignore SUNW_dof when calculating the elf checksum
1588 6918143 symbol capabilities (D)
1589         PSARC/2010/022 Linker-editors: Symbol Capabilities
1590 6910387 .tdata and .tbss separation invalidates TLS program header information
1591 6934123 elfdump -d core dumps on PA-RISC elf
1592 6931044 ld should not allow SHT_PROGBITS .eh_frame sections on amd64 (D)
1593 6931056 pvs -r output can include empty versions in output
1594 6938628 ld.so.1 should produce diagnostics for all dl*() entry points
1595 6938111 nm 'No symbol table data' message goes to stdout
1596 6941727 ld relocation cache memory use is excessive
1597 6932220 ld -z allextact skips objects that lack global symbols
1598 6943772 Testing for a symbols existence with RTLD_PROBE is compromised by
1599         RTLD_BIND_NOW
1600         PSARC/2010/XXX Deferred symbol references
1601 6943432 dlsym(RTLD_PROBE) should only bind to symbol definitions
1602 6668759 an external method for determining whether an ELF dependency is optional
1603 6954032 Support library with ld_open and -z allextact in snv_139 do not mix
1604 6949596 wrong section alignment generated in joint compilation with shared
1605         library
1606 6961755 ld.so.1's -e arguments should take precedence over environment
1607         variables. (D)
1608 6748925 moe returns wrong hwcap library in some circumstances
1609 6916796 OSnet mapfiles should use version 2 link-editor syntax
1610 6964517 OSnet mapfiles should use version 2 link-editor syntax (2nd pass)
1611 6948720 SHT_INIT_ARRAY etc. section names don't follow ELF gABI (D)
1612 6962343 sgsmsg should use mkstemp() for temporary file creation
1613 6965723 libsoftcrypto symbol capabilities rely on compiler generated
1614         capabilities - gcc failure (link-editor components only)
1615 6952219 ld support for archives larger than 2 GB (D, P)
1616         PSARC/2010/224 Support for archives larger than 2 GB
1617 6956152 dlclose() from an auditor can be fatal. Preinit/activity events should
1618         be more flexible. (D)
1619 6971440 moe can core dump while processing libc.
1620 6972234 sgs demo's could use some cleanup
1621 6935867 .dynamic could be readonly in sharable objects
1622 6975290 ld mishandles GOT relocation against local ABS symbol
1623 6972860 ld should provide user guidance to improve objects (D)
1624         PSARC/2010/312 Link-editor guidance
1625 -----
1627 -----
1628 Illumos
1629 -----
1630 Bugid Risk Synopsis
1631 =====
1633 308 ld may misalign sections only preceded by empty sections
1634 1301 ld crashes with '-z ignore' due to a null data descriptor
1635 1626 libld may accidentally return success while failing
1636 2413 %ymm* need to be preserved on way through PLT
1637 3210 ld should tolerate SHT_PROGBITS for .eh_frame sections on amd64
1638 3228 Want -zassert-deflib for ld
1639 3230 ld.so.1 should check default paths for DT_DEPAUDIT
1640 3260 linker is insufficiently careful with strtok
1641 3261 linker should ignore unknown hardware capabilities
1642 3265 link-editor builds bogus .eh_frame_hdr on ia32
1643 3439 discarded sections shouldn't end up on output lists
1644 3436 relocatable objects also need sloppy relocation

```

