```
**********************************************************
   60351 Sun May  4 06:58:06 2014
new/usr/src/cmd/sgs/libld/common/libld.msg
4843 add --as-needed and --no-as-needed to ld
**********************************************************
```

```
   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 #
  27 # Copyright (c) 2012, Joyent, Inc.  All rights reserved.
  28 #

  30 @ _START_

  32 # Message file for cmd/sgs/libld.

  34 @ MSG_ID_LIBLD

  36 #
  37 # TRANSLATION_NOTE -- Beginning of USAGE message
  38 #       The following messages are the usage messages for the ld command.
  39 #       Tab characters (\t) are used to align the messages.
  40 #
  41 #       Each usage message starts with \t, and if the message has more than one
  42 #               line, the following messages are aligned by 3 tab characters.
  43 #               When you see \n\t\t\t, the first \n is used to change the line,
  44 #               and following 3 tab characters are used to align the line.
  45 #
  46 #       Each usage message option is surrounded by [ and ].  Then the
  47 #       description of the option follows. The descriptions should be aligned,
  48 #       so tab characters are padded as needed after the closing bracket ].
  49 #
  50 #       How to align the messages are up to the translators and the
  51 #       localization engineers.
  52 #
  53 #       In C locale, the first 3 messages would look like the following:
  54 #
  55 #       usage: ld [-6:abc:.....] file(s)
  56 #               [-a]            create an absolute file
  57 #               [-b]            do not do special PIC relocations in a.out
  58 #               [-c file]       record configuration 'file'
  59 #
  60 @ MSG_ARG_USAGE         "usage: ld [-%s] file(s)\n"
  61 @ MSG_ARG_DETAIL_3      "\t[-32]\t\tenforce a 32-bit link-edit\n"
```



```
  62 @ MSG_ARG_DETAIL_6      "\t[-64]\t\tenforce a 64-bit link-edit\n"
  63 @ MSG_ARG_DETAIL_A      "\t[-a]\t\tcreate an absolute file\n"
  64 @ MSG_ARG_DETAIL_B      "\t[-b]\t\tdo not do special PIC relocations in a.out\n"
  65 @ MSG_ARG_DETAIL_CBDR   "\t[-B direct | nodirect]\n\
  66                          \t\t\testablish direct bindings, or inhibit direct \
  67                          binding\n\
  68                          \t\t\tto, the object being created\n"
  69 @ MSG_ARG_DETAIL_CBDY   "\t[-B dynamic | static]\n\
  70                          \t\t\tsearch for shared libraries|archives\n"
  71 @ MSG_ARG_DETAIL_CBE    "\t[-B eliminate]\teliminate unqualified global \
  72                          symbols from the\n\t\t\tsymbol table\n"
  73 @ MSG_ARG_DETAIL_CBG    "\t[-B group]\trelocate object from within group\n"
  74 @ MSG_ARG_DETAIL_CBL    "\t[-B local]\treduce unqualified global symbols to \
  75                          local\n"
  76 @ MSG_ARG_DETAIL_CBR    "\t[-B reduce]\tprocess symbol reductions\n"
  77 @ MSG_ARG_DETAIL_CBS    "\t[-B symbolic]\tbind external references to \
  78                          definitions when creating\n\
  79                          \t\t\tshared objects\n"
  80 @ MSG_ARG_DETAIL_C      "\t[-c name]\trecord configuration file 'name'\n"
  81 @ MSG_ARG_DETAIL_CC     "\t[-C]\t\tdemangle C++ symbol name diagnostics\n"
  82 @ MSG_ARG_DETAIL_D      "\t[-d y | n]\toperate in dynamic|static mode\n"
  83 @ MSG_ARG_DETAIL_CD     "\t[-D token,...]\tprint diagnostic messages\n"
  84 @ MSG_ARG_DETAIL_E      "\t[-e epsym], [--entry epsym]\n\
  85                          \t\t\tuse 'epsym' as entry point address\n"
  86 @ MSG_ARG_DETAIL_F      "\t[-f name], [--auxiliary name]\n\
  87                          \t\t\tspecify library for which this file is an \
  88                          auxiliary\n\t\t\tfilter\n"
  89 @ MSG_ARG_DETAIL_CF     "\t[-F name], [--filter name]\n\
  90                          \t\t\tspecify library for which this file is a filter\n
  91 @ MSG_ARG_DETAIL_CG     "\t[-G], [-shared]\n\
  92                          \t\t\tcreate a shared object\n"
  93 @ MSG_ARG_DETAIL_H      "\t[-h name], [--soname name]\n\
  94                          \t\t\tuse 'name' as internal shared object identifier\n
  95 @ MSG_ARG_DETAIL_I      "\t[-i]\t\tignore LD_LIBRARY_PATH setting\n"
  96 @ MSG_ARG_DETAIL_CI     "\t[-I name]\tuse 'name' as path of interpreter\n"
  97 @ MSG_ARG_DETAIL_L      "\t[-l x], [--library x]\n\
  98                          \t\t\tsearch for libx.so or libx.a\n"
  99 @ MSG_ARG_DETAIL_CL     "\t[-L path], [--library-path path]\n\
 100                          \t\t\tsearch for libraries in directory 'path'\n"
 101 @ MSG_ARG_DETAIL_M      "\t[-m]\t\tprint memory map\n"
 102 @ MSG_ARG_DETAIL_CM     "\t[-M mapfile]\tuse processing directives contained \
 103                          in 'mapfile'\n"
 104 @ MSG_ARG_DETAIL_CN     "\t[-N string]\tcreate a dynamic dependency for \
 105                          'string'\n"
 106 @ MSG_ARG_DETAIL_O      "\t[-o outfile], [--output outfile]\n\
 107                          \t\t\tname the output file 'outfile'\n"
 108 @ MSG_ARG_DETAIL_P      "\t[-p auditlib]\tidentify audit library to accompany \
 109                          this object\n"
 110 @ MSG_ARG_DETAIL_CP     "\t[-P auditlib]\tidentify audit library for \
 111                          processing the dependencies\n\
 112                          \t\t\tof this object\n"
 113 @ MSG_ARG_DETAIL_CQ     "\t[-Q y | n]\tdo|do not place version information in \
 114                          output file\n"
 115 @ MSG_ARG_DETAIL_R      "\t[-r], [--relocatable]\n\
 116                          \t\t\tcreate a relocatable object\n"
 117 @ MSG_ARG_DETAIL_CR     "\t[-R path], [-rpath path]\n\
 118                          \t\t\tspecify a library search path to be used at run \
 119                          time\n"
 120 @ MSG_ARG_DETAIL_S      "\t[-s], [--strip-all]\n\
 121                          \t\t\tstrip any symbol and debugging information\n"
 122 @ MSG_ARG_DETAIL_CS     "\t[-S supportlib]\n\
 123                          \t\t\tspecify a link-edit support library\n"
 124 @ MSG_ARG_DETAIL_T      "\t[-t]\t\tdo not warn of multiply-defined symbols \
 125                          that have\n\t\t\tdifferent sizes or alignments\n"
 126 @ MSG_ARG_DETAIL_U      "\t[-u symname], [--undefined symname]\n\
 127                          \t\t\tcreate an undefined symbol 'symname'\n"
```

```
128 @ MSG_ARG_DETAIL_CV     "\t[-V], [--version]\n\
129                          \t\t\tprint version information\n"
130 @ MSG_ARG_DETAIL_CY     "\t[-Y P,dirlist]\tuse 'dirlist' as a default path \
131                          when searching for\n\
132                          \t\t\tlibraries\n"
133 @ MSG_ARG_DETAIL_ZA     "\t[-z absexec]\twhen building an executable absolute \
134                          symbols\n \
135                          \t\t\treferenced in dynamic objects are promoted to\n \
136                          \t\t\tthe executable\n"
137 @ MSG_ARG_DETAIL_ZAE    "\t[-z allextract | defaultextract | weakextract],\n\
138                          \t[--whole-archive | --no-whole-archive]\n\
139                          \t\t\textract all member files, only members that \
140                          resolve\n\
141                          \t\t\tundefined or tentative symbols, or \
142                          allow extraction of\n\
143                          \t\t\tarchive members to resolve weak references from \
144                          \n\t\t\t\tarchive files\n"
145 @ MSG_ARG_DETAIL_ZAL    "\t[-z altexec64]\texecute the 64-bit link-editor\n"
146 @ MSG_ARG_DETAIL_ZADLIB "\t[-z assert-deflib]\n\
147                         \t\t\tenables warnings for linking with libraries in \
148                         the \n\t\t\tdefault search path\n\
149                         \t[-z assert-deflib=libname]\n\
150                         \t\t\tenables warnings for linking with libraries in \
151                         the \n\t\t\tdefault search path, but 'libname' is exempt
152 @ MSG_ARG_DETAIL_ZC     "\t[-z combreloc | nocombreloc]\n\
153                          \t\t\tcombine|do not combine multiple relocation \
154                          sections\n"
155 @ MSG_ARG_DETAIL_ZNC    "\t[-z nocompstrtab]\n\t\t\tdisable compression of \
156                          string tables\n"
157 @ MSG_ARG_DETAIL_ZDEF   "\t[-z deferred | nodeferred]\n\
158                          \t\t\tenable|disable deferred identification of \
159                          shared object\n\t\t\tdependencies\n"
160 @ MSG_ARG_DETAIL_ZDFS   "\t[-z defs], [--no-undefined]\n\
161                          \t\t\tdisallow undefined symbol references\n"
162 @ MSG_ARG_DETAIL_ZDRS   "\t[-z direct | nodirect]\n\
163                          \t\t\tenable|disable direct binding to shared object\n\
164                          \t\t\tdependencies\n"
165 @ MSG_ARG_DETAIL_ZE     "\t[-z endfiltee]\tmarks a filtee such that it will \
166                          terminate a filters\n\t\t\tsearch\n"
167 @ MSG_ARG_DETAIL_ZFATW  "\t[-z fatal-warnings | nofatal-warnings],\n\
168                          \t[--fatal-warnings | --no-fatal-warnings]\n\
169                          \t\tenable|disable treatment of warnings as fatal\n"
170 @ MSG_ARG_DETAIL_ZFA    "\t[-z finiarray=function]\n\
171                          \t\t\tname of function to be appended to the \
172                          .fini_array\n"
173 @ MSG_ARG_DETAIL_ZGP    "\t[-z groupperm | nogroupperm]\n\
174                          \t\t\tenable|disable setting of group permissions\n\
175                          \t\t\ton dynamic dependencies\n"
176 @ MSG_ARG_DETAIL_ZGUIDE "\t[-z guidance | -z guidance=item1,item2,...]\n\
177                          \t\t\tenable guidance warnings. items: \
178                          noall, nodefs,\n\
179                          \t\t\tnodirect, nolazyload, nomapfile, notext, \
180                          nounused\n"
181 @ MSG_ARG_DETAIL_ZH     "\t[-z help], [--help]\n\
182                          \t\t\tprint this usage message\n"
183 @ MSG_ARG_DETAIL_ZIG    "\t[-z ignore | record], [--as-needed | --no-as-needed]\
183 @ MSG_ARG_DETAIL_ZIG    "\t[-z ignore | record]\n\
184                          \t\t\tignore|record unused dynamic dependencies\n"
185 @ MSG_ARG_DETAIL_ZINA   "\t[-z initarray=function]\n\
186                          \t\t\tname of function to be appended to the \
187                          .init_array\n"
188 @ MSG_ARG_DETAIL_ZINI   "\t[-z initfirst]\tmark object to indicate that its \
189                          .init section should\n\
190                          \t\t\tbe executed before the .init section of any \
191                          other\n\t\t\tobjects\n"
192 @ MSG_ARG_DETAIL_ZINT   "\t[-z interpose]\
```



```
193                          \tdynamic object is to be an 'interposer' on direct\n\
194                          \t\t\tbindings\n"
195 @ MSG_ARG_DETAIL_ZLAZY  "\t[-z lazyload | nolazyload]\n\
196                          \t\t\tenable|disable delayed loading of shared \
197                          object\n\t\t\tdependencies\n"
198 @ MSG_ARG_DETAIL_ZLD32  "\t[-z ld32=arg1,arg2,...]\n\
199                          \t\t\tdefine arguments applicable to the \
200                          32-bit class of ld(1)\n"
201 @ MSG_ARG_DETAIL_ZLD64  "\t[-z ld64=arg1,arg2,...]\n\
202                          \t\t\tdefine arguments applicable to the \
203                          64-bit class of ld(1)\n"
204 @ MSG_ARG_DETAIL_ZLO    "\t[-z loadfltr]\tmark filter as requiring immediate \
205                          loading of its\n\
206                          \t\t\tfiltees at runtime\n"
207 @ MSG_ARG_DETAIL_ZM     "\t[-z muldefs], [--allow-multiple-definition]\n\
208                          \t\t\tallow multiply-defined symbols\n"
209 @ MSG_ARG_DETAIL_ZNDFS  "\t[-z nodefs]\tallow undefined symbol references\n"
210 @ MSG_ARG_DETAIL_ZNDEF  "\t[-z nodefaultlib]\n\
211                          \t\t\tmark object to ignore any default library \
212                          search path\n"
213 @ MSG_ARG_DETAIL_ZNDEL  "\t[-z nodelete]\tmark object as non-deletable\n"
214 @ MSG_ARG_DETAIL_ZNDLO  "\t[-z nodlopen]\tmark object as non-dlopen()'able\n"
215 @ MSG_ARG_DETAIL_ZNDU   "\t[-z nodump]\tmark object as non-dldump()'able\n"
216 @ MSG_ARG_DETAIL_ZNLD   "\t[-z noldynsym]\tdo not add a .SUNW_ldynsym section\n"
217 @ MSG_ARG_DETAIL_ZNPA   "\t[-z nopartial]\texpand any partially initialized \
218                          symbols\n"
219 @ MSG_ARG_DETAIL_ZNV    "\t[-z noversion]\tdo not record any version sections\n"
220 @ MSG_ARG_DETAIL_ZNOW   "\t[-z now]\tmark object as requiring non-lazy \
221                          binding\n"
222 @ MSG_ARG_DETAIL_ZO     "\t[-z origin]\tmark object as requiring $ORIGIN \
223                          processing\n"
224 @ MSG_ARG_DETAIL_ZPIA   "\t[-z preinitarray=function]\n\
225                          \t\t\tname of function to be appended to the \
226                          .preinit_array\n"
227 @ MSG_ARG_DETAIL_ZRL    "\t[-z redlocsym]\treduce local syms in .symtab to \
228                          a minimum\n"
229 @ MSG_ARG_DETAIL_ZRREL  "\t[-z relaxreloc]\trelax rules used for relocations \
230                          against COMDAT sections\n"
231 @ MSG_ARG_DETAIL_ZRS    "\t[-z rescan]\tafter processing all arguments, rescan \
232                          archive list\n\
233                          \t\t\tuntil no further member extraction occurs\n"
234 @ MSG_ARG_DETAIL_ZRSN   "\t[-z rescan-now]\timmediately rescan archive list \
235                          until\n\
236                          \t\t\tno further member extraction occurs\n"
237 @ MSG_ARG_DETAIL_ZRSGRP "\t[-z rescan-start archives... -z rescan-end],\n\
238                          \t[--start-group archives... --end-group], \
239                          [-( archives... -)]\n\
240                          \t\t\trescan specified archive group upon reaching\n\
241                          \t\t\tthe end of the group, until no further\n\
242                          \t\t\tmember extraction occurs\n"
243 @ MSG_ARG_DETAIL_ZSCAP  "\t[-z symbolcap]\tconvert object capabilities to \
244                          symbol capabilities\n"
245 @ MSG_ARG_DETAIL_ZTARG  "\t[-z target=platform]\n\
246                          \t\t\ttarget machine for cross linking\n"
247 @ MSG_ARG_DETAIL_ZT     "\t[-z text]\tdisallow output relocations against \
248                          text\n"
249 @ MSG_ARG_DETAIL_ZTO    "\t[-z textoff]\tallow output relocations against \
250                          text\n"
251 @ MSG_ARG_DETAIL_ZTW    "\t[-z textwarn]\twarn if there are relocations \
252                          against text\n"
253 @ MSG_ARG_DETAIL_ZWRAP  "\t[-z wrap=symbol], [-wrap=symbol], [--wrap=symbol]\n\
254                          \t\t\twrap symbol references\n"
255 @ MSG_ARG_DETAIL_ZVER   "\t[-z verbose]\t\
256                          generate warnings for suspicious processings\n"

258 #
```

```
259 # TRANSLATION_NOTE -- End of USAGE message
260 #
261 @ MSG_GRP_INVALNDX      "file %s: group section [%u]%s: entry %d: \
262                          invalid section index: %d"
263 @ MSG_GRP_INVALSYM      "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_INVALOFFSET   "relocation error: %s: file %s section [%u]%s: \
298                          invalid offset symbol '%s': offset 0x%llx"
299 @ MSG_REL_INVALRELT     "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                             0x1c       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"
392 @ MSG_REL_REMAIN_2      "%-35s 0x%-8llx\t%s"
393 @ MSG_REL_REMAIN_3      "relocations remain against allocatable but \
394                          non-writable sections"

396 # Files processing messages

398 @ MSG_FIL_MULINC_1      "file %s: attempted multiple inclusion of file"
399 @ MSG_FIL_MULINC_2      "file %s: linked to %s: attempted multiple inclusion \
400                          of file"
401 @ MSG_FIL_SOINSTAT      "input of shared object '%s' in static mode"
402 @ MSG_FIL_INVALSEC      "file %s: section [%u]%s has invalid type %s"
403 @ MSG_FIL_NOTFOUND      "file %s: required by %s, not found"
404 @ MSG_FIL_MALSTR        "file %s: section [%u]%s: malformed string table, \
405                          initial or final byte"
406 @ MSG_FIL_PTHTOLONG     "'%s/%s' pathname too long"
407 @ MSG_FIL_EXCLUDE       "file %s: section [%u]%s contains both SHF_EXCLUDE and \
408                          SHF_ALLOC flags: SHF_EXCLUDE ignored"
409 @ MSG_FIL_INTERRUPT     "file %s: creation interrupted: %s"
410 @ MSG_FIL_INVRELOC1     "file %s: section [%u]%s: relocations can not be \
411                          applied against section [%u]%s"
412 @ MSG_FIL_INVSHINFO     "file %s: section [%u]%s: has invalid sh_info: %lld"
413 @ MSG_FIL_INVSHLINK     "file %s: section [%u]%s: has invalid sh_link: %lld"
414 @ MSG_FIL_INVSHENTSIZE  "file %s: section [%u]%s: has invalid sh_entsize: %lld"
415 @ MSG_FIL_NOSTRTABLE    "file %s: section [%u]%s: symbol[%d]: specifies string \
416                          table offset 0x%llx: no string table is available"
417 @ MSG_FIL_EXCSTRTABLE   "file %s: section [%u]%s: symbol[%d]: specifies string \
418                          table offset 0x%llx: exceeds string table %s: \
419                          size 0x%llx"
420 @ MSG_FIL_NONAMESYM     "file %s: section [%u]%s: symbol[%d]: global symbol has
421                          no name"
422 @ MSG_FIL_UNKCAP        "file %s: section [%u]%s: unknown capability tag: %d"
423 @ MSG_FIL_BADSF1        "file %s: section [%u]%s: unknown software \
424                          capabilities: 0x%llx; ignored"
425 @ MSG_FIL_INADDR32SF1   "file %s: section [%u]%s: software capability ADDR32: is
426                          ineffective when building 32-bit object; ignored"
427 @ MSG_FIL_EXADDR32SF1   "file %s: section [%u]%s: software capability ADDR32: \
428                          requires executable be built with ADDR32 capability"

430 @ MSG_FIL_BADORDREF     "file %s: section [%u]%s: contains illegal reference \
431                          to discarded section: [%u]%s"

433 # Recording name conflicts

435 @ MSG_REC_OPTCNFLT      "recording name conflict: file '%s' and %s provide \
436                          identical dependency names: %s"
437 @ MSG_REC_OBJCNFLT      "recording name conflict: file '%s' and file '%s' \
438                          provide identical dependency names: %s  %s"
439 @ MSG_REC_CNFLTHINT     "(possible multiple inclusion of the same file)"

441 # System call messages

443 @ MSG_SYS_OPEN          "file %s: open failed: %s"
444 @ MSG_SYS_UNLINK        "file %s: unlink failed: %s"
445 @ MSG_SYS_MMAPANON      "mmap anon failed: %s"
446 @ MSG_SYS_MALLOC        "malloc failed: %s"


449 # Messages related to platform support

451 @ MSG_TARG_UNSUPPORTED  "unsupported ELF machine type: %s"


454 # ELF processing messages

456 @ MSG_ELF_LIBELF        "libelf: version not supported: %d"
```



```
458 @ MSG_ELF_ARMEM         "file %s: unable to locate archive member;\n\t\
459                          offset=%x, symbol=%s"

461 @ MSG_ELF_ARSYM         "file %s ignored: unable to locate archive symbol table"

463 @ MSG_ELF_VERSYM        "file %s: version symbol section entry mismatch:\n\t\
464                          (section [%u]%s entries=%d; section [%u]%s entries=%d)"

466 @ MSG_ELF_NOGROUPSECT   "file %s: section [%u]%s: SHF_GROUP flag set, but no \
467                          corresponding SHT_GROUP section found"

469 # Section processing errors

471 @ MSG_SCN_NONALLOC      "%s: non-allocatable section '%s' directed to a \
472                          loadable segment: %s"

474 @ MSG_SCN_MULTICOMDAT   "file %s: section [%u]%s: cannot be susceptible to multi
475                          COMDAT mechanisms: %s"

477 # Symbol processing errors

479 @ MSG_SYM_NOSECDEF      "symbol '%s' in file %s has no section definition"
480 @ MSG_SYM_INVSEC        "symbol '%s' in file %s associated with invalid \
481                          section[%lld]"
482 @ MSG_SYM_TLS           "symbol '%s' in file %s (STT_TLS), is defined \
483                          in a non-SHF_TLS section"
484 @ MSG_SYM_BADADDR       "symbol '%s' in file %s: section [%u]%s: size %#llx: \
485                          symbol (address %#llx, size %#llx) lies outside \
486                          of containing section"
487 @ MSG_SYM_BADADDR_ROTXT "symbol '%s' in file %s: readonly text section \
488                          [%u]%s: size %#llx: symbol (address %#llx, \
489                          size %#llx) lies outside of containing section"
490 @ MSG_SYM_MULDEF        "symbol '%s' is multiply-defined:"
491 @ MSG_SYM_CONFVIS       "symbol '%s' has conflicting visibilities:"
492 @ MSG_SYM_DIFFTYPE      "symbol '%s' has differing types:"
493 @ MSG_SYM_DIFFATTR      "symbol '%s' has differing %s:\n\
494                          \t(file %s value=0x%llx; file %s value=0x%llx);"
495 @ MSG_SYM_FILETYPES     "\t(file %s type=%s; file %s type=%s);"
496 @ MSG_SYM_VISTYPES      "\t(file %s visibility=%s; file %s visibility=%s);"
497 @ MSG_SYM_DEFTAKEN      "\t%s definition taken"
498 @ MSG_SYM_DEFUPDATE     "\t%s definition taken and updated with larger size"
499 @ MSG_SYM_LARGER        "\tlargest value applied"
500 @ MSG_SYM_TENTERR       "\ttentative symbol cannot override defined symbol \
501                          of smaller size"

503 @ MSG_SYM_INVSHNDX      "symbol %s has invalid section index; \
504                          ignored:\n\t(file %s value=%s);"
505 @ MSG_SYM_NONGLOB       "global symbol %s has non-global binding:\n\
506                          \t(file %s value=%s);"
507 @ MSG_SYM_RESERVE       "reserved symbol '%s' already defined in file %s"
508 @ MSG_SYM_NOTNULL       "undefined symbol '%s' with non-zero value encountered \
509                          from file %s"
510 @ MSG_SYM_DUPSORTADDR   "section %s: symbol '%s' and symbol '%s' have the \
511                          same address: %#llx: remove duplicate with \
512                          NOSORTSYM mapfile directive"

514 @ MSG_PSYM_INVMINFO1    "file %s: section [%u]%s: entry[%d] has invalid m_info:
515                          0x%llx for symbol index"
516 @ MSG_PSYM_INVMINFO2    "file %s: section [%u]%s: entry[%d] has invalid m_info:
517                          0x%llx for size"
518 @ MSG_PSYM_INVMREPEAT   "file %s: section [%u]%s: entry[%d] has invalid m_repeat
519                          0x%llx"
520 @ MSG_PSYM_CANNOTEXPND  "file %s: section [%u]%s: entry[%d] can not be expanded:
521                          associated symbol size is unknown %s"
522 @ MSG_PSYM_NOSTATIC     "and partial initialization cannot be deferred to \
```

```
523                          a static object"
524 @ MSG_MOVE_OVERLAP      "file %s: section [%u]%s: symbol '%s' overlapping move \
525                          initialization: start=0x%llx, length=0x%llx: \
526                          start=0x%llx, length=0x%llx"
527 @ MSG_PSYM_EXPREASON1   "output file is static object"
528 @ MSG_PSYM_EXPREASON2   "-z nopartial option in effect"
529 @ MSG_PSYM_EXPREASON3   "move infrastructure size is greater than move data"

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


540 #
541 # TRANSLATION_NOTE
542 #       The following 7 messages are the message to print the
543 #       following example messages.
544 #
545 #Undefined                      first referenced
546 # symbol                           in file
547 #inquire                          halt_hold.o
548 #
549 @ MSG_SYM_FMT_UNDEF     "%s\t\t\t%s\
550                          \n %s  \t\t\t    %s"

552 #
553 # TRANSLATION_NOTE
554 #       The next two msdid make a sentence. So translate:
555 #               "Undefined symbol"
556 #       And separate them into two msgstr considering the proper
557 #       alignment.
558 @ MSG_SYM_UNDEF_ITM_11  "Undefined"
559 @ MSG_SYM_UNDEF_ITM_12  "symbol"
560 #
561 # TRANSLATION_NOTE
562 #       The next two msdid make a sentence. So translate:
563 #               "first referenced in file"
564 #       And separate them into two msgstr considering the proper
565 #       alignment.
566 @ MSG_SYM_UNDEF_ITM_21  "first referenced"
567 @ MSG_SYM_UNDEF_ITM_22  "in file"
568 #

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

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

584 # Output file update messages

586 @ MSG_UPD_NOREADSEG     "No read-only segments found.  Setting '_etext' to 0"
587 @ MSG_UPD_NORDWRSEG     "No read-write segments found.  Setting '_edata' to 0"
588 @ MSG_UPD_NOSEG         "Setting 'end' and '_end' to 0"
```



```
590 @ MSG_UPD_SEGOVERLAP    "%s: segment address overlap;\n\
591                          \tprevious segment ending at address 0x%llx overlaps\n\
592                          \tuser defined segment '%s' starting at address 0x%llx"
593 @ MSG_UPD_LARGSIZE      "%s: segment %s calculated size 0x%llx\n\
594                          \tis larger than user-defined size 0x%llx"

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


603 # Version processing messages

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

621 # Capabilities messages

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

631 @ MSG_CAPINFO_INVALSYM  "file %s: capabilities info section [%u]%s: index %d: \
632                          family member symbol '%s': invalid"
633 @ MSG_CAPINFO_INVALLEAD "file %s: capabilities info section [%u]%s: index %d: \
634                          family lead symbol '%s': invalid symbol index %d"

636 # Basic strings

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

646 #
647 # TRANSLATION_NOTE
648 #       The elf_ function name represents a man page reference and should not
649 #       be translated.
650 @ MSG_ELF_BEGIN         "file %s: elf_begin"
651 @ MSG_ELF_CNTL          "file %s: elf_cntl"
652 @ MSG_ELF_GETARHDR      "file %s: elf_getarhdr"
653 @ MSG_ELF_GETARSYM      "file %s: elf_getarsym"
654 @ MSG_ELF_GETDATA       "file %s: elf_getdata"
```

```
655 @ MSG_ELF_GETEHDR       "file %s: elf_getehdr"
656 @ MSG_ELF_GETPHDR       "file %s: elf_getphdr"
657 @ MSG_ELF_GETSCN        "file %s: elf_getscn: scnndx: %d"
658 @ MSG_ELF_GETSHDR       "file %s: elf_getshdr"
659 @ MSG_ELF_MEMORY        "file %s: elf_memory"
660 @ MSG_ELF_NDXSCN        "file %s: elf_ndxscn"
661 @ MSG_ELF_NEWDATA       "file %s: elf_newdata"
662 @ MSG_ELF_NEWEHDR       "file %s: elf_newehdr"
663 @ MSG_ELF_NEWSCN        "file %s: elf_newscn"
664 @ MSG_ELF_NEWPHDR       "file %s: elf_newphdr"
665 @ MSG_ELF_STRPTR        "file %s: elf_strptr"
666 @ MSG_ELF_UPDATE        "file %s: elf_update"
667 @ MSG_ELF_SWAP_WRIMAGE  "file %s: _elf_swap_wrimage"


670 @ MSG_REJ_MACH          "file %s: wrong ELF machine type: %s"
671 @ MSG_REJ_CLASS         "file %s: wrong ELF class: %s"
672 @ MSG_REJ_DATA          "file %s: wrong ELF data format: %s"
673 @ MSG_REJ_TYPE          "file %s: bad ELF type: %s"
674 @ MSG_REJ_BADFLAG       "file %s: bad ELF flags value: %s"
675 @ MSG_REJ_MISFLAG       "file %s: mismatched ELF flags value: %s"
676 @ MSG_REJ_VERSION       "file %s: mismatched ELF/lib version: %s"
677 @ MSG_REJ_HAL           "file %s: HAL R1 extensions required"
678 @ MSG_REJ_US3           "file %s: Sun UltraSPARC III extensions required"
679 @ MSG_REJ_STR           "file %s: %s"
680 @ MSG_REJ_UNKFILE       "file %s: unknown file type"
681 @ MSG_REJ_UNKCAP        "file=%s; unknown capability: %d"
682 @ MSG_REJ_HWCAP_1       "file %s: hardware capability (CA_SUNW_HW_1) \
683                          unsupported: %s"
684 @ MSG_REJ_SFCAP_1       "file %s: software capability (CA_SUNW_SF_1) \
685                          unsupported: %s"
686 @ MSG_REJ_MACHCAP       "file %s: machine capability (CA_SUNW_MACH) \
687                          unsupported: %s"
688 @ MSG_REJ_PLATCAP       "file %s: platform capability (CA_SUNW_PLAT) \
689                          unsupported: %s"
690 @ MSG_REJ_HWCAP_2       "file %s: hardware capability (CA_SUNW_HW_2) \
691                          unsupported: %s"
692 @ MSG_REJ_ARCHIVE       "file %s: invalid archive use"

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

706 @ _END_


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

712 @ MSG_STR_EOF           "<eof>"
713 @ MSG_STR_ERROR         "<error>"
714 @ MSG_STR_EMPTY         ""
715 @ MSG_QSTR_BANG         "'!'"
716 @ MSG_STR_COLON         ":"
717 @ MSG_QSTR_COLON        "':'"
718 @ MSG_QSTR_SEMICOLON    "';'"
719 @ MSG_QSTR_EQUAL        "'='"
720 @ MSG_QSTR_PLUSEQ       "'+='"
```



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

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

754 @ MSG_STR_GROUP         "GROUP"
755 @ MSG_STR_SUNW_COMDAT   "SUNW_COMDAT"

757 @ MSG_FMT_ARMEM         "%s(%s)"
758 @ MSG_FMT_COLPATH       "%s:%s"
759 @ MSG_FMT_SYMNAM        "'%s'"
760 @ MSG_FMT_NULLSYMNAM    "%s[%d]"
761 @ MSG_FMT_STRCAT        "%s%s"

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

765 @ MSG_SUNW_OST_SGS      "SUNW_OST_SGS"


768 # Section strings

770 @ MSG_SCN_BSS           ".bss"
771 @ MSG_SCN_DATA          ".data"
772 @ MSG_SCN_COMMENT       ".comment"
773 @ MSG_SCN_DEBUG         ".debug"
774 @ MSG_SCN_DEBUG_INFO    ".debug_info"
775 @ MSG_SCN_DYNAMIC       ".dynamic"
776 @ MSG_SCN_DYNSYMSORT    ".SUNW_dynsymsort"
777 @ MSG_SCN_DYNTLSSORT    ".SUNW_dyntlssort"
778 @ MSG_SCN_DYNSTR        ".dynstr"
779 @ MSG_SCN_DYNSYM        ".dynsym"
780 @ MSG_SCN_DYNSYM_SHNDX  ".dynsym_shndx"
781 @ MSG_SCN_LDYNSYM       ".SUNW_ldynsym"
782 @ MSG_SCN_LDYNSYM_SHNDX ".SUNW_ldynsym_shndx"
783 @ MSG_SCN_EX_SHARED     ".ex_shared"
784 @ MSG_SCN_EX_RANGES     ".exception_ranges"
785 @ MSG_SCN_EXCL          ".excl"
786 @ MSG_SCN_FINI          ".fini"
```

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

 826 @ MSG_SYM_FINIARRAY     "finiarray"
 827 @ MSG_SYM_INITARRAY     "initarray"
 828 @ MSG_SYM_PREINITARRAY  "preinitarray"

 830 #
 831 # GNU section names
 832 #
 833 @ MSG_SCN_CTORS         ".ctors"
 834 @ MSG_SCN_DTORS         ".dtors"
 835 @ MSG_SCN_EHFRAME       ".eh_frame"
 836 @ MSG_SCN_EHFRAME_HDR   ".eh_frame_hdr"
 837 @ MSG_SCN_GCC_X_TBL     ".gcc_except_table"
 838 @ MSG_SCN_JCR           ".jcr"

 840 # Segment names for segments referenced by entrance criteria

 842 @ MSG_ENT_BSS           "bss"
 843 @ MSG_ENT_DATA          "data"
 844 @ MSG_ENT_EXTRA         "extra"
 845 @ MSG_ENT_LDATA         "ldata"
 846 @ MSG_ENT_LRODATA       "lrodata"
 847 @ MSG_ENT_NOTE          "note"
 848 @ MSG_ENT_TEXT          "text"

 850 # Symbol names

 852 @ MSG_SYM_START         "_start"
```



```
 853 @ MSG_SYM_MAIN          "main"

 855 @ MSG_SYM_FINI_U        "_fini"
 856 @ MSG_SYM_INIT_U        "_init"
 857 @ MSG_SYM_DYNAMIC       "DYNAMIC"
 858 @ MSG_SYM_DYNAMIC_U     "_DYNAMIC"
 859 @ MSG_SYM_EDATA         "edata"
 860 @ MSG_SYM_EDATA_U       "_edata"
 861 @ MSG_SYM_END           "end"
 862 @ MSG_SYM_END_U         "_end"
 863 @ MSG_SYM_ETEXT         "etext"
 864 @ MSG_SYM_ETEXT_U       "_etext"
 865 @ MSG_SYM_GOFTBL        "GLOBAL_OFFSET_TABLE_"
 866 @ MSG_SYM_GOFTBL_U      "_GLOBAL_OFFSET_TABLE_"
 867 @ MSG_SYM_PLKTBL        "PROCEDURE_LINKAGE_TABLE_"
 868 @ MSG_SYM_PLKTBL_U      "_PROCEDURE_LINKAGE_TABLE_"
 869 @ MSG_SYM_TLSGETADDR_U  "__tls_get_addr"
 870 @ MSG_SYM_TLSGETADDR_UU "___tls_get_addr"

 872 @ MSG_SYM_L_END         "END_"
 873 @ MSG_SYM_L_END_U       "_END_"
 874 @ MSG_SYM_L_START       "START_"
 875 @ MSG_SYM_L_START_U     "_START_"

 877 # Support functions

 879 @ MSG_SUP_VERSION       "ld_version"
 880 @ MSG_SUP_INPUT_DONE    "ld_input_done"

 882 @ MSG_SUP_START_64      "ld_start64"
 883 @ MSG_SUP_ATEXIT_64     "ld_atexit64"
 884 @ MSG_SUP_OPEN_64       "ld_open64"
 885 @ MSG_SUP_FILE_64       "ld_file64"
 886 @ MSG_SUP_INSEC_64      "ld_input_section64"
 887 @ MSG_SUP_SEC_64        "ld_section64"

 889 @ MSG_SUP_START         "ld_start"
 890 @ MSG_SUP_ATEXIT        "ld_atexit"
 891 @ MSG_SUP_OPEN          "ld_open"
 892 @ MSG_SUP_FILE          "ld_file"
 893 @ MSG_SUP_INSEC         "ld_input_section"
 894 @ MSG_SUP_SEC           "ld_section"

 896 #
 897 # Message previously in 'ld'
 898 #
 899 #
 900 @ _START_

 902 # System error messages

 904 @ MSG_SYS_STAT          "file %s: stat failed: %s"
 905 @ MSG_SYS_READ          "file %s: read failed: %s"
 906 @ MSG_SYS_NOTREG        "file %s: is not a regular file"

 908 # Argument processing messages

 910 @ MSG_ARG_DY_INCOMP     "%s option is incompatible with building a dynamic \
 911                          executable"
 912 @ MSG_MARG_DY_INCOMP    "%s is incompatible with building a dynamic \
 913                          executable"
 914 @ MSG_ARG_ST_INCOMP     "%s option is incompatible with building a static \
 915                          object (-dn, -r, --relocatable)"
 916 @ MSG_MARG_ST_INCOMP    "%s is incompatible with building a static \
 917                          object (-dn, -r, --relocatable)"
 918 @ MSG_MARG_ST_ONLYAVL   "%s is only available when building a shared object"
```

```
 919 @ MSG_ARG_INCOMP        "option %s and %s are incompatible"
 920 @ MSG_MARG_INCOMP       "%s and %s are incompatible"
 921 @ MSG_ARG_MTONCE        "option %s appears more than once, first setting taken"
 922 @ MSG_MARG_MTONCE       "%s appears more than once, first setting taken"
 923 @ MSG_ARG_ILLEGAL       "option %s has illegal argument '%s'"
 924 @ MSG_ARG_YP            "option -YP and -Y%c may not be specified concurrently"
 925 @ MSG_ARG_STRIP         "%s specified with %s; only debugging \
 926                          information stripped"
 927 @ MSG_ARG_NOFILES       "no files on input command line"
 928 @ MSG_ARG_NOFLTR        "option %s is only meaningful when building a filter"
 929 @ MSG_ARG_NODEFLIB      "the default library search path has been suppressed, \
 930                          but no runpaths have been specified via %s"
 931 @ MSG_ARG_NOENTRY       "entry point symbol '%s' is undefined"
 932 @ MSG_ARG_UNSUPPORTED   "option %s is no longer supported; ignored"
 933 @ MSG_MARG_ONLY         "option %s can only be used with a %s"
 934 @ MSG_ARG_UNKNOWN       "unrecognized option '-%c'"
 935 @ MSG_ARG_LONG_UNKNOWN  "unrecognized option '%s'"
 936 @ MSG_ARG_USEHELP       "use the -z help option for usage information"


 939 @ MSG_ARG_FLAGS         "flags processing errors"
 940 @ MSG_ARG_FILES         "file processing errors. No output written to %s"
 941 @ MSG_ARG_SYM_WARN      "symbol referencing errors"
 942 @ MSG_ARG_SYM_FATAL     "symbol referencing errors. No output written to %s"
 943 @ MSG_ARG_AR_GRP_OLAP   "%s cannot be nested"
 944 @ MSG_ARG_AR_GRP_BAD    "%s used without corresponding %s"


 947 # Messages used to refer to options where there is more than
 948 # one name accepted.

 950 @ MSG_MARG_AR_GRPS      "archive rescan groups \
 951                          (-z rescan-start, -(, --start-group)"
 952 @ MSG_MARG_AR_GRP_END   "archive rescan group end option \
 953                          (-z rescan-end, -), --end-group)"
 954 @ MSG_MARG_AR_GRP_START "archive rescan group start option \
 955                          (-z rescan-start, -(, --start-group)"
 956 @ MSG_MARG_ENTRY        "entry point option (-e, --entry)"
 957 @ MSG_MARG_FILTER_AUX   "auxiliary filter option (-f, --auxiliary)"
 958 @ MSG_MARG_FILTER       "filter option (-F, --filter)"
 959 @ MSG_MARG_OUTFILE      "output object option (-o, --output)"
 960 @ MSG_MARG_REL          "relocatable object option (-r, --relocatable)"
 961 @ MSG_MARG_RPATH        "runpath option (-R, -rpath)"
 962 @ MSG_MARG_SO           "shared object option (-G, -shared)"
 963 @ MSG_MARG_SONAME       "soname option (-h, --soname)"
 964 @ MSG_MARG_STRIP        "strip option (-s, --strip-all)"

 966 # Entrance criteria messages

 968 @ MSG_ENT_MAP_FMT_TIL_1 "\t\t%s\n\n"
 969 @ MSG_ENT_MAP_TITLE_1   "LINK EDITOR MEMORY MAP"

 971 #
 972 # TRANSLATION_NOTE -- Entry map header
 973 #
 974 # The next message is a format string for a title. The title is composed of
 975 # two lines. In C locale, it would look like:
 976 #
 977 #       output          input           new
 978 #       section         section         displacement    size
 979 #
 980 # The \t characters are used for alignment.  (output section), (input section),
 981 # and (new displacement) have to be aligned.
 982 #
 983 @ MSG_ENT_MAP_FMT_TIL_2 "\n%s\t\t%s\t\t%s\n%s\t\t%s\t\t%s\t%s\n\n"
 984 @ MSG_ENT_MAP_FMT_TIL_3 "\n%s\t\t%s\t\t%s\n%s\t\t%s\t\t%s\t\t%s\n\n"
```



```
 985 @ MSG_ENT_ITM_OUTPUT    "output"
 986 @ MSG_ENT_ITM_INPUT     "input"
 987 @ MSG_ENT_ITM_NEW       "new"
 988 @ MSG_ENT_ITM_SECTION   "section"
 989 @ MSG_ENT_ITM_DISPMNT   "displacement"
 990 @ MSG_ENT_ITM_SIZE      "size"
 991 @ MSG_ENT_ITM_VIRTUAL   "virtual"
 992 @ MSG_ENT_ITM_ADDRESS   "address"

 994 @ MSG_ENT_MAP_ENTRY_1   "%-8.8s\t\t\t%08.2llx\t%08.2llx\n"
 995 @ MSG_ENT_MAP_ENTRY_2   "\t\t%-8.8s\t%08.2llx\t%08.2llx %s\n"

 997 #
 998 # TRANSLATION_NOTE -- multiple defined symbol table header
 999 #
1000 # In C locale, an example output is:
1001 #
1002 #               MULTIPLY DEFINED SYMBOLS
1003 #
1004 #
1005 #symbol                          definition used     also defined in
1006 #
1007 #variable1                          main.o
1008 #                                                      ./libfred.so
1009 @ MSG_ENT_MUL_FMT_TIL_0 "\n\n\t\t%s\n\n\n"
1010 @ MSG_ENT_MUL_TIL_0     "MULTIPLY DEFINED SYMBOLS"

1012 #
1013 # TRANSLATION_NOTE -- This is the format string for:
1014 #
1015 #symbol                          definition used     also defined in
1016 #
1017 @ MSG_ENT_MUL_FMT_TIL_1 "%s\t\t\t\t  %s     %s\n\n"
1018 @ MSG_ENT_MUL_ITM_SYM   "symbol"
1019 @ MSG_ENT_MUL_ITM_DEF_0 "definition used"
1020 @ MSG_ENT_MUL_ITM_DEF_1 "also defined in"

1022 #
1023 # TRANSLATION_NOTE -- This is the format string for the second item:
1024 #
1025 @ MSG_ENT_MUL_ENTRY_1   "%-35s %s\n"

1027 #
1028 # TRANSLATION_NOTE -- This is the format string for the third item:
1029 #
1030 @ MSG_ENT_MUL_ENTRY_2   "\t\t\t\t\t\t\t%s\n"

1032 @ MSG_ENT_NOSEC_1       "mapfile: %s segment: section '%s' does not appear \
1033                          in mapfile specified input file(s)"
1034 @ MSG_ENT_NOSEC_2       "mapfile: %s segment: section '%s' does not appear \
1035                          in any input file"

1037 # Library messages

1039 @ MSG_LIB_NOTFOUND      "library -l%s: not found"
1040 @ MSG_LIB_MALFORM       "LD_LIBRARY_PATH malformed"
1041 @ MSG_LIB_BADYP         "-YP library path malformed"


1044 # Mapfile processing messages

1046 @ MSG_MAP_BADAUTORED    "%s: %llu: auto-reduction ('*') can only be used in \
1047                          hidden/local, or eliminate scope"
1048 @ MSG_MAP_BADFLAG       "%s: %llu: badly formed section flags '%s'"
1049 @ MSG_MAP_BADBNAME      "%s: %llu: basename cannot contain path \
1050                          separator ('/'): %s"
```

```
1051 @ MSG_MAP_BADONAME      "%s: %llu: object name cannot contain path \
1052                          separator (’/’): %s"
1053 @ MSG_MAP_REDEFATT      "%s: %llu: redefining %s attribute for ’%s’"
1054 @ MSG_MAP_PREMEOF       "%s: %llu: premature EOF"
1055 @ MSG_MAP_ILLCHAR       "%s: %llu: illegal character ’\\%03o’"
1056 @ MSG_MAP_MALFORM       "%s: %llu: malformed entry"
1057 @ MSG_MAP_NONLOAD       "%s: %llu: %s not allowed on non-LOAD segments"
1058 @ MSG_MAP_NOSTACK1      "%s: %llu: %s not allowed on STACK segment"
1059 @ MSG_MAP_MOREONCE      "%s: %llu: %s set more than once on same line"
1060 @ MSG_MAP_NOTERM        "%s: %llu: unterminated quoted string: %s"
1061 @ MSG_MAP_SECINSEG      "%s: %llu: section within segment ordering done on \
1062                          a non-existent segment ’%s’"
1063 @ MSG_MAP_UNEXINHERIT   "%s: %llu: unnamed version cannot inherit from other \
1064                          versions: %s"
1065 @ MSG_MAP_UNEXTOK       "%s: %llu: unexpected occurrence of ’%c’ token"

1067 @ MSG_MAP_SEGEMPLOAD    "%s: %llu: empty segment must be of type LOAD or NULL"
1068 @ MSG_MAP_SEGEMPEXE     "%s: %llu: a LOAD empty segment definition is only \
1069                          allowed when creating a dynamic executable"
1070 @ MSG_MAP_SEGEMPATT     "%s: %llu: a LOAD empty segment must have an address \
1071                          and size"
1072 @ MSG_MAP_SEGEMPNOATT   "%s: %llu: a NULL empty segment must not have an \
1073                          address or size"
1074 @ MSG_MAP_SEGEMPSEC     "%s: %llu: empty segment can not have sections \
1075                          assigned to it"
1076 @ MSG_MAP_SEGEMNOPERM   "%s: %llu: empty segment must not have \
1077                          p_flags set: 0x%x"

1079 @ MSG_MAP_CNTADDRORDER  "%s: %llu: segment cannot have an explicit address \
1080                          and also be in the SEGMENT_ORDER list: %s"
1081 @ MSG_MAP_CNTDISSEG     "%s: %llu: segment cannot be disabled: %s"
1082 @ MSG_MAP_DUPNAMENT     "%s: %llu: cannot redefine entrance criteria: %s"
1083 @ MSG_MAP_DUPORDSEG     "%s: %llu: segment is already in %s list: %s"
1084 @ MSG_MAP_DUP_OS_ORD    "%s: %llu: section is already in OS_ORDER list: %s"
1085 @ MSG_MAP_DUP_IS_ORD    "%s: %llu: entrance criteria is already in \
1086                          IS_ORDER list: %s"
1087 @ MSG_MAP_UNKENT        "%s: %llu: unknown entrance criteria \
1088                          (ASSIGN_SECTION): %s"
1089 @ MSG_MAP_UNKSEG        "%s: %llu: unknown segment: %s"
1090 @ MSG_MAP_UNKSYMDEF     "%s: %llu: unknown symbol definition: %s"
1091 @ MSG_MAP_UNKSEGTYP     "%s: %llu: unknown internal segment type %d"
1092 @ MSG_MAP_UNKSOTYP      "%s: %llu: unknown shared object type: %s"
1093 @ MSG_MAP_UNKSEGATT     "%s: %llu: unknown segment attribute: %s"
1094 @ MSG_MAP_UNKSEGFLG     "%s: %llu: unknown segment flag: ?%c"
1095 @ MSG_MAP_UNKSECTYP     "%s: %llu: unknown section type: %s"

1097 @ MSG_MAP_SEGSIZE       "%s: %lld: existing segment size symbols cannot \
1098                          be reset: %s"
1099 @ MSG_MAP_SEGADDR       "%s: %llu: segment address or length ’%s’ %s"
1100 @ MSG_MAP_BADCAPVAL     "%s: %llu: bad capability value: %s"
1101 @ MSG_MAP_UNKCAPATTR    "%s: %llu: unknown capability attribute ’%s’"
1102 @ MSG_MAP_EMPTYCAP      "%s: %llu: empty capability definition; ignored"

1104 @ MSG_MAP_SYMDEF1       "%s: %llu: symbol ’%s’ is already defined in file: \
1105                          %s: %s"
1106 @ MSG_MAP_SYMDEF2       "%s: %llu: symbol ’%s’: %s"

1108 @ MSG_MAP_EXPSCOL       "%s: %llu: expected a ’;’"
1109 @ MSG_MAP_EXPEQU        "%s: %llu: expected a ’=’, ’:’, ’|’, or ’@’"
1110 @ MSG_MAP_EXPSEGATT     "%s: %llu: expected one or more segment attributes \
1111                          after an ’=’"
1112 @ MSG_MAP_EXPSEGNAM     "%s: %llu: expected a segment name at the beginning \
1113                          of a line"
1114 @ MSG_MAP_EXPSEGTYPE    "%s: %llu: %s segment cannot be used with %s \
1115                          directive: %s"
1116 @ MSG_MAP_EXPSYM_1      "%s: %llu: expected a symbol name after ’@’"
```



```
1117 @ MSG_MAP_EXPSYM_2      "%s: %llu: expected a symbol name after ’{’"
1118 @ MSG_MAP_EXPSEC        "%s: %llu: expected a section name after ’|’"
1119 @ MSG_MAP_EXPSO         "%s: %llu: expected a shared object definition \
1120                          after ’-’"
1121 @ MSG_MAP_MULTFILTEE    "%s: %llu: multiple filtee definitions are unsupported"
1122 @ MSG_MAP_NOFILTER      "%s: %llu: filtee definition required"
1123 @ MSG_MAP_BADSF1        "%s: %llu: unknown software capabilities: 0x%llx; \
1124                          ignored"
1125 @ MSG_MAP_INADDR32SF1   "%s: %llu: software capability ADDR32: is ineffective \
1126                          when building 32-bit object: ignored"
1127 @ MSG_MAP_NOINTPOSE     "%s: %llu: interposition symbols can only be defined \
1128                          when building a dynamic executable"
1129 @ MSG_MAP_NOEXVLSZ      "%s: %llu: value and size attributes are incompatible \
1130                          with extern or parent symbols"
1131 @ MSG_MAP_FLTR_ONLYAVL  "%s: %llu: symbol filtering is only available when \
1132                          building a shared object"

1134 @ MSG_MAP_SEGSAME       "segments ’%s’ and ’%s’ have the same assigned \
1135                          virtual address"
1136 @ MSG_MAP_EXCLIMIT      "exceeds internal limit"
1137 @ MSG_MAP_NOBADFRM      "number is badly formed"

1139 @ MSG_MAP_SEGTYP        "segment type"
1140 @ MSG_MAP_SEGVADDR      "segment virtual address"
1141 @ MSG_MAP_SEGPHYS       "segment physical address"
1142 @ MSG_MAP_SEGLEN        "segment length"
1143 @ MSG_MAP_SEGFLAG       "segment flags"
1144 @ MSG_MAP_SEGALIGN      "segment alignment"
1145 @ MSG_MAP_SEGROUND      "segment rounding"

1147 @ MSG_MAP_SECTYP        "section type"
1148 @ MSG_MAP_SECFLAG       "section flags"
1149 @ MSG_MAP_SECNAME       "section name"

1151 @ MSG_MAP_SYMVAL        "symbol value"