```

1645 #endif /* ! codereview */

```

```

*****
14910 Wed Jan 2 21:49:18 2013
new/usr/src/cmd/sgs/rtld/common/rtld.msg
3441 -zassert-deflib messages are in the wrong place
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1995, 2010, Oracle and/or its affiliates. All rights reserved.
24 #
25 #
26 @ _START_
27 #
28 # Message file for cmd/sgs/rtld (ld.so.1)
29 #
30 @ MSG_ID_RTLD
31 #
32 # Usage error
33 @ MSG_USG_BADOPT "usage: ld.so.1 [-e option,...] \
34                 dynamic-object [object args,...]"
35 #
36 # Message formatting error.
37 @ MSG_EMG_BUFVRFW "ld.so.1: internal: message buffer overflow"
38 #
39 # Argument processing errors
40 #
41 @ MSG_ARG_ILLMODE_1 "illegal mode: RTLD_NOW or RTLD_LAZY or RTLD_NOLOAD \
42                    required"
43 @ MSG_ARG_ILLMODE_2 "illegal mode: RTLD_NOW cannot be combined with \
44                    RTLD_LAZY"
45 @ MSG_ARG_ILLMODE_3 "illegal mode: LM_ID_NEWLM requires non-zero path"
46 @ MSG_ARG_ILLMODE_4 "illegal mode: LM_ID_NEWLM cannot be combined with \
47                    RTLD_PARENT"
48 @ MSG_ARG_ILLMODE_5 "illegal mode: potential multiple path expansion \
49                    requires RTLD_FIRST"
50 #
51 @ MSG_ARG_ILLPATH "illegal pathname"
52 @ MSG_ARG_ILLSYM "illegal symbol name"
53 @ MSG_ARG_ILLNAME "illegal name"
54 @ MSG_ARG_INVADDR "address 0x%llx does not fall within any mapped object"
55 @ MSG_ARG_INVHNDL "invalid handle: 0x%llx"
56 @ MSG_ARG_ILLVAL "illegal request value"
57 @ MSG_ARG_NOCONFIG "no configuration file in use"
58 @ MSG_ARG_NOPROFNAME "no profile target specified"
59 @ MSG_ARG_ATEXIT "purge of atexit() registrations failed: %d"
60 @ MSG_ARG_SERCNT "information path count (%d) insufficient"
61 @ MSG_ARG_SERSIZE "information buffer size (%lld) insufficient"

```

```

62 @ MSG_ARG_ILLFLAGS "illegal flags value: %d"
63 @ MSG_ARG_ILLINFO "non-null info field required for flags value: %d"
64 @ MSG_ARG_INVSIG "invalid signal supplied: %d"
65 #
66 # General error diagnostics
67 #
68 @ MSG_GEN_NOOPEN "DF_1_NOOPEN tagged object may not be dlopen()'ed"
69 #
70 @ MSG_GEN_NOFILE "%s: can't find file"
71 @ MSG_GEN_ALTER "%s: alternate file in use"
72 @ MSG_GEN_NOSYM "%s: can't find symbol"
73 @ MSG_GEN_NODUMP "%s: DF_1_NODUMP tagged object may not be dldump()'ed"
74 #
75 # Move related messages
76 #
77 @ MSG_MOVE_ERR1 "move entry with illegal size; ignored"
78 #
79 #
80 # Relocation processing messages (some of these are required to satisfy
81 # do_reloc(), which is common code used by cmd/sgs/libld - make sure both
82 # message files remain consistent).
83 #
84 @ MSG_REL_NOSYM "relocation error: file %s: symbol %s: \
85                referenced symbol not found"
86 @ MSG_REL_PLTREF "relocation error: %s: unidentifiable procedure \
87                reference: link-map 0x%llx, offset 0x%llx, \
88                called from 0x%llx"
89 @ MSG_REL_UNSUPT "relocation error: %s: file %s: symbol %s: \
90                offset size (%d bytes) is not supported"
91 @ MSG_REL_BADTLS "relocation error: %s: file %s: symbol %s: \
92                file contains insufficient TLS support information"
93 #
94 # System call messages.
95 #
96 @ MSG_SYS_BRK "%s: brk failed: %s"
97 @ MSG_SYS_OPEN "%s: open failed: %s"
98 @ MSG_SYS_MMAP "%s: mmap failed: %s"
99 @ MSG_SYS_MPROT "%s: mprotect failed: %s"
100 @ MSG_SYS_MMAPANON "%s: mmap anon failed: %s"
101 #
102 @ MSG_SEC_OPEN "%s: open failed: No such file in secure directories"
103 @ MSG_SEC_ILLEGAL "%s: open failed: illegal insecure pathname"
104 #
105 #
106 # Configuration failures
107 #
108 @ MSG_CONF_APP "configuration file: %s: is specific to application: %s"
109 @ MSG_CONF_DSTAT "configuration file: %s: original directory %s: stat \
110                 failed: %s"
111 @ MSG_CONF_FSTAT "configuration file: %s: original file %s: stat \
112                 failed: %s"
113 @ MSG_CONF_FCMP "configuration file: %s: original file %s: modified \
114                 since configuration file creation"
115 #
116 # Link Audit diagnostic message formats
117 #
118 @ MSG_AUD_BADVERS "version mismatch: current %d: required %d"
119 @ MSG_AUD_DISABLED "%s: audit initialization failure: disabled"
120 #
121 #
122 # Versioning diagnostics.
123 #
124 @ MSG_VER_NFOUND "%s: version '%s' not found (required by file %s)"
125 #
126 #
127 # Diagnostics generated under the control of ldd(1).

```

```

129 @ MSG_LDD_VER_FIND      " find version=%s\n"
130 @ MSG_LDD_VER_NFOUND    "\t%s (%s) =>\t (version not found)\n"

132 @ MSG_LDD_SYM_NFOUND    "\tsymbol not found: %s\t\t(%s)\n"

134 @ MSG_LDD_PTH_TRYING    " trying path=%s%s\n"
135 @ MSG_LDD_PTH_LIBPATH   " search path=%s (LD_LIBRARY_PATH)\n"
136 @ MSG_LDD_PTH_LIBPATHC " search path=%s (configuration \
LD_LIBRARY_PATH - %s)\n"
137 @ MSG_LDD_PTH_RUNPATH   " search path=%s (RUNPATH/RPATH from file %s)\n"
138 @ MSG_LDD_PTH_BGNDPFL  " search path="
139 @ MSG_LDD_PTH_BGNDPFL  " (default)\n"
140 @ MSG_LDD_PTH_ENDDPFL  " (configuration default - %s)\n"
141 @ MSG_LDD_PTH_ENDDPFLC " ignore path=%s (insecure directory name)\n"
142 @ MSG_LDD_PTH_IGNORE

144 @ MSG_LDD_FIL_FILTER    "\n object=%s; filter for %s\n"
145 @ MSG_LDD_FIL_FIND      "\n find object=%s; required by %s\n"
146 @ MSG_LDD_FIL_NFOUND    "\t%s =>\t (file not found)\n"
147 @ MSG_LDD_FIL_ILLEGAL  "\t%s =>\t (illegal insecure pathname)\n"
148 @ MSG_LDD_FIL_ALTER     " (alternate)"

150 @ MSG_LDD_CAP_NFOUND    "\t%s =>\t (no capability objects found)\n"

152 @ MSG_LDD_SEC_NFOUND    "\t%s =>\t (file not found in secure directories)\n"

154 @ MSG_LDD_REL_ERR1     "\trelocation %s offset invalid: %s: offset=0x%llx \
lies outside memory image; relocation discarded\n"
155 @ MSG_LDD_REL_ERR1     "\tloading after relocation has started: interposition \
request (DF_1_INTERPOSE) ignored: %s\n"
156 @ MSG_LDD_REL_ERR2     "\tmove %lld offset invalid: %s: offset=0x%llx \
lies outside memory image; move discarded\n"
157 @ MSG_LDD_MOVE_ERR     "\trelocation %s sizes differ: %s\n\
\t\t(file %s size=0x%llx; file %s size=0x%llx)\n"
158 @ MSG_LDD_COPY_SIZDIF "\t\t%s size used; possible insufficient data copied\n"
159 @ MSG_LDD_COPY_DATRUNC "\t\t%s size used; possible data truncation\n"
160 @ MSG_LDD_COPY_SIZE    "\trelocation %s symbol: %s: file %s: relocation bound \
to a symbol with STV_PROTECTED visibility\n"
161 @ MSG_LDD_COPY_SIZE    "\t\t%s size used; possible insufficient data copied\n"
162 @ MSG_LDD_COPY_SIZE    "\t\t%s size used; possible data truncation\n"
163 @ MSG_LDD_COPY_SIZE    "\trelocation %s symbol: %s: file %s: relocation bound \
to a symbol with STV_PROTECTED visibility\n"
164 @ MSG_LDD_COPY_SIZE    "\t\t%s size used; possible insufficient data copied\n"
165 @ MSG_LDD_COPY_SIZE    "\t\t%s size used; possible data truncation\n"

167 @ MSG_LDD_INIT_FMT_01  "\n cyclic dependencies detected, group [%d]:\n"
168 @ MSG_LDD_INIT_FMT_02  " init object=%s\n"
169 @ MSG_LDD_INIT_FMT_03  " init object=%s - cyclic group [%d], referenced \
by:\n"
170 @ MSG_LDD_INIT_FMT_03