1152 @ MSG_MAP_SYMSIZE       "symbol size"

1154 @ MSG_MAP_DIFF_SYMVAL   "symbol values differ"
1155 @ MSG_MAP_DIFF_SYMSZ    "symbol sizes differ"
1156 @ MSG_MAP_DIFF_SYMTYP   "symbol types differ"
1157 @ MSG_MAP_DIFF_SYMNDX   "symbol indexes differ"
1158 @ MSG_MAP_DIFF_SYMLCL   "symbol scope conflict against local and non-local"
1159 @ MSG_MAP_DIFF_SYMGLOB  "symbol scope conflict against singleton/exported"
1160 @ MSG_MAP_DIFF_SYMPROT  "symbol scope conflict against protected"
1161 @ MSG_MAP_DIFF_SYMVER   "symbol version conflict"
1162 @ MSG_MAP_DIFF_SYMMUL   "symbol multiple definition"
1163 @ MSG_MAP_DIFF_SNGLDIR  "singleton scope and direct declaration are \
1164                          incompatible"
1165 @ MSG_MAP_DIFF_PROTNDIR "protected scope and no-direct declaration \
1166                          are incompatible"


1169 @ MSG_MAP_SECORDER      "section ordering requested, but no matching section \
1170                          found: segment: %s section: %s"


1173 # Mapfile Directives

1175 @ MSG_MAP_EXP_ATTR      "%s: %llu: expected attribute name (%s), or \
1176                          terminator (’;’, ’}’): %s"
1177 @ MSG_MAP_EXP_CAPMASK   "%s: %llu: expected capability name, integer value, or \
1178                          terminator (’;’, ’}’): %s"
1179 @ MSG_MAP_EXP_CAPNAME   "%s: %llu: expected name, or terminator (’;’, ’}’): %s"
1180 @ MSG_MAP_EXP_CAPID     "%s: %llu: expected name, or ’{’ following %s: %s"
1181 @ MSG_MAP_EXP_CAPHW     "%s: %llu: expected hardware capability, or \
1182                          terminator (’;’, ’}’): %s"
```

```
1183 @ MSG_MAP_EXP_CAPSF     "%s: %llu: expected software capability, or \
1184                          terminator (';', '}'): %s"
1185 @ MSG_MAP_EXP_EQ        "%s: %llu: expected '=' following %s: %s"
1186 @ MSG_MAP_EXP_EQ_ALL    "%s: %llu: expected '=', '+=', or '-=' following %s: %s"
1187 @ MSG_MAP_EXP_EQ_PEQ    "%s: %llu: expected '=' following %s: %s"
1188 @ MSG_MAP_EXP_DIR       "%s: %llu: expected mapfile directive (%s): %s"
1189 @ MSG_MAP_SFLG_EXBANG   "%s: %llu: '!' appears without corresponding flag"
1190 @ MSG_MAP_EXP_FILNAM    "%s: %llu: expected file name following %s: %s"
1191 @ MSG_MAP_EXP_FILPATH   "%s: %llu: expected file path following %s: %s"
1192 @ MSG_MAP_EXP_INT       "%s: %llu: expected integer value following %s: %s"
1193 @ MSG_MAP_EXP_LBKT      "%s: %llu: expected '{' following %s: %s"
1194 @ MSG_MAP_EXP_OBJNAM    "%s: %llu: expected object name following %s: %s"
1195 @ MSG_MAP_SFLG_ONEBANG  "%s: %llu: '!' can only be specified once per flag"
1196 @ MSG_MAP_EXP_SECFLAG   "%s: %llu: expected section flag (%s), '!', or \
1197                          terminator (';', '}'): %s"
1198 @ MSG_MAP_EXP_SECNAM    "%s: %llu: expected section name following %s: %s"
1199 @ MSG_MAP_EXP_SEGFLAG   "%s: %llu: expected segment flag (%s), or \
1200                          terminator (';', '}'): %s"
1201 @ MSG_MAP_EXP_ECNAM     "%s: %llu: expected entrance criteria (ASSIGN_SECTION) \
1202                          name, or terminator (';', '}'): %s"
1203 @ MSG_MAP_EXP_SEGNAM    "%s: %llu: expected segment name following %s: %s"
1204 @ MSG_MAP_EXP_SEM       "%s: %llu: expected ';' to terminate %s: %s"
1205 @ MSG_MAP_EXP_SEMLBKT   "%s: %llu: expected ';' or '{' following %s: %s"
1206 @ MSG_MAP_EXP_SEMRBKT   "%s: %llu: expected ';' or '}' to terminate %s: %s"
1207 @ MSG_MAP_EXP_SHTYPE    "%s: %llu: expected section type: %s"
1208 @ MSG_MAP_EXP_SYM       "%s: %llu: expected symbol name, symbol scope, \
1209                          or '*': %s"
1210 @ MSG_MAP_EXP_SYMEND    "%s: %llu: expected inherited version name, or \
1211                          terminator (';'): %s"
1212 @ MSG_MAP_EXP_SYMDELIM  "%s: %llu: expected one of ':', ';', or '{': %s"
1213 @ MSG_MAP_EXP_SYMFLAG   "%s: %llu: expected symbol flag (%s), or \
1214                          terminator (';', '}'): %s"
1215 @ MSG_MAP_EXP_SYMNAM    "%s: %llu: expected symbol name following %s: %s"
1216 @ MSG_MAP_EXP_SYMSCOPE  "%s: %llu: expected symbol scope (%s): %s"
1217 @ MSG_MAP_EXP_SYMTYPE   "%s: %llu: expected symbol type (%s): %s"
1218 @ MSG_MAP_EXP_VERSION   "%s: %llu: expected version name following %s: %s"
1219 @ MSG_MAP_BADEXTRA      "%s: %llu: unexpected text found following %s directive"
1220 @ MSG_MAP_VALUELIMIT    "%s: %llu: numeric value exceeds word size: %s"
1221 @ MSG_MAP_MALVALUE      "%s: %llu: malformed numeric value: %s"
1222 @ MSG_MAP_BADVALUETAIL  "%s: %llu: unexpected characters following numeric \
1223                          constant: %s"
1224 @ MSG_MAP_WSNEEDED      "%s: %llu: whitespace needed before token: %s"
1225 @ MSG_MAP_BADCHAR       "%s: %llu: unexpected text: %s"
1226 @ MSG_MAP_BADKWQUOTE    "%s: %llu: mapfile keywords should not be quoted: %s"
1227 @ MSG_MAP_CDIR_NOTBOL   "%s: %llu: mapfile control directive not at start of \
1228                          line: %s"
1229 @ MSG_MAP_NOATTR        "%s: %llu: %s specified no attributes (empty {})"
1230 @ MSG_MAP_NOVALUES      "%s: %llu: %s specified without values"
1231 @ MSG_MAP_INTERR        "<internal error>"
1232 @ MSG_MAP_ISORDVER      "%s: %llu: version 0 mapfile ?O flag and version 1 \
1233                          segment IS_ORDER attribute are mutually exclusive: %s"
1234 @ MSG_MAP_SYMATTR       "symbol attributes";

1236 # Mapfile Control Directives

1238 @ MSG_MAP_CDIR_BADVDIR  "%s: %llu: $mapfile_version directive must specify \
1239                          version 2 or higher: %d"
1240 @ MSG_MAP_CDIR_BADVER   "%s: %llu: unknown mapfile version: %d"
1241 @ MSG_MAP_CDIR_REPVER   "%s: %llu: $mapfile_version must be first directive \
1242                          in file"
1243 @ MSG_MAP_CDIR_REQARG   "%s: %llu: %s directive requires an argument"
1244 @ MSG_MAP_CDIR_REQNOARG "%s: %llu: %s directive does not accept arguments"
1245 @ MSG_MAP_CDIR_BAD      "%s: %llu: unrecognized mapfile control directive"
1246 @ MSG_MAP_CDIR_NOIF     "%s: %llu: %s directive used without opening $if"
1247 @ MSG_MAP_CDIR_ELSE     "%s: %llu: %s directive preceded by $else on line %d"
1248 @ MSG_MAP_CDIR_NOEND    "%s: %llu: EOF encountered without closing $endif \
```



```
1249                          for $if on line %d"
1250 @ MSG_MAP_CDIR_ERROR    "%s: %llu: error: %s"


1253 # Mapfile Conditional Expressions

1255 @ MSG_MAP_CEXP_TOKERR   "%s: %llu: syntax error in conditional expression at: %s
1256 @ MSG_MAP_CEXP_SEMERR   "%s: %llu: malformed conditional expression"
1257 @ MSG_MAP_CEXP_BADOPUSE "%s: %llu: invalid operator use in conditional \
1258                          expression"
1259 @ MSG_MAP_CEXP_UNBALPAR "%s: %llu: unbalanced parenthesis in conditional \
1260                          expression"
1261 @ MSG_MAP_BADCESC       "%s: %llu: unrecognized escape in double quoted \
1262                          token: \\%c\n"

1264 # Generic error diagnostic labels

1266 @ MSG_STR_NULL          "(null)"

1268 @ MSG_DBG_DFLT_FMT      "debug: "
1269 @ MSG_DBG_AOUT_FMT      "debug: a.out: "
1270 @ MSG_DBG_NAME_FMT      "debug: %s: "

1272 # -z assert-deflib strings

1274 @ MSG_ARG_ASSDEFLIB_MALFORMED   "library name malformed: %s"
1275 @ MSG_ARG_ASSDEFLIB_FOUND       "dynamic library found on default search path \
1276                                 (%s): lib%s.so"

1278 @ _END_


1281 # Software identification.  Note, the SGU strings is historic, and has
1282 # little relevance.  It is preserved as applications have used this
1283 # string to identify the Solaris link-editor.

1285 @ MSG_SGS_ID            "ld: Software Generation Utilities - \
1286                          Solaris Link Editors: "

1288 # The following strings represent reserved words, files, pathnames and symbols.
1289 # Reference to this strings is via the MSG_ORIG() macro, and thus no message
1290 # translation is required.

1292 @ MSG_DBG_FOPEN_MODE    "w"

1294 @ MSG_DBG_CLS32_FMT     "32: "
1295 @ MSG_DBG_CLS64_FMT     "64: "

1297 @ MSG_STR_PATHTOK       ";:"
1298 @ MSG_STR_AOUT          "a.out"

1300 @ MSG_STR_LIB_A         "%s/lib%s.a"
1301 @ MSG_STR_LIB_SO        "%s/lib%s.so"
1302 @ MSG_STR_PATH          "%s/%s"
1303 @ MSG_STR_STRNL         "%s\n"
1304 @ MSG_STR_NL            "\n"
1305 @ MSG_STR_CAPGROUPID    "CAP_GROUP_%d"

1307 @ MSG_STR_LD_DYNAMIC    "dynamic"
1308 @ MSG_STR_SYMBOLIC      "symbolic"
1309 @ MSG_STR_ELIMINATE     "eliminate"
1310 @ MSG_STR_LOCAL         "local"
1311 @ MSG_STR_PROGBITS      "progbits"
1312 @ MSG_STR_SYMTAB        "symtab"
1313 @ MSG_STR_DYNSYM        "dynsym"
1314 @ MSG_STR_REL           "rel"
```

```
1315 @ MSG_STR_RELA          "rela"
1316 @ MSG_STR_STRTAB        "strtab"
1317 @ MSG_STR_HASH          "hash"
1318 @ MSG_STR_LIB           "lib"
1319 @ MSG_STR_NOTE          "note"
1320 @ MSG_STR_NOBITS        "nobits"
1321 @ MSG_STR_HWCAP_1       "hwcap_1"
1322 @ MSG_STR_SFCAP_1       "sfcap_1"
1323 @ MSG_STR_SOEXT         ".so"

1325 @ 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:\
1326                          S:VW:Y:?"

1328 # Argument processing strings

1330 @ MSG_ARG_3             "-3"
1331 @ MSG_ARG_6             "-6"
1332 @ MSG_ARG_A             "-a"
1333 @ MSG_ARG_B             "-b"
1334 @ MSG_ARG_CB            "-B"
1335 @ MSG_ARG_BDIRECT       "-Bdirect"
1336 @ MSG_ARG_BDYNAMIC      "-Bdynamic"
1337 @ MSG_ARG_BELIMINATE    "-Beliminate"
1338 @ MSG_ARG_BGROUP        "-Bgroup"
1339 @ MSG_ARG_BLOCAL        "-Blocal"
1340 @ MSG_ARG_BNODIRECT     "-Bnodirect"
1341 @ MSG_ARG_BSYMBOLIC     "-Bsymbolic"
1342 @ MSG_ARG_BTRANSLATOR   "-Btranslator"
1343 @ MSG_ARG_C             "-c"
1344 @ MSG_ARG_D             "-d"
1345 @ MSG_ARG_DY            "-dy"
1346 @ MSG_ARG_CI            "-I"
1347 @ MSG_ARG_CN            "-N"
1348 @ MSG_ARG_P             "-p"
1349 @ MSG_ARG_CP            "-P"
1350 @ MSG_ARG_CQ            "-Q"
1351 @ MSG_ARG_CY            "-Y"
1352 @ MSG_ARG_CYL           "-YL"
1353 @ MSG_ARG_CYP           "-YP"
1354 @ MSG_ARG_CYU           "-YU"
1355 @ MSG_ARG_Z             "-z"
1356 @ MSG_ARG_ZDEFNODEF     "-z[defs|nodefs]"
1357 @ MSG_ARG_ZGUIDE        "-zguidance"
1358 @ MSG_ARG_ZNODEF        "-znodefs"
1359 @ MSG_ARG_ZNOINTERP     "-znointerp"
1360 @ MSG_ARG_ZRELAXRELOC   "-zrelaxreloc"
1361 @ MSG_ARG_ZNORELAXRELOC "-znorelaxreloc"
1362 @ MSG_ARG_ZTEXT         "-ztext"
1363 @ MSG_ARG_ZTEXTOFF      "-ztextoff"
1364 @ MSG_ARG_ZTEXTWARN     "-ztextwarn"
1365 @ MSG_ARG_ZTEXTALL      "-z[text|textwarn|textoff]"
1366 @ MSG_ARG_ZLOADFLTR     "-zloadfltr"
1367 @ MSG_ARG_ZCOMBRELOC    "-zcombreloc"
1368 @ MSG_ARG_ZSYMBOLCAP    "-zsymbolcap"
1369 @ MSG_ARG_ZFATWNOFATW   "-z[fatal-warnings|nofatalwarnings]"

1371 @ MSG_ARG_ABSEXEC       "absexec"
1372 @ MSG_ARG_ALTEXEC64     "altexec64"
1373 @ MSG_ARG_NOCOMPSTRTAB  "nocompstrtab"
1374 @ MSG_ARG_GROUPPERM     "groupperm"
1375 @ MSG_ARG_NOGROUPPERM   "nogroupperm"
1376 @ MSG_ARG_LAZYLOAD      "lazyload"
1377 @ MSG_ARG_NOLAZYLOAD    "nolazyload"
1378 @ MSG_ARG_INTERPOSE     "interpose"
1379 @ MSG_ARG_DIRECT        "direct"
1380 @ MSG_ARG_NODIRECT      "nodirect"
```



```
1381 @ MSG_ARG_IGNORE        "ignore"
1382 @ MSG_ARG_RECORD        "record"
1383 @ MSG_ARG_INITFIRST     "initfirst"
1384 @ MSG_ARG_INITARRAY     "initarray="
1385 @ MSG_ARG_FINIARRAY     "finiarray="
1386 @ MSG_ARG_PREINITARRAY  "preinitarray="
1387 @ MSG_ARG_RTLDINFO      "rtldinfo="
1388 @ MSG_ARG_DTRACE        "dtrace="
1389 @ MSG_ARG_TRANSLATOR    "translator"
1390 @ MSG_ARG_NOOPEN        "nodlopen"
1391 @ MSG_ARG_NOW           "now"
1392 @ MSG_ARG_ORIGIN        "origin"
1393 @ MSG_ARG_DEFS          "defs"
1394 @ MSG_ARG_NODEFS        "nodefs"
1395 @ MSG_ARG_NODUMP        "nodump"
1396 @ MSG_ARG_NOVERSION     "noversion"
1397 @ MSG_ARG_TEXT          "text"
1398 @ MSG_ARG_TEXTOFF       "textoff"
1399 @ MSG_ARG_TEXTWARN      "textwarn"
1400 @ MSG_ARG_MULDEFS       "muldefs"
1401 @ MSG_ARG_NODELETE      "nodelete"
1402 @ MSG_ARG_NOINTERP      "nointerp"
1403 @ MSG_ARG_NOPARTIAL     "nopartial"
1404 @ MSG_ARG_NORELOC       "noreloc"
1405 @ MSG_ARG_REDLOCSYM     "redlocsym"
1406 @ MSG_ARG_VERBOSE       "verbose"
1407 @ MSG_ARG_WEAKEXT       "weakextract"
1408 @ MSG_ARG_LOADFLTR      "loadfltr"
1409 @ MSG_ARG_ALLEXTRT      "allextract"
1410 @ MSG_ARG_DFLEXTRT      "defaultextract"
1411 @ MSG_ARG_COMBRELOC     "combreloc"
1412 @ MSG_ARG_NOCOMBRELOC   "nocombreloc"
1413 @ MSG_ARG_NODEFAULTLIB  "nodefaultlib"
1414 @ MSG_ARG_ENDFILTEE     "endfiltee"
1415 @ MSG_ARG_LD32          "ld32="
1416 @ MSG_ARG_LD64          "ld64="
1417 @ MSG_ARG_RESCAN        "rescan"
1418 @ MSG_ARG_RESCAN_NOW    "rescan-now"
1419 @ MSG_ARG_RESCAN_START  "rescan-start"
1420 @ MSG_ARG_RESCAN_END    "rescan-end"
1421 @ MSG_ARG_GUIDE         "guidance"
1422 @ MSG_ARG_NOLDYNSYM     "noldynsym"
1423 @ MSG_ARG_RELAXRELOC    "relaxreloc"
1424 @ MSG_ARG_NORELAXRELOC  "norelaxreloc"
1425 @ MSG_ARG_NOSIGHANDLER  "nosighandler"
1426 @ MSG_ARG_GLOBAUDIT     "globalaudit"
1427 @ MSG_ARG_TARGET        "target="
1428 @ MSG_ARG_WRAP          "wrap="
1429 @ MSG_ARG_FATWARN       "fatal-warnings"
1430 @ MSG_ARG_NOFATWARN     "nofatal-warnings"
1431 @ MSG_ARG_HELP          "help"
1432 @ MSG_ARG_GROUP         "group"
1433 @ MSG_ARG_REDUCE        "reduce"
1434 @ MSG_ARG_STATIC        "static"
1435 @ MSG_ARG_SYMBOLCAP     "symbolcap"
1436 @ MSG_ARG_DEFERRED      "deferred"
1437 @ MSG_ARG_NODEFERRED    "nodeferred"
1438 @ MSG_ARG_ASSDEFLIB     "assert-deflib"

1440 @ MSG_ARG_LCOM          "L,"
1441 @ MSG_ARG_PCOM          "P,"
1442 @ MSG_ARG_UCOM          "U,"

1444 @ MSG_ARG_T_RPATH       "rpath"
1445 @ MSG_ARG_T_SHARED      "shared"
1446 @ MSG_ARG_T_SONAME      "soname"
```

```
1447 @ MSG_ARG_T_WL          "l,-"

1449 @ MSG_ARG_T_ASNEEDED    "-as-needed"
1450 @ MSG_ARG_T_AUXFLTR     "-auxiliary"
1451 @ MSG_ARG_T_MULDEFS     "-allow-multiple-definition"
1452 @ MSG_ARG_T_INTERP      "-dynamic-linker"
1453 @ MSG_ARG_T_ENDGROUP    "-end-group"
1454 @ MSG_ARG_T_ENTRY       "-entry"
1455 @ MSG_ARG_T_STDFLTR     "-filter"
1456 @ MSG_ARG_T_FATWARN     "-fatal-warnings"
1457 @ MSG_ARG_T_NOFATWARN   "-no-fatal-warnings"
1458 @ MSG_ARG_T_HELP        "-help"
1459 @ MSG_ARG_T_LIBRARY     "-library"
1460 @ MSG_ARG_T_LIBPATH     "-library-path"
1461 @ MSG_ARG_T_NOASNEEDED  "-no-as-needed"
1462 @ MSG_ARG_T_NOUNDEF     "-no-undefined"
1463 @ MSG_ARG_T_NOWHOLEARC  "-no-whole-archive"
1464 @ MSG_ARG_T_OUTPUT      "-output"
1465 @ MSG_ARG_T_RELOCATABLE "-relocatable"
1466 @ MSG_ARG_T_STARTGROUP  "-start-group"
1467 @ MSG_ARG_T_STRIP       "-strip-all"
1468 @ MSG_ARG_T_UNDEF       "-undefined"
1469 @ MSG_ARG_T_VERSION     "-version"
1470 @ MSG_ARG_T_WHOLEARC    "-whole-archive"
1471 @ MSG_ARG_T_WRAP        "-wrap"
1472 @ MSG_ARG_T_OPAR        "("
1473 @ MSG_ARG_T_CPAR        ")"

1475 # -z guidance=item strings
1476 @ MSG_ARG_GUIDE_DELIM          ",: \t"
1477 @ MSG_ARG_GUIDE_NO_ALL         "noall"
1478 @ MSG_ARG_GUIDE_NO_DEFS        "nodefs"
1479 @ MSG_ARG_GUIDE_NO_DIRECT      "nodirect"
1480 @ MSG_ARG_GUIDE_NO_LAZYLOAD    "nolazyload"
1481 @ MSG_ARG_GUIDE_NO_MAPFILE     "nomapfile"
1482 @ MSG_ARG_GUIDE_NO_TEXT        "notext"
1483 @ MSG_ARG_GUIDE_NO_UNUSED      "nounused"

1485 # Environment variable strings

1487 @ MSG_LD_RUN_PATH       "LD_RUN_PATH"
1488 @ MSG_LD_LIBPATH_32     "LD_LIBRARY_PATH_32"
1489 @ MSG_LD_LIBPATH_64     "LD_LIBRARY_PATH_64"
1490 @ MSG_LD_LIBPATH        "LD_LIBRARY_PATH"

1492 @ MSG_LD_NOVERSION_32   "LD_NOVERSION_32"
1493 @ MSG_LD_NOVERSION_64   "LD_NOVERSION_64"
1494 @ MSG_LD_NOVERSION      "LD_NOVERSION"

1496 @ MSG_SGS_SUPPORT_32    "SGS_SUPPORT_32"
1497 @ MSG_SGS_SUPPORT_64    "SGS_SUPPORT_64"
1498 @ MSG_SGS_SUPPORT       "SGS_SUPPORT"


1501 # Symbol names

1503 @ MSG_SYM_LIBVER_U      "_lib_version"


1506 # Mapfile tokens

1508 @ MSG_MAP_LOAD          "load"
1509 @ MSG_MAP_NOTE          "note"
1510 @ MSG_MAP_NULL          "null"
1511 @ MSG_MAP_STACK         "stack"
1512 @ MSG_MAP_ADDVERS       "addvers"
```



```
1513 @ MSG_MAP_FUNCTION      "function"
1514 @ MSG_MAP_DATA          "data"
1515 @ MSG_MAP_COMMON        "common"
1516 @ MSG_MAP_PARENT        "parent"
1517 @ MSG_MAP_EXTERN        "extern"
1518 @ MSG_MAP_DIRECT        "direct"
1519 @ MSG_MAP_NODIRECT      "nodirect"
1520 @ MSG_MAP_FILTER        "filter"
1521 @ MSG_MAP_AUXILIARY     "auxiliary"
1522 @ MSG_MAP_OVERRIDE      "override"
1523 @ MSG_MAP_INTERPOSE     "interpose"
1524 @ MSG_MAP_DYNSORT       "dynsort"