172 @ MSG_LDD_UNUSED_FMT   " unused object=%s\n"
173 @ MSG_LDD_UNCYC_FMT    " unused object=%s; member of cyclic group [%d]\n"
174 @ MSG_LDD_UNREF_FMT    " unreferenced object=%s; unused dependency of %s\n"

176 @ MSG_LDD_REL_COPYDISP "\tsymbol %s: file %s: copy relocation symbol may \
have been displacement relocated\n"
177 @ MSG_LDD_REL_COPYDISP

179 @ MSG_LDD_REJ_MACH      " - wrong ELF machine type: %s"
180 @ MSG_LDD_REJ_CLASS    " - wrong ELF class: %s"
181 @ MSG_LDD_REJ_DATA     " - wrong ELF data format: %s"
182 @ MSG_LDD_REJ_TYPE     " - bad ELF type: %s"
183 @ MSG_LDD_REJ_BADFLAG  " - bad ELF flags value: %s"
184 @ MSG_LDD_REJ_MISFLAG  " - mismatched ELF flags value: %s"
185 @ MSG_LDD_REJ_VERSION  " - mismatched ELF/lib version: %s"
186 @ MSG_LDD_REJ_HAL      " - HAL R1 extensions required"
187 @ MSG_LDD_REJ_US3      " - Sun UltraSPARC III extensions required"
188 @ MSG_LDD_REJ_STR      " - %s"
189 @ MSG_LDD_REJ_UNKFILE  " - unknown file type"
190 @ MSG_LDD_REJ_UNKCAP   " - unknown capability: %d"
191 @ MSG_LDD_REJ_HWCAP_1  " - hardware capability (CA_SUNW_HW_1) unsupported: %s"
192 @ MSG_LDD_REJ_SFCAP_1  " - software capability (CA_SUNW_SF_1) unsupported: %s"
193 @ MSG_LDD_REJ_MACHCAP  " - machine capability (CA_SUNW_MACH) unsupported: %s"

```

```

194 @ MSG_LDD_REJ_PLATCAP  " - platform capability (CA_SUNW_PLAT) unsupported: %s"
195 @ MSG_LDD_REJ_HWCAP_2  " - hardware capability (CA_SUNW_HW_2) unsupported: %s"
196 @ MSG_LDD_REJ_ARCHIVE  " - invalid archive use"

198 @ MSG_LDD_WARN_UNKCAP  "%s: unknown capability: %d"
199 @ MSG_LDD_WARN_HWCAP_1 "%s: warning: hardware capability (CA_SUNW_HW_1) \
unsupported: %s\n"
200 @ MSG_LDD_WARN_SFCAP_1 "%s: warning: software capability (CA_SUNW_SF_1) \
unsupported: %s\n"
201 @ MSG_LDD_WARN_MACHCAP "%s: warning: machine capability (CA_SUNW_MACH) \
unsupported: %s\n"
202 @ MSG_LDD_WARN_PLATCAP "%s: warning: platform capability (CA_SUNW_PLAT) \
unsupported: %s\n"
203 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"
204 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"
205 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"
206 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"
207 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"
208 @ MSG_LDD_WARN_HWCAP_2 "%s: warning: hardware capability (CA_SUNW_HW_2) \
unsupported: %s\n"

210 # Error rejection messages.

212 @ MSG_ERR_REJ_MACH      "%s: wrong ELF machine type: %s"
213 @ MSG_ERR_REJ_CLASS    "%s: wrong ELF class: %s"
214 @ MSG_ERR_REJ_DATA     "%s: wrong ELF data format: %s"
215 @ MSG_ERR_REJ_TYPE     "%s: bad ELF type: %s"
216 @ MSG_ERR_REJ_BADFLAG  "%s: bad ELF flags value: %s"
217 @ MSG_ERR_REJ_MISFLAG  "%s: mismatched ELF flags value: %s"
218 @ MSG_ERR_REJ_VERSION  "%s: mismatched ELF/lib version: %s"
219 @ MSG_ERR_REJ_HAL      "%s: HAL R1 extensions required"
220 @ MSG_ERR_REJ_US3      "%s: Sun UltraSPARC III extensions required"
221 @ MSG_ERR_REJ_STR      "%s: %s"
222 @ MSG_ERR_REJ_UNKFILE  "%s: unknown file type"
223 @ MSG_ERR_REJ_UNKCAP   "%s: unknown capability: %d"
224 @ MSG_ERR_REJ_HWCAP_1 "%s: hardware capability (CA_SUNW_HW_1) unsupported: %s"
225 @ MSG_ERR_REJ_SFCAP_1 "%s: software capability (CA_SUNW_SF_1) unsupported: %s"
226 @ MSG_ERR_REJ_MACHCAP "%s: machine capability (CA_SUNW_MACH) unsupported: %s"
227 @ MSG_ERR_REJ_PLATCAP "%s: platform capability (CA_SUNW_PLAT) unsupported: %s"
228 @ MSG_ERR_REJ_HWCAP_2 "%s: hardware capability (CA_SUNW_HW_2) unsupported: %s"
229 @ MSG_ERR_REJ_ARCHIVE  "%s: invalid archive use"

231 # Error TLS failures

233 @ MSG_TLS_NOSUPPORT    "%s: TLS requirement failure : TLS support is \
unavailable"
234 @ MSG_TLS_STATBASE     "%s: static TLS failure: object is not part of primary \
link-map list"
235 @ MSG_TLS_STATSIZE    "%s: static TLS failure: object loaded after process \
initialization: size (%#llx) exceeds available backup \
reservation (%#llx)"
236 @ MSG_TLS_STATINIT    "%s: static TLS failure: object loaded after process \
initialization: can not accommodate initialized data"
237 @ MSG_TLS_STATINIT
238 @ MSG_TLS_STATINIT
239 @ MSG_TLS_STATINIT
240 @ MSG_TLS_STATINIT
241 @ MSG_TLS_STATINIT

243 # Error expand()

245 @ MSG_ERR_EXPAND1     "%s: %s: path name too long"
246 @ MSG_ERR_EXPAND2     "%s: %s: token %s could not be expanded"

248 # Specific dldinfo() messages.

250 @ MSG_DEF_NODEPFOUND   "%s: no deferred dependency found"
251 @ MSG_DEF_NOSYMFFOUND  "%s: no deferred symbol found"
252 @ MSG_DEF_DEPLOADED    "%s: deferred dependency is already loaded"

254 # Error diagnostic standard prefixes.

256 @ MSG_ERR_WARNING      "warning: "
257 @ MSG_ERR_GUIDANCE     "guidance: "
258 @ MSG_ERR_FATAL        "fatal: "
259 @ MSG_ERR_ELF          "elf error: "

```



```

261 @ MSG_STR_UNKNOWN      "(unknown)"
262 @ MSG_STR_NULL         "(null)"

264 # Unused errors - used by ldd.

266 @ MSG_USD_LDLIBPATH    " unused search path=%s (LD_LIBRARY_PATH)\n"
267 @ MSG_DUP_LDLIBPATH    " unused (duplicate) search path=%s \
268 (LD_LIBRARY_PATH)\n"
269 @ MSG_USD_LDLIBPATHC   " unused search path=%s (configuration \
270 LD_LIBRARY_PATH - %s)\n"
271 @ MSG_DUP_LDLIBPATHC   " unused (duplicate) search path=%s (configuration \
272 LD_LIBRARY_PATH - %s)\n"
273 @ MSG_USD_RUNPATH      " unused search path=%s (RUNPATH/RPATH from \
274 file %s)\n"

276 @ MSG_CAP_IGN_UNKCAP   "ignoring unknown capability: %s"

278 #endif /* ! codereview */
279 @ _END_

281 # The following strings represent reserved words, files, pathnames and symbols.
282 # Reference to this strings is via the MSG_ORIG() macro, and thus no message
283 # translation is required.