1525 @ MSG_MAP_NODYNSORT     "nodynsort"

1527 @ MSG_MAPKW_ALIGN               "ALIGN"
1528 @ MSG_MAPKW_ALLOC               "ALLOC"
1529 @ MSG_MAPKW_ALLOW               "ALLOW"
1530 @ MSG_MAPKW_AMD64_LARGE         "AMD64_LARGE"
1531 @ MSG_MAPKW_ASSIGN_SECTION      "ASSIGN_SECTION"
1532 @ MSG_MAPKW_AUX                 "AUXILIARY"
1533 @ MSG_MAPKW_CAPABILITY          "CAPABILITY"
1534 @ MSG_MAPKW_COMMON              "COMMON"
1535 @ MSG_MAPKW_DATA                "DATA"
1536 @ MSG_MAPKW_DEFAULT             "DEFAULT"
1537 @ MSG_MAPKW_DEPEND_VERSIONS     "DEPEND_VERSIONS"
1538 @ MSG_MAPKW_DIRECT              "DIRECT"
1539 @ MSG_MAPKW_DISABLE             "DISABLE"
1540 @ MSG_MAPKW_DYNSORT             "DYNSORT"
1541 @ MSG_MAPKW_ELIMINATE           "ELIMINATE"
1542 @ MSG_MAPKW_EXECUTE             "EXECUTE"
1543 @ MSG_MAPKW_EXPORTED            "EXPORTED"
1544 @ MSG_MAPKW_EXTERN              "EXTERN"
1545 @ MSG_MAPKW_FILTER              "FILTER"
1546 @ MSG_MAPKW_FILE_BASENAME       "FILE_BASENAME"
1547 @ MSG_MAPKW_FILE_PATH           "FILE_PATH"
1548 @ MSG_MAPKW_FILE_OBJNAME        "FILE_OBJNAME"
1549 @ MSG_MAPKW_FUNCTION            "FUNCTION"
1550 @ MSG_MAPKW_FLAGS               "FLAGS"
1551 @ MSG_MAPKW_GLOBAL              "GLOBAL"
1552 @ MSG_MAPKW_INTERPOSE           "INTERPOSE"
1553 @ MSG_MAPKW_HIDDEN              "HIDDEN"
1554 @ MSG_MAPKW_HDR_NOALLOC         "HDR_NOALLOC"
1555 @ MSG_MAPKW_HW                  "HW"
1556 @ MSG_MAPKW_HW_1                "HW_1"
1557 @ MSG_MAPKW_HW_2                "HW_2"
1558 @ MSG_MAPKW_IS_NAME             "IS_NAME"
1559 @ MSG_MAPKW_IS_ORDER            "IS_ORDER"
1560 @ MSG_MAPKW_LOAD_SEGMENT        "LOAD_SEGMENT"
1561 @ MSG_MAPKW_LOCAL               "LOCAL"
1562 @ MSG_MAPKW_MACHINE             "MACHINE"
1563 @ MSG_MAPKW_MAX_SIZE            "MAX_SIZE"
1564 @ MSG_MAPKW_NOHDR               "NOHDR"
1565 @ MSG_MAPKW_NODIRECT            "NODIRECT"
1566 @ MSG_MAPKW_NODYNSORT           "NODYNSORT"
1567 @ MSG_MAPKW_NOTE_SEGMENT        "NOTE_SEGMENT"
1568 @ MSG_MAPKW_NULL_SEGMENT        "NULL_SEGMENT"
1569 @ MSG_MAPKW_OS_ORDER            "OS_ORDER"
1570 @ MSG_MAPKW_PADDR               "PADDR"
1571 @ MSG_MAPKW_PARENT              "PARENT"
1572 @ MSG_MAPKW_PHDR_ADD_NULL       "PHDR_ADD_NULL"
1573 @ MSG_MAPKW_PLATFORM            "PLATFORM"
1574 @ MSG_MAPKW_PROTECTED           "PROTECTED"
1575 @ MSG_MAPKW_READ                "READ"
1576 @ MSG_MAPKW_ROUND               "ROUND"
1577 @ MSG_MAPKW_REQUIRE             "REQUIRE"
1578 @ MSG_MAPKW_SEGMENT_ORDER       "SEGMENT_ORDER"
```

```
1579 @ MSG_MAPKW_SF                  "SF"
1580 @ MSG_MAPKW_SF_1                "SF_1"
1581 @ MSG_MAPKW_SINGLETON           "SINGLETON"
1582 @ MSG_MAPKW_SIZE                "SIZE"
1583 @ MSG_MAPKW_SIZE_SYMBOL         "SIZE_SYMBOL"
1584 @ MSG_MAPKW_STACK               "STACK"
1585 @ MSG_MAPKW_SYMBOL_SCOPE        "SYMBOL_SCOPE"
1586 @ MSG_MAPKW_SYMBOL_VERSION      "SYMBOL_VERSION"
1587 @ MSG_MAPKW_SYMBOLIC            "SYMBOLIC"
1588 @ MSG_MAPKW_TYPE                "TYPE"
1589 @ MSG_MAPKW_VADDR               "VADDR"
1590 @ MSG_MAPKW_VALUE               "VALUE"
1591 @ MSG_MAPKW_WRITE               "WRITE"


1594 @ MSG_STR_DTRACE        "PT_SUNWDTRACE"
```

```
**********************************************************
   20683 Sun May  4 06:58:06 2014
new/usr/src/cmd/sgs/libld/common/util.c
4843 add --as-needed and --no-as-needed to ld
**********************************************************
______unchanged_portion_omitted_

 386 /*
 387  * Parse an individual option.  The intent of this function is to determine if
 388  * any known, non-Solaris options have been passed to ld(1).  This condition
 389  * can occur as a result of build configuration tools, because of users
 390  * familiarity with other systems, or simply the users preferences.  If a known
 391  * non-Solaris option can be determined, translate that option into the Solaris
 392  * counterpart.
 393  *
 394  * This function will probably never be a complete solution, as new, non-Solaris
 395  * options are discovered, their translation will have to be added.  Other
 396  * non-Solaris options are incompatible with the Solaris link-editor, and will
 397  * never be recognized.  We support what we can.
 398  */
 399 int
 400 ld_getopt(Lm_list *lml, int ndx, int argc, char **argv)
 401 {
 402         int     c;

 404         if ((optind < argc) && argv[optind] && (argv[optind][0] == '-')) {
 405                 char    *arg = &argv[optind][1];

 407                 switch (*arg) {
 408                 case 'r':
 409                         /* Translate -rpath <optarg> to -R <optarg> */
 410                         if ((c = str2chr(lml, ndx, argc, argv, arg, 'R',
 411                             MSG_ORIG(MSG_ARG_T_RPATH),
 412                             MSG_ARG_T_RPATH_SIZE, NULL)) != 0) {
 413                                 return (c);
 414                         }
 415                         break;
 416                 case 's':
 417                         /* Translate -shared to -G */
 418                         if ((c = str2chr(lml, ndx, argc, argv, arg, 'G',
 419                             MSG_ORIG(MSG_ARG_T_SHARED), 0, NULL)) != 0) {
 420                                 return (c);

 422                         /* Translate -soname <optarg> to -h <optarg> */
 423                         } else if ((c = str2chr(lml, ndx, argc, argv, arg, 'h',
 424                             MSG_ORIG(MSG_ARG_T_SONAME),
 425                             MSG_ARG_T_SONAME_SIZE, NULL)) != 0) {
 426                                 return (c);
 427                         }
 428                         break;
 429                 case 'w':
 430                         /* Translate -wrap to -z wrap= */
 431                         if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
 432                             MSG_ORIG(MSG_ARG_T_WRAP) + 1,
 433                             MSG_ARG_T_WRAP_SIZE - 1, str2chr_wrap_cb)) != 0) {
 434                                 return (c);
 435                         }
 436                         break;
 437                 case '(':
 438                         /*
 439                          * Translate -( to -z rescan-start
 440                          */
 441                         if ((c = str2chr(lml, ndx, argc, argv,
 442                             arg, 'z', MSG_ORIG(MSG_ARG_T_OPAR), 0, NULL)) !=
 443                             0) {
 444                                 optarg = (char *)MSG_ORIG(MSG_ARG_RESCAN_START);
```



```
 445                                 return (c);
 446                         }
 447                         break;
 448                 case ')':
 449                         /*
 450                          * Translate -) to -z rescan-end
 451                          */
 452                         if ((c = str2chr(lml, ndx, argc, argv,
 453                             arg, 'z', MSG_ORIG(MSG_ARG_T_CPAR), 0, NULL)) !=
 454                             0) {
 455                                 optarg = (char *)MSG_ORIG(MSG_ARG_RESCAN_END);
 456                                 return (c);
 457                         }
 458                         break;
 459                 case '-':
 460                         switch (*(arg + 1)) {
 461                         case 'a':
 462                                 /*
 463                                  * Translate --as-needed to -zignore
 464                                  */
 465                                 if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
 466                                     MSG_ORIG(MSG_ARG_T_ASNEEDED), 0, NULL)) !=
 467                                     0) {
 468                                         optarg =
 469                                             (char *)MSG_ORIG(MSG_ARG_IGNORE);
 470                                         return (c);
 471                                 }
 472                                 /*
 473                                  * Translate --allow-multiple-definition to
 474                                  * -zmuldefs
 475                                  */
 476                                 if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
 477                                     MSG_ORIG(MSG_ARG_T_MULDEFS), 0, NULL)) !=
 478                                     0) {
 479                                         optarg =
 480                                             (char *)MSG_ORIG(MSG_ARG_MULDEFS);
 481                                         return (c);
 482                                 }

 483                                 /*
 484                                  * Translate --auxiliary <optarg> to
 485                                  * -f <optarg>
 486                                  */
 487                                 if ((c = str2chr(lml, argc, ndx, argv,
 477                                 } else if ((c = str2chr(lml, argc, ndx, argv,
 488                                     arg, 'f', MSG_ORIG(MSG_ARG_T_AUXFLTR),
 489                                     MSG_ARG_T_AUXFLTR_SIZE, NULL)) != 0) {
 490                                         return (c);
 491                                 }
 492                                 break;
 493                         case 'd':
 494                                 /*
 495                                  * Translate --dynamic-linker <optarg> to
 496                                  * -I <optarg>
 497                                  */
 498                                 if ((c = str2chr(lml, ndx, argc, argv, arg, 'I',
 499                                     MSG_ORIG(MSG_ARG_T_INTERP),
 500                                     MSG_ARG_T_INTERP_SIZE, NULL)) != 0) {
 501                                         return (c);
 502                                 }
 503                                 break;
 504                         case 'e':
 505                                 /* Translate --entry <optarg> to -e <optarg> */
 506                                 if ((c = str2chr(lml, ndx, argc, argv, arg, 'e',
 507                                     MSG_ORIG(MSG_ARG_T_ENTRY),
 508                                     MSG_ARG_T_ENTRY_SIZE, NULL)) != 0) {
```

```
509                                     return (c);
510                             }
511                             /*
512                              * Translate --end-group to -z rescan-end
513                              */
514                             if ((c = str2chr(lml, ndx, argc, argv,
515                                 arg, 'z', MSG_ORIG(MSG_ARG_T_ENDGROUP),
516                                 0, NULL)) != 0) {
517                                     optarg = (char *)
518                                         MSG_ORIG(MSG_ARG_RESCAN_END);
519                                     return (c);
520                             }
521                             break;
522                     case 'f':
523                             /*
524                              * Translate --fatal-warnings to
525                              * -z fatal-warnings.
526                              */
527                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
528                                 MSG_ORIG(MSG_ARG_T_FATWARN),
529                                 0, NULL)) != 0) {
530                                     optarg = (char *)
531                                         MSG_ORIG(MSG_ARG_FATWARN);
532                                     return (c);
533                             }
534                             /* Translate --filter <optarg> to -F <optarg> */
535                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'F',
536                                 MSG_ORIG(MSG_ARG_T_STDFLTR),
537                                 MSG_ARG_T_STDFLTR_SIZE, NULL)) != 0) {
538                                     return (c);
539                             }
540                             break;
541                     case 'h':
542                             /* Translate --help to -zhelp */
543                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
544                                 MSG_ORIG(MSG_ARG_T_HELP), 0, NULL)) !=
545                                 0) {
546                                     optarg = (char *)MSG_ORIG(MSG_ARG_HELP);
547                                     return (c);
548                             }
549                             break;
550                     case 'l':
551                             /*
552                              * Translate --library <optarg> to -l <optarg>
553                              */
554                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'l',
555                                 MSG_ORIG(MSG_ARG_T_LIBRARY),
556                                 MSG_ARG_T_LIBRARY_SIZE, NULL)) != 0) {
557                                     return (c);

559                             /*
560                              * Translate --library-path <optarg> to
561                              * -L <optarg>
562                              */
563                             } else if ((c = str2chr(lml, ndx, argc, argv,
564                                 arg, 'L', MSG_ORIG(MSG_ARG_T_LIBPATH),
565                                 MSG_ARG_T_LIBPATH_SIZE, NULL)) != 0) {
566                                     return (c);
567                             }
568                             break;
569                     case 'n':
570                             /*
571                              * Translate --no-as-needed to -zrecord
572                              */
573                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
574                                 MSG_ORIG(MSG_ARG_T_NOASNEEDED), 0, NULL)) !=
```



```
575                                 0) {
576                                     optarg =
577                                         (char *)MSG_ORIG(MSG_ARG_RECORD);
578                                     return (c);
579                             }
580                             /*
581                              * Translate --no-fatal-warnings to
582                              * -z nofatal-warnings.
583                              */
584                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
585                                 MSG_ORIG(MSG_ARG_T_NOFATWARN),
586                                 0, NULL)) != 0) {
587                                     optarg = (char *)
588                                         MSG_ORIG(MSG_ARG_NOFATWARN);
589                                     return (c);
590                             }

592                             /* Translate --no-undefined to -zdefs */
593                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'z',
594                                 MSG_ORIG(MSG_ARG_T_NOUNDEF), 0, NULL)) !=
595                                 0) {
596                                     optarg = (char *)MSG_ORIG(MSG_ARG_DEFS);
597                                     return (c);

599                             /*
600                              * Translate --no-whole-archive to
601                              * -z defaultextract
602                              */
603                             } else if ((c = str2chr(lml, ndx, argc, argv,
604                                 arg, 'z', MSG_ORIG(MSG_ARG_T_NOWHOLEARC),
605                                 0, NULL)) != 0) {
606                                     optarg =
607                                         (char *)MSG_ORIG(MSG_ARG_DFLEXTRT);
608                                     return (c);
609                             }
610                             break;
611                     case 'o':
612                             /* Translate --output <optarg> to -o <optarg> */
613                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'o',
614                                 MSG_ORIG(MSG_ARG_T_OUTPUT),
615                                 MSG_ARG_T_OUTPUT_SIZE, NULL)) != 0) {
616                                     return (c);
617                             }
618                             break;
619                     case 'r':
620                             /* Translate --relocatable to -r */
621                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'r',