285 @ MSG_LDD_FIL_PATH      "\t%s%s%s\n"
286 @ MSG_LDD_FIL_EQUIV    "\t%s =>\t %s%s%s\n"
287 @ MSG_LDD_FMT_PATH1    "%s"
288 @ MSG_LDD_FMT_PATHN    ":%s"
289 @ MSG_LDD_INIT_FMT_FILE "\t%s\n"
290 @ MSG_LDD_VER_FOUND     "\t%s (%s) =>\t %s\n"

292 @ MSG_STR_EMPTY        ""
293 @ MSG_STR_NEGATE        "- "
294 @ MSG_STR_ZERO         "0"
295 @ MSG_STR_HEX          "0x"
296 @ MSG_STR_ELF          "ELF"
297 @ MSG_STR_MSGFOR1      "%s: %s: %s"
298 @ MSG_STR_MSGFOR2      "%s: %s"
299 @ MSG_STR_HEXNUM       "0123456789abcdef"
300 @ MSG_STR_NL           "\n"
301 @ MSG_STR_SLASH        "/"
302 @ MSG_STR_DELIMIT     " : "
303 @ MSG_STR_ONE          "1"

305 @ MSG_CAP_DELIMIT     " , "
276 @ MSG_CAP_IGN_UNKCAP   "ignoring unknown capability: %s"

307 @ MSG_SUNW_OST_SGS    "SUNW_OST_SGS"
308 @ MSG_SUNW_OST_OSLIB  "SUNW_OST_OSLIB"

310 @ MSG_TKN_CAPABILITY   "CAPABILITY"
311 @ MSG_TKN_MACHINE      "MACHINE"
312 @ MSG_TKN_PLATFORM     "PLATFORM"
313 @ MSG_TKN_ORIGIN       "ORIGIN"
314 @ MSG_TKN_ISALIST      "ISALIST"
315 @ MSG_TKN_OSNAME       "OSNAME"
316 @ MSG_TKN_OSREL        "OSREL"
317 @ MSG_TKN_HWCAP        "HWCAP"
318 @ MSG_TKN_BINDINGS     "bindings"
319 @ MSG_TKN_POSIX        "POSIX"
320 @ MSG_TKN_DOTDOT       ".."

322 @ MSG_FMT_CWD          " ."
323 @ MSG_FMT_MSGFILE      "/usr/lib/locale/%s/LC_MESSAGES/%s.mo"

```

```

325 @ MSG_FIL_RTLD        "ld.so.1"
326 @ MSG_FIL_LIBC        "libc.so.1"

328 @ MSG_SYM_ELFERRMSG   "elf_errmsg"
329 @ MSG_SYM_ELFERRNO    "elf_errno"
330 @ MSG_SYM_ELFPLTTRACE "elf_plt_trace"
331 @ MSG_SYM_ENVIRON     "_environ"

333 @ MSG_SYM_LAPREINIT   "la_preinit"
334 @ MSG_SYM_LAVERSION   "la_version"
335 @ MSG_SYM_LAACTIVITY  "la_activity"
336 @ MSG_SYM_LAOBJSEARCH "la_objsearch"
337 @ MSG_SYM_LAOBJOPEN   "la_objopen"
338 @ MSG_SYM_LAOBJFILTER "la_objfilter"
339 @ MSG_SYM_LAOBJCLOSE  "la_objclose"
340 @ MSG_SYM_LADYNDATA   "la_dyndata"

342 @ MSG_SYM_START      "_START_"

344 @ MSG_SPECFIL_DYNPLT "dyn_plt(ld.so.1)"

346 @ MSG_PTH_LDPROF     "/usr/lib/link_audit/ldprof.so.1"
347 @ MSG_PTH_LDPROFSE   "/usr/lib/secure/ldprof.so.1"
348 @ MSG_PTH_LIBSYS     "/usr/lib/libsys.so.1"
349 @ MSG_PTH_RTLD       "/usr/lib/ld.so.1"
350 @ MSG_PTH_LIB        "/lib"
351 @ MSG_PTH_USRLIB     "/usr/lib"
352 @ MSG_PTH_LIBSE      "/lib/secure"
353 @ MSG_PTH_USRLIBSE   "/usr/lib/secure"
354 @ MSG_PTH_DEVNULL    "/dev/null"
355 @ MSG_PTH_CONFIG     "/var/ld/ld.config"
356 @ MSG_PTH_VARTMP     "/var/tmp"

358 @ MSG_ORG_CONFIG     "$ORIGIN/ld.config.%s"