622                                 MSG_ORIG(MSG_ARG_T_RELOCATABLE), 0,
623                                 NULL)) != 0) {
624                                     return (c);
625                             }
626                             break;
627                     case 's':
628                             /* Translate --strip-all to -s */
629                             if ((c = str2chr(lml, ndx, argc, argv, arg, 's',
630                                 MSG_ORIG(MSG_ARG_T_STRIP), 0, NULL)) !=
631                                 0) {
632                                     return (c);
633                             }
634                             /*
635                              * Translate --start-group to -z rescan-start
636                              */
637                             if ((c = str2chr(lml, ndx, argc, argv,
638                                 arg, 'z', MSG_ORIG(MSG_ARG_T_STARTGROUP),
639                                 0, NULL)) != 0) {
640                                     optarg = (char *)
```

```
641                                         MSG_ORIG(MSG_ARG_RESCAN_START);
642                                 return (c);
643                             }
644                             break;
645                         case 'u':
646                             /*
647                              * Translate --undefined <optarg> to
648                              * -u <optarg>
649                              */
650                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'u',
651                                 MSG_ORIG(MSG_ARG_T_UNDEF),
652                                 MSG_ARG_T_UNDEF_SIZE, NULL)) != 0) {
653                                 return (c);
654                             }
655                             break;
656                         case 'v':
657                             /* Translate --version to -V */
658                             if ((c = str2chr(lml, ndx, argc, argv, arg, 'V',
659                                 MSG_ORIG(MSG_ARG_T_VERSION), 0, NULL)) !=
660                                 0) {
661                                 return (c);
662                             }
663                             break;
664                         case 'w':
665                             /*
666                              * Translate --whole-archive to -z alltextract
667                              */
668                             if ((c = str2chr(lml, ndx, argc, argv,
669                                 arg, 'z', MSG_ORIG(MSG_ARG_T_WHOLEARC),
670                                 0, NULL)) != 0) {
671                                 optarg =
672                                     (char *)MSG_ORIG(MSG_ARG_ALLEXTRT);
673                                 return (c);
674                             }
675                             /*
676                              * Translate --wrap to -z wrap=
677                              */
678                             if ((c = str2chr(lml, ndx, argc, argv,
679                                 arg, 'z', MSG_ORIG(MSG_ARG_T_WRAP),
680                                 MSG_ARG_T_WRAP_SIZE, str2chr_wrap_cb)) !=
681                                 0) {
682                                 return (c);
683                             }
684                             break;
685                     }
686                     break;
687             }
688         }

690         if ((c = getopt(argc, argv, MSG_ORIG(MSG_STR_OPTIONS))) != -1) {
691                 /*
692                  * It is possible that a "-Wl," argument has been used to
693                  * specify an option.  This isn't advertized ld(1) syntax, but
694                  * compiler drivers and configuration tools, have been known to
695                  * pass this compiler option to ld(1).  Strip off the "-Wl,"
696                  * prefix and pass the option through.
697                  */
698                 if ((c == 'W') && (strncmp(optarg,
699                     MSG_ORIG(MSG_ARG_T_WL), MSG_ARG_T_WL_SIZE) == 0)) {
700                     DBG_CALL(Dbg_args_Wldel(lml, ndx, optarg));
701                     c = optarg[MSG_ARG_T_WL_SIZE];
702                     optarg += MSG_ARG_T_WL_SIZE + 1;
703                 }
704         }

706         return (c);
```



```
707 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   59105 Sun May  4 06:58:07 2014
new/usr/src/man/man1/ld.1
4843 add --as-needed and --no-as-needed to ld
**********************************************************
   1 '\" te
   2 .\" Copyright 1989 AT&T
   3 .\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved
   4 .\" Copyright (c) 2012, Joyent, Inc. All Rights Reserved
   5 .\" The contents of this file are subject to the terms of the Common Development
   6 .\"  See the License for the specific language governing permissions and limitat
   7 .\" the fields enclosed by brackets "[]" replaced with your own identifying info
   8 .TH LD 1 "Sep 10, 2013"
   9 .SH NAME
  10 ld \- link-editor for object files
  11 .SH SYNOPSIS
  12 .LP
  13 .nf
  14 \fBld\fR [\fB-32\fR | \fB-64\fR] [\fB-a\fR | \fB-r\fR] [\fB-b\fR] [\fB-B\fRdirec
  15 [\fB-B\fR dynamic | static] [\fB-B\fR eliminate] [\fB-B\fR group] [\fB-B\fR loca
  16 [\fB-B\fR reduce] [\fB-B\fR symbolic] [\fB-c\fR \fIname\fR] [\fB-C\fR] [\fB-d\fR
  17 [\fB-D\fR \fItoken\fR,...] [\fB-e\fR \fIepsym\fR] [\fB-f\fR \fIname\fR | \fB-F\f
  18 [\fB-i\fR] [\fB-I\fR \fIname\fR] [\fB-l\fR \fIx\fR] [\fB-L\fR \fIpath\fR] [\fB-m
  19 [\fB-N\fR \fIstring\fR] [\fB-o\fR \fIoutfile\fR] [\fB-p\fR \fIauditlib\fR] [\fB-
  20 [\fB-Q\fR y | n] [\fB-R\fR \fIpath\fR] [\fB-s\fR] [\fB-S\fR \fIsupportlib\fR] [\
  21 [\fB-u\fR \fIsymname\fR] [\fB-V\fR] [\fB-Y P\fR\fI,dirlist\fR] [\fB-z\fR absexec
  22 [\fB-z\fR allextract | defaultextract | weakextract ] [\fB-z\fR altexec64]
  23 [\fB-z\fR assert-deflib ] [ \fB-z\fR assert-deflib=\fIlibname\fR ]
  24 [\fB-z\fR combreloc | nocombreloc ] [\fB-z\fR defs | nodefs]
  25 [\fB-z\fR direct | nodirect] [\fB-z\fR endfiltee]
  26 [\fB-z\fR fatal-warnings | nofatal-warnings ] [\fB-z\fR finiarray=\fIfunction\fR
  27 [\fB-z\fR globalaudit] [\fB-z\fR groupperm | nogroupperm]
  28 [\fB-z\fR guidance[=\fIid1\fR,\fIid2\fR...] [\fB-z\fR help ]
  29 [\fB-z\fR ignore | record] [\fB-z\fR initarray=\fIfunction\fR] [\fB-z\fR initfir
  30 [\fB-z\fR interpose] [\fB-z\fR lazyload | nolazyload]
  31 [\fB-z\fR ld32=\fIarg1\fR,\fIarg2\fR,...] [\fB-z\fR ld64=\fIarg1\fR,\fIarg2\fR,.
  32 [\fB-z\fR loadfltr] [\fB-z\fR muldefs] [\fB-z\fR nocompstrtab] [\fB-z\fR nodefau
  33 [\fB-z\fR nodelete] [\fB-z\fR nodlopen] [\fB-z\fR nodump] [\fB-z\fR noldynsym]
  34 [\fB-z\fR nopartial] [\fB-z\fR noversion] [\fB-z\fR now] [\fB-z\fR origin]
  35 [\fB-z\fR preinitarray=\fIfunction\fR] [\fB-z\fR redlocsym] [\fB-z\fR relaxreloc
  36 [\fB-z\fR rescan-now] [\fB-z\fR recan] [\fB-z\fR rescan-start \fI\&...\fR \fB-z\
  37 [\fB-z\fR target=sparc|x86] [\fB-z\fR text | textwarn | textoff]
  38 [\fB-z\fR verbose] [\fB-z\fR wrap=\fIsymbol\fR] \fIfilename\fR...
  39 .fi

  41 .SH DESCRIPTION
  42 .sp
  43 .LP
  44 The link-editor, \fBld\fR, combines relocatable object files by resolving
  45 symbol references to symbol definitions, together with performing relocations.
  46 \fBld\fR operates in two modes, static or dynamic, as governed by the \fB-d\fR
  47 option. In all cases, the output of \fBld\fR is left in the file \fBa.out\fR by
  48 default. See NOTES.
  49 .sp
  50 .LP
  51 In dynamic mode, \fB-dy\fR, the default, relocatable object files that are
  52 provided as arguments are combined to produce an executable object file. This
  53 file is linked at execution with any shared object files that are provided as
  54 arguments. If the \fB-G\fR option is specified, relocatable object files are
  55 combined to produce a shared object. Without the \fB-G\fR option, a dynamic
  56 executable is created.
  57 .sp
  58 .LP
  59 In static mode, \fB-dn\fR, relocatable object files that are provided as
  60 arguments are combined to produce a static executable file. If the \fB-r\fR
  61 option is specified, relocatable object files are combined to produce one
```



```
  62 relocatable object file. See \fBStatic Executables\fR.
  63 .sp
  64 .LP
  65 Dynamic linking is the most common model for combining relocatable objects, and
  66 the eventual creation of processes within Solaris. This environment tightly
  67 couples the work of the link-editor and the runtime linker, \fBld.so.1\fR(1).
  68 Both of these utilities, together with their related technologies and
  69 utilities, are extensively documented in the \fILinker and Libraries Guide\fR.
  70 .sp
  71 .LP
  72 If any argument is a library, \fBld\fR by default searches the library exactly
  73 once at the point the library is encountered on the argument list. The library
  74 can be either a shared object or relocatable archive. See \fBar.h\fR(3HEAD)).
  75 .sp
  76 .LP
  77 A shared object consists of an indivisible, whole unit that has been generated
  78 by a previous link-edit of one or more input files. When the link-editor
  79 processes a shared object, the entire contents of the shared object become a
  80 logical part of the resulting output file image. The shared object is not
  81 physically copied during the link-edit as its actual inclusion is deferred
  82 until process execution. This logical inclusion means that all symbol entries
  83 defined in the shared object are made available to the link-editing process.
  84 See Chapter 4, \fIShared Objects,\fR in \fILinker and Libraries Guide\fR
  85 .sp
  86 .LP
  87 For an archive library, \fBld\fR loads only those routines that define an
  88 unresolved external reference. \fBld\fR searches the symbol table of the
  89 archive library sequentially to resolve external references that can be
  90 satisfied by library members. This search is repeated until no external
  91 references can be resolved by the archive. Thus, the order of members in the
  92 library is functionally unimportant, unless multiple library members exist that
  93 define the same external symbol. Archive libraries that have interdependencies
  94 can require multiple command line definitions, or the use of one of the
  95 \fB-z\fR \fBrescan\fR options. See \fIArchive Processing\fR in \fILinker and
  96 Libraries Guide\fR.
  97 .sp
  98 .LP
  99 \fBld\fR is a cross link-editor, able to link 32-bit objects or 64-bit objects,
 100 for Sparc or x86 targets. \fBld\fR uses the \fBELF\fR class and machine type of
 101 the first relocatable object on the command line to govern the mode in which to
 102 operate. The mixing of 32-bit objects and 64-bit objects is not permitted.
 103 Similarly, only objects of a single machine type are allowed. See the
 104 \fB-32\fR, \fB-64\fR and \fB-z target\fR options, and the \fBLD_NOEXEC_64\fR
 105 environment variable.
 106 .SS "Static Executables"
 107 .sp
 108 .LP
 109 The creation of static executables has been discouraged for many releases. In
 110 fact, 64-bit system archive libraries have never been provided. Because a
 111 static executable is built against system archive libraries, the executable
 112 contains system implementation details. This self-containment has a number of
 113 drawbacks.
 114 .RS +4
 115 .TP
 116 .ie t \(bu
 117 .el o
 118 The executable is immune to the benefits of system patches delivered as shared
 119 objects. The executable therefore, must be rebuilt to take advantage of many
 120 system improvements.
 121 .RE
 122 .RS +4
 123 .TP
 124 .ie t \(bu
 125 .el o
 126 The ability of the executable to run on future releases can be compromised.
 127 .RE
```

```
.RS +4
.TP
.ie t \(bu
.el o
The duplication of system implementation details negatively affects system
performance.
.RE
.sp
.LP
With Solaris 10, 32-bit system archive libraries are no longer provided.
Without these libraries, specifically \fBlibc.a\fR, the creation of static
executables is no longer achievable without specialized system knowledge.
However, the capability of \fBld\fR to process static linking options, and the
processing of archive libraries, remains unchanged.
.SH OPTIONS
.sp
.LP
The following options are supported.
.sp
.ne 2
.na
\fB\fB-32\fR | \fB-64\fR\fR
.ad
.sp .6
.RS 4n
Creates a 32-bit, or 64-bit object.
.sp
By default, the class of the object being generated is determined from the
first \fBELF\fR object processed from the command line. If no objects are
specified, the class is determined by the first object encountered within the
first archive processed from the command line. If there are no objects or
archives, the link-editor creates a 32-bit object.
.sp
The \fB-64\fR option is required to create a 64-bit object solely from a
mapfile.
.sp
This \fB-32\fR or \fB-64\fR options can also be used in the rare case of
linking entirely from an archive that contains a mixture of 32 and 64-bit
objects. If the first object in the archive is not the class of the object that
is required to be created, then the \fB-32\fR or \fB-64\fR option can be used
to direct the link-editor. See \fIThe 32-bit link-editor and 64-bit
link-editor\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
In static mode only, produces an executable object file. Undefined references
are not permitted. This option is the default behavior for static mode. The
\fB-a\fR option can not be used with the \fB-r\fR option. See \fBStatic
Executables\fR under DESCRIPTION.
.RE

.sp
.ne 2
.na
\fB\fB-b\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, provides no special processing for dynamic executable
relocations that reference symbols in shared objects. Without the \fB-b\fR
```



```
option, the link-editor applies techniques within a dynamic executable so that
the text segment can remain read-only. One technique is the creation of special
position-independent relocations for references to functions that are defined
in shared objects. Another technique arranges for data objects that are defined
in shared objects to be copied into the memory image of an executable at
runtime.
.sp
The \fB-b\fR option is intended for specialized dynamic objects and is not
recommended for general use. Its use suppresses all specialized processing
required to ensure an object's shareability, and can even prevent the
relocation of 64-bit executables.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBdirect\fR | \fBnodirect\fR\fR
.ad
.sp .6
.RS 4n
These options govern direct binding. \fB-B\fR \fBdirect\fR establishes direct
binding information by recording the relationship between each symbol reference
together with the dependency that provides the definition. In addition, direct
binding information is established between each symbol reference and an
associated definition within the object being created. The runtime linker uses
this information to search directly for a symbol in the associated object
rather than to carry out a default symbol search.
.sp
Direct binding information can only be established to dependencies specified
with the link-edit. Thus, you should use the \fB-z\fR \fBdefs\fR option.
Objects that wish to interpose on symbols in a direct binding environment
should identify themselves as interposers with the \fB-z\fR \fBinterpose\fR
option. The use of \fB-B\fR \fBdirect\fR enables \fB-z\fR \fBlazyload\fR for
all dependencies.
.sp
The \fB-B\fR \fBnodirect\fR option prevents any direct binding to the
interfaces offered by the object being created. The object being created can
continue to directly bind to external interfaces by specifying the \fB-z\fR
\fBdirect\fR option. See Appendix D, \fIDirect Bindings,\fR in \fILinker and
Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBdynamic\fR | \fBstatic\fR\fR
.ad
.sp .6
.RS 4n
Options governing library inclusion. \fB-B\fR \fBdynamic\fR is valid in dynamic
mode only. These options can be specified any number of times on the command
line as toggles: if the \fB-B\fR \fBstatic\fR option is given, no shared
objects are accepted until \fB-B\fR \fBdynamic\fR is seen. See the \fB-l\fR
option.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBeliminate\fR\fR
.ad
.sp .6
.RS 4n
Causes any global symbols, not assigned to a version definition, to be
eliminated from the symbol table. Version definitions can be supplied by means
of a \fBmapfile\fR to indicate the global symbols that should remain visible in
```

the generated object. This option achieves the same symbol elimination as the \fIauto-elimination\fR directive that is available as part of a \fBmapfile\fR version definition. This option can be useful when combining versioned and non-versioned relocatable objects. See also the \fB-B\fR \fBlocal\fR option and the \fB-B\fR \fBreduce\fR option. See \fIDefining Additional Symbols with a mapfile\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBgroup\fR\fR
.ad
.sp .6
.RS 4n
Establishes a shared object and its dependencies as a group. Objects within the group are bound to other members of the group at runtime. This mode is similar to adding the object to the process by using \fBdlopen\fR(3C) with the \fBRTLD_GROUP\fR mode. An object that has an explicit dependency on a object identified as a group, becomes a member of the group.
.sp
As the group must be self contained, use of the \fB-B\fR \fBgroup\fR option also asserts the \fB-z\fR \fBdefs\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBlocal\fR\fR
.ad
.sp .6
.RS 4n
Causes any global symbols, not assigned to a version definition, to be reduced to local. Version definitions can be supplied by means of a \fBmapfile\fR to indicate the global symbols that should remain visible in the generated object. This option achieves the same symbol reduction as the \fIauto-reduction\fR directive that is available as part of a \fBmapfile\fR version definition. This option can be useful when combining versioned and non-versioned relocatable objects. See also the \fB-B\fR \fBeliminate\fR option and the \fB-B\fR \fBreduce\fR option. See \fIDefining Additional Symbols with a mapfile\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBreduce\fR\fR
.ad
.sp .6
.RS 4n
When generating a relocatable object, causes the reduction of symbolic information defined by any version definitions. Version definitions can be supplied by means of a \fBmapfile\fR to indicate the global symbols that should remain visible in the generated object. By default, when a relocatable object is generated, version definitions are only recorded in the output image. The actual reduction of symbolic information is carried out when the object is used in the construction of a dynamic executable or shared object. The \fB-B\fR \fBreduce\fR option is applied automatically when a dynamic executable or shared object is created.
.RE