360 @ MSG_LD_AUDIT        "AUDIT"
361 @ MSG_LD_AUDIT_ARGS   "AUDIT_ARGS"
362 @ MSG_LD_BIND_LAZY    "BIND_LAZY"
363 @ MSG_LD_BIND_NOW     "BIND_NOW"
364 @ MSG_LD_BIND_NOT     "BIND_NOT"
365 @ MSG_LD_BINDINGS     "BINDINGS"
366 @ MSG_LD_CONFGEN     "CONFGEN"
367 @ MSG_LD_CAP_FILES    "CAP_FILES"
368 @ MSG_LD_CONFIG       "CONFIG"
369 @ MSG_LD_DEBUG        "DEBUG"
370 @ MSG_LD_DEBUG_OUTPUT "DEBUG_OUTPUT"
371 @ MSG_LD_DEMANGLE     "DEMANGLE"
372 @ MSG_LD_FLAGS        "FLAGS"
373 @ MSG_LD_HWCAP        "HWCAP"
374 @ MSG_LD_INIT         "INIT"
375 @ MSG_LD_LIBPATH      "LIBRARY_PATH"
376 @ MSG_LD_LOADAVAIL    "LOADAVAIL"
377 @ MSG_LD_LOADFLTR     "LOADFLTR"
378 @ MSG_LD_MACHCAP      "MACHCAP"
379 @ MSG_LD_NOAUDIT      "NOAUDIT"
380 @ MSG_LD_NOAUXFLTR    "NOAUXFLTR"
381 @ MSG_LD_NOBAPLT      "NOBAPLT"
382 @ MSG_LD_NOCONFIG     "NOCONFIG"
383 @ MSG_LD_NODIRCONFIG  "NODIRCONFIG"
384 @ MSG_LD_NODIRECT     "NODIRECT"
385 @ MSG_LD_NOENVCONFIG  "NOENVCONFIG"
386 @ MSG_LD_NOENVIRON    "NOENVIRON"
387 @ MSG_LD_NOFLTCONFIG  "NOFLTCONFIG"
388 @ MSG_LD_NOLAZY       "NOLAZYLOAD"
389 @ MSG_LD_NOOBJALTER   "NOOBJALTER"
390 @ MSG_LD_NOPAREXT     "NOPAREXT"

```

```
391 @ MSG_LD_NOUNRESWEAK      "NOUNRESWEAK"
392 @ MSG_LD_NOVERSION        "NOVERSION"
393 @ MSG_LD_PLATCAP          "PLATCAP"
394 @ MSG_LD_PRELOAD          "PRELOAD"
395 @ MSG_LD_PROFILE          "PROFILE"
396 @ MSG_LD_PROFILE_OUTPUT    "PROFILE_OUTPUT"
397 @ MSG_LD_SFCAP            "SFCAP"
398 @ MSG_LD_SIGNAL           "SIGNAL"
399 @ MSG_LD_TRACE_OBJS        "TRACE_LOADED_OBJECTS"
400 @ MSG_LD_TRACE_OBJS_E      "TRACE_LOADED_OBJECTS_E"
401 @ MSG_LD_TRACE_OBJS_A      "TRACE_LOADED_OBJECTS_A"
402 @ MSG_LD_TRACE_PATHS      "TRACE_SEARCH_PATHS"
403 @ MSG_LD_UNREF             "UNREF"
404 @ MSG_LD_UNUSED            "UNUSED"
405 @ MSG_LD_VERBOSE           "VERBOSE"
406 @ MSG_LD_DEFERRED          "DEFERRED"
407 @ MSG_LD_WARN              "WARN"

409 @ MSG_LD_BRAND_PREFIX     "BRAND_"

411 @ MSG_LC_ALL                "ALL="
412 @ MSG_LC_MESSAGES          "MESSAGES="

414 @ MSG_EMG_ENOMEM           "internal: Not enough space"

416 @ MSG_DBG_PID              "%5.5d: "
417 @ MSG_DBG_RESET            "-----\n"
418 @ MSG_DBG_UNDEF            "debug: "
419 @ MSG_DBG_LMID              "%s: "
420 @ MSG_DBG_THREAD           "%d: "
421 @ MSG_DBG_FILE              "%s.%5.5d"

423 @ MSG_LMID_BASE            "BASE"
424 @ MSG_LMID_LDSDO           "LDSDO"
425 @ MSG_LMID_ALT              "ALT"

427 @ MSG_LMID_FMT              "%s%d"
428 @ MSG_LMID_MAXED           "ALTMAXEDOUT"
```