.sp
.ne 2
.na
\fB\fB-B\fR \fBsymbolic\fR\fR
.ad

.sp .6
.RS 4n
In dynamic mode only. When building a shared object, binds references to global symbols to their definitions, if available, within the object. Normally, references to global symbols within shared objects are not bound until runtime, even if definitions are available. This model allows definitions of the same symbol in an executable or other shared object to override the object's own definition. \fBld\fR issues warnings for undefined symbols unless \fB-z\fR \fBdefs\fR overrides.
.sp
The \fB-B\fR \fBsymbolic\fR option is intended for specialized dynamic objects and is not recommended for general use. To reduce the runtime relocation processing that is required an object, the creation of a version definition is recommended.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
Records the configuration file \fIname\fR for use at runtime. Configuration files can be employed to alter default search paths, provide a directory cache, together with providing alternative object dependencies. See \fBcrle\fR(1).
.RE

.sp
.ne 2
.na
\fB\fB-C\fR\fR
.ad
.sp .6
.RS 4n
Demangles C++ symbol names displayed in diagnostic messages.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fBy\fR | \fBn\fR\fR
.ad
.sp .6
.RS 4n
When \fB-d\fR \fBy\fR, the default, is specified, \fBld\fR uses dynamic linking. When \fB-d\fR \fBn\fR is specified, \fBld\fR uses static linking. See \fBStatic Executables\fR under DESCRIPTION, and \fB-B\fR \fBdynamic\fR|\fBstatic\fR.
.RE

.sp
.ne 2
.na
\fB\fB-D\fR \fItoken\fR,...\fR
.ad
.sp .6
.RS 4n
Prints debugging information as specified by each \fItoken\fR, to the standard error. The special token \fBhelp\fR indicates the full list of tokens available. See \fIDebugging Aids\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na

```
\fB\fB-e\fR \fIepsym\fR\fR
.ad
.br
.na
\fB\fB--entry\fR \fIepsym\fR\fR
.ad
.sp .6
.RS 4n
Sets the entry point address for the output file to be the symbol \fIepsym\fR.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIname\fR\fR
.ad
.br
.na
\fB\fB--auxiliary\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
Useful only when building a shared object. Specifies that the symbol table of
the shared object is used as an auxiliary filter on the symbol table of the
shared object specified by \fIname\fR. Multiple instances of this option are
allowed. This option can not be combined with the \fB-F\fR option. See
\fIGenerating Auxiliary Filters\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-F\fR \fIname\fR\fR
.ad
.br
.na
\fB\fB--filter\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
Useful only when building a shared object. Specifies that the symbol table of
the shared object is used as a filter on the symbol table of the shared object
specified by \fIname\fR. Multiple instances of this option are allowed. This
option can not be combined with the \fB-f\fR option. See \fIGenerating Standard
Filters\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-G\fR\fR
.ad
.br
.na
\fB\fB-shared\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, produces a shared object. Undefined symbols are allowed.
See Chapter 4, \fIShared Objects,\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-h\fR \fIname\fR\fR
```



```
.ad
.br
.na
\fB\fB--soname\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, when building a shared object, records \fIname\fR in the
object's dynamic section. \fIname\fR is recorded in any dynamic objects that
are linked with this object rather than the object's file system name.
Accordingly, \fIname\fR is used by the runtime linker as the name of the shared
object to search for at runtime. See \fIRecording a Shared Object Name\fR in
\fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR\fR
.ad
.sp .6
.RS 4n
Ignores \fBLD_LIBRARY_PATH\fR. This option is useful when an
\fBLD_LIBRARY_PATH\fR setting is in effect to influence the runtime library
search, which would interfere with the link-editing being performed.
.RE

.sp
.ne 2
.na
\fB\fB-I\fR \fIname\fR\fR
.ad
.br
.na
\fB\fB--dynamic-linker\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
When building an executable, uses \fIname\fR as the path name of the
interpreter to be written into the program header. The default in static mode
is no interpreter. In dynamic mode, the default is the name of the runtime
linker, \fBld.so.1\fR(1). Either case can be overridden by \fB-I\fR \fIname\fR.
\fBexec\fR(2) loads this interpreter when the \fBa.out\fR is loaded, and passes
control to the interpreter rather than to the \fBa.out\fR directly.
.RE

.sp
.ne 2
.na
\fB\fB-l\fR \fIx\fR\fR
.ad
.br
.na
\fB\fB--library\fR \fIx\fR\fR
.ad
.sp .6
.RS 4n
Searches a library \fBlib\fR\fIx\fR\fB\&.so\fR or \fBlib\fR\fIx\fR\fB\&.a\fR,
the conventional names for shared object and archive libraries, respectively.
In dynamic mode, unless the \fB-B\fR \fBstatic\fR option is in effect, \fBld\fR
searches each directory specified in the library search path for a
\fBlib\fR\fIx\fR\fB\&.so\fR or \fBlib\fR\fIx\fR\fB\&.a\fR file. The directory
search stops at the first directory containing either. \fBld\fR chooses the
file ending in \fB\&.so\fR if \fB-l\fR\fIx\fR expands to two files with names
of the form \fBlib\fR\fIx\fR\fB\&.so\fR and \fBlib\fR\fIx\fR\fB\&.a\fR. If no
\fBlib\fR\fIx\fR\fB\&.so\fR is found, then \fBld\fR accepts
```

\fBlib\fR\fIx\fR\fB\&.a\fR. In static mode, or when the \fB-B\fR \fBstatic\fR option is in effect, \fBld\fR selects only the file ending in \fB\&.a\fR. \fBld\fR searches a library when the library is encountered, so the placement of \fB-l\fR is significant. See \fILinking With Additional Libraries\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-L\fR \fIpath\fR\fR
.ad
.br
.na
\fB\fB--library-path\fR \fIpath\fR\fR
.ad
.sp .6
.RS 4n
Adds \fIpath\fR to the library search directories. \fBld\fR searches for libraries first in any directories specified by the \fB-L\fR options and then in the standard directories. This option is useful only if the option precedes the \fB-l\fR options to which the \fB-L\fR option applies. See \fIDirectories Searched by the Link-Editor\fR in \fILinker and Libraries Guide\fR.
.sp
The environment variable \fBLD_LIBRARY_PATH\fR can be used to supplement the library search path, however the \fB-L\fR option is recommended, as the environment variable is also interpreted by the runtime environment. See \fBLD_LIBRARY_PATH\fR under ENVIRONMENT VARIABLES.
.RE

.sp
.ne 2
.na
\fB\fB-m\fR\fR
.ad
.sp .6
.RS 4n
Produces a memory map or listing of the input/output sections, together with any non-fatal multiply-defined symbols, on the standard output.
.RE

.sp
.ne 2
.na
\fB\fB-M\fR \fImapfile\fR\fR
.ad
.sp .6
.RS 4n
Reads \fImapfile\fR as a text file of directives to \fBld\fR. This option can be specified multiple times. If \fImapfile\fR is a directory, then all regular files, as defined by \fBstat\fR(2), within the directory are processed. See Chapter 9, \fIMapfile Option,\fR in \fILinker and Libraries Guide\fR. Example mapfiles are provided in \fB/usr/lib/ld\fR. See FILES.
.RE

.sp
.ne 2
.na
\fB\fB-N\fR \fIstring\fR\fR
.ad
.sp .6
.RS 4n
This option causes a \fBDT_NEEDED\fR entry to be added to the \fB\&.dynamic\fR section of the object being built. The value of the \fBDT_NEEDED\fR string is the \fIstring\fR that is specified on the command line. This option is position dependent, and the \fBDT_NEEDED\fR \fB\&.dynamic\fR entry is relative to the

other dynamic dependencies discovered on the link-edit line. This option is useful for specifying dependencies within device driver relocatable objects when combined with the \fB-dy\fR and \fB-r\fR options.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIoutfile\fR\fR
.ad
.br
.na
\fB\fB--output\fR \fIoutfile\fR\fR
.ad
.sp .6
.RS 4n
Produces an output object file that is named \fIoutfile\fR. The name of the default object file is \fBa.out\fR.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR \fIauditlib\fR\fR
.ad
.sp .6
.RS 4n
Identifies an audit library, \fIauditlib\fR. This audit library is used to audit the object being created at runtime. A shared object identified as requiring auditing with the \fB-p\fR option, has this requirement inherited by any object that specifies the shared object as a dependency. See the \fB-P\fR option. See \fIRuntime Linker Auditing Interface\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-P\fR \fIauditlib\fR\fR
.ad
.sp .6
.RS 4n
Identifies an audit library, \fIauditlib\fR. This audit library is used to audit the dependencies of the object being created at runtime. Dependency auditing can also be inherited from dependencies that are identified as requiring auditing. See the \fB-p\fR option, and the \fB-z\fR \fBglobalaudit\fR option. See \fIRuntime Linker Auditing Interface\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-Q\fR \fBy\fR | \fBn\fR\fR
.ad
.sp .6
.RS 4n
Under \fB-Q\fR \fBy\fR, an \fBident\fR string is added to the \fB\&.comment\fR section of the output file. This string identifies the version of the \fBld\fR used to create the file. This results in multiple \fBld\fR \fBidents\fR when there have been multiple linking steps, such as when using \fBld\fR \fB-r\fR. This identification is identical with the default action of the \fBcc\fR command. \fB-Q\fR \fBn\fR suppresses version identification. \fB\&.comment\fR sections can be manipulated by the \fBmcs\fR(1) utility.
.RE

```
.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.br
.na
\fB\fB--relocatable\fR\fR
.ad
.sp .6
.RS 4n
Combines relocatable object files to produce one relocatable object file.
\fBld\fR does not complain about unresolved references. This option cannot be
used with the \fB-a\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR \fIpath\fR\fR
.ad
.br
.na
\fB\fB-rpath\fR \fIpath\fR\fR
.ad
.sp .6
.RS 4n
A colon-separated list of directories used to specify library search
directories to the runtime linker. If present and not NULL, the path is
recorded in the output object file and passed to the runtime linker. Multiple
instances of this option are concatenated together with each \fIpath\fR
separated by a colon. See \fIDirectories Searched by the Runtime Linker\fR in
\fILinker and Libraries Guide\fR.
.sp
The use of a runpath within an associated object is preferable to setting
global search paths such as through the \fBLD_LIBRARY_PATH\fR environment
variable. Only the runpaths that are necessary to find the objects dependencies
should be recorded. \fBldd\fR(1) can also be used to discover unused runpaths
in dynamic objects, when used with the \fB-U\fR option.
.sp
Various tokens can also be supplied with a runpath that provide a flexible
means of identifying system capabilities or an objects location. See Appendix
C, \fIEstablishing Dependencies with Dynamic String Tokens,\fR in \fILinker and
Libraries Guide\fR. The \fB$ORIGIN\fR token is especially useful in allowing
dynamic objects to be relocated to different locations in the file system.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR\fR
.ad
.br
.na
\fB\fB--strip-all\fR\fR
.ad
.sp .6
.RS 4n
Strips symbolic information from the output file. Any debugging information,
that is, \fB\&.line\fR, \fB\&.debug*\fR, and \fB\&.stab*\fR sections, and their
associated relocation entries are removed. Except for relocatable files, a
symbol table \fBSHT_SYMTAB\fR and its associated string table section are not
created in the output object file. The elimination of a \fBSHT_SYMTAB\fR symbol
table can reduce the \fB\&.stab*\fR debugging information that is generated
using the compiler drivers \fB-g\fR option. See the \fB-z\fR \fBredlocsym\fR
and \fB-z\fR \fBnoldynsym\fR options.
```



```
.RE

.sp
.ne 2
.na
\fB\fB-S\fR \fIsupportlib\fR\fR
.ad
.sp .6
.RS 4n
The shared object \fIsupportlib\fR is loaded with \fBld\fR and given
information regarding the linking process. Shared objects that are defined by
using the \fB-S\fR option can also be supplied using the \fBSGS_SUPPORT\fR
environment variable. See \fILink-Editor Support Interface\fR in \fILinker and
Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR\fR
.ad
.sp .6
.RS 4n
Turns off the warning for multiply-defined symbols that have different sizes or
different alignments.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIsymname\fR\fR
.ad
.br
.na
\fB\fB--undefined\fR \fIsymname\fR\fR
.ad
.sp .6
.RS 4n
Enters \fIsymname\fR as an undefined symbol in the symbol table. This option is
useful for loading entirely from an archive library. In this instance, an
unresolved reference is needed to force the loading of the first routine. The
placement of this option on the command line is significant. This option must
be placed before the library that defines the symbol. See \fIDefining
Additional Symbols with the u option\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-V\fR\fR
.ad
.br
.na
\fB\fB--version\fR\fR
.ad
.sp .6
.RS 4n
Outputs a message giving information about the version of \fBld\fR being used.
.RE

.sp
.ne 2
.na
\fB\fB-Y\fR \fBP,\fR\fIdirlist\fR\fR
.ad
.sp .6
```

```
.RS 4n
Changes the default directories used for finding libraries. \fIdirlist\fR is a
colon-separated path list.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBabsexec\fR\fR
.ad
.sp .6
.RS 4n
Useful only when building a dynamic executable. Specifies that references to
external absolute symbols should be resolved immediately instead of being left
for resolution at runtime. In very specialized circumstances, this option
removes text relocations that can result in excessive swap space demands by an
executable.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBallextract\fR | \fBdefaultextract\fR | \fBweakextract\fR\fR
.ad
.br
.na
\fB\fB--whole-archive\fR | \fB--no-whole-archive\fR\fR
.ad
.sp .6
.RS 4n
Alters the extraction criteria of objects from any archives that follow. By
default, archive members are extracted to satisfy undefined references and to
promote tentative definitions with data definitions. Weak symbol references do
not trigger extraction. Under the \fB-z\fR \fBallextract\fR or
\fB--whole-archive\fR options, all archive members are extracted from the
archive. Under \fB-z\fR \fBweakextract\fR, weak references trigger archive
extraction. The \fB-z\fR \fBdefaultextract\fR or \fB--no-whole-archive\fR
options provide a means of returning to the default following use of the former
extract options. See \fIArchive Processing\fR in \fILinker and Libraries
Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBaltexec64\fR\fR
.ad
.sp .6
.RS 4n
Execute the 64-bit \fBld\fR. The creation of very large 32-bit objects can
exhaust the virtual memory that is available to the 32-bit \fBld\fR. The
\fB-z\fR \fBaltexec64\fR option can be used to force the use of the associated
64-bit \fBld\fR. The 64-bit \fBld\fR provides a larger virtual address space
for building 32-bit objects. See \fIThe 32-bit link-editor and 64-bit
link-editor\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBcombreloc\fR | \fBnocombreloc\fR\fR
.ad
.sp .6
.RS 4n
By default, \fBld\fR combines multiple relocation sections when building
executables or shared objects. This section combination differs from
```



```
relocatable objects, in which relocation sections are maintained in a
one-to-one relationship with the sections to which the relocations must be
applied. The \fB-z\fR \fBnocombreloc\fR option disables this merging of
relocation sections, and preserves the one-to-one relationship found in the
original relocatable objects.
.sp
\fBld\fR sorts the entries of data relocation sections by their symbol
reference. This sorting reduces runtime symbol lookup. When multiple relocation
sections are combined, this sorting produces the least possible relocation
overhead when objects are loaded into memory, and speeds the runtime loading of
dynamic objects.
.sp
Historically, the individual relocation sections were carried over to any
executable or shared object, and the \fB-z\fR \fBcombreloc\fR option was
required to enable the relocation section merging previously described.
Relocation section merging is now the default. The \fB-z\fR \fBcombreloc\fR
option is still accepted for the benefit of old build environments, but the
option is unnecessary, and has no effect.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBassert-deflib\fR\fR
.ad
.br
.na
\fB\fB-z\fR \fBassert-deflib=\fR\fIlibname\fR\fR
.ad
.sp .6
.RS 4n
Enables warnings that check the location of where libraries passed in with
\fB-l\fR are found. If the link-editor finds a library on its default search
path it will emit a warning. This warning can be made fatal in conjunction with
the option \fB-z fatal-warnings\fR. Passing \fIlibname\fR white lists a library
from this check. The library must be the full name of the library, e.g.
\fIlibc.so\fR. To white list multiple libraries, the \fB-z
assert-deflib=\fR\fIlibname\fR option can be repeated multiple times. This
option is useful when trying to build self-contained objects where a referenced
library might exist in the default system library path and in alternate paths
specified by \fB-L\fR, but you only want the alternate paths to be used.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBdefs\fR | \fBnodefs\fR\fR
.ad
.br
.na
\fB\fB--no-undefined\fR\fR
.ad
.sp .6
.RS 4n
The \fB-z\fR \fBdefs\fR option and the \fB--no-undefined\fR option force a
fatal error if any undefined symbols remain at the end of the link. This mode
is the default when an executable is built. For historic reasons, this mode is
\fBnot\fR the default when building a shared object. Use of the \fB-z\fR
\fBdefs\fR option is recommended, as this mode assures the object being built
is self-contained. A self-contained object has all symbolic references resolved
internally, or to the object's immediate dependencies.
.sp
The \fB-z\fR \fBnodefs\fR option allows undefined symbols. For historic
reasons, this mode is the default when a shared object is built. When used with
executables, the behavior of references to such undefined symbols is
unspecified. Use of the \fB-z\fR \fBnodefs\fR option is not recommended.
```

```
 920 .RE

 922 .sp
 923 .ne 2
 924 .na
 925 \fB\fB-z\fR \fBdirect\fR | \fBnodirect\fR\fR
 926 .ad
 927 .sp .6
 928 .RS 4n
 929 Enables or disables direct binding to any dependencies that follow on the
 930 command line. These options allow finer control over direct binding than the
 931 global counterpart \fB-B\fR \fBdirect\fR. The \fB-z\fR \fBdirect\fR option also
 932 differs from the \fB-B\fR \fBdirect\fR option in the following areas. Direct
 933 binding information is not established between a symbol reference and an
 934 associated definition within the object being created. Lazy loading is not
 935 enabled.
 936 .RE

 938 .sp
 939 .ne 2
 940 .na
 941 \fB\fB-z\fR \fBendfiltee\fR\fR
 942 .ad
 943 .sp .6
 944 .RS 4n
 945 Marks a filtee so that when processed by a filter, the filtee terminates any
 946 further filtee searches by the filter. See \fIReducing Filtee Searches\fR in
 947 \fILinker and Libraries Guide\fR.
 948 .RE

 950 .sp
 951 .ne 2
 952 .na
 953 \fB\fB-z\fR \fBfatal-warnings\fR | \fBnofatal-warnings\fR\fR
 954 .ad
 955 .br
 956 .na
 957 \fB\fB--fatal-warnings\fR | \fB--no-fatal-warnings\fR
 958 .ad
 959 .sp .6
 960 .RS 4n
 961 Controls the behavior of warnings emitted from the link-editor. Setting \fB-z
 962 fatal-warnings\fR promotes warnings emitted by the link-editor to fatal errors
 963 that will cause the link-editor to fail before linking. \fB-z
 964 nofatal-warnings\fR instead demotes these warnings such that they will not cause
 965 the link-editor to exit prematurely.
 966 .RE


 969 .sp
 970 .ne 2
 971 .na
 972 \fB\fB-z\fR \fBfiniarray=\fR\fIfunction\fR\fR
 973 .ad
 974 .sp .6
 975 .RS 4n
 976 Appends an entry to the \fB\&.finiarray\fR section of the object being built.
 977 If no \fB\&.finiarray\fR section is present, a section is created. The new
 978 entry is initialized to point to \fIfunction\fR. See \fIInitialization and
 979 Termination Sections\fR in \fILinker and Libraries Guide\fR.
 980 .RE

 982 .sp
 983 .ne 2
 984 .na
 985 \fB\fB-z\fR \fBglobalaudit\fR\fR
```



```
 986 .ad
 987 .sp .6
 988 .RS 4n
 989 This option supplements an audit library definition that has been recorded with
 990 the \fB-P\fR option. This option is only meaningful when building a dynamic
 991 executable. Audit libraries that are defined within an object with the \fB-P\fR
 992 option typically allow for the auditing of the immediate dependencies of the
 993 object. The \fB-z\fR \fBglobalaudit\fR promotes the auditor to a global
 994 auditor, thus allowing the auditing of all dependencies. See \fIInvoking the
 995 Auditing Interface\fR in \fILinker and Libraries Guide\fR.
 996 .sp
 997 An auditor established with the \fB-P\fR option and the \fB-z\fR
 998 \fBglobalaudit\fR option, is equivalent to the auditor being established with
 999 the \fBLD_AUDIT\fR environment variable. See \fBld.so.1\fR(1).
1000 .RE

1002 .sp
1003 .ne 2
1004 .na
1005 \fB\fB-z\fR \fBgroupperm\fR | \fBnogroupperm\fR\fR
1006 .ad
1007 .sp .6
1008 .RS 4n
1009 Assigns, or deassigns each dependency that follows to a unique group. The
1010 assignment of a dependency to a group has the same effect as if the dependency
1011 had been built using the \fB-B\fR \fBgroup\fR option.
1012 .RE

1014 .sp
1015 .ne 2
1016 .na
1017 \fB-z\fR \fBguidance\fR[=\fIid1\fR,\fIid2\fR...]
1018 .ad
1019 .sp .6
1020 .RS 4n
1021 Give messages suggesting link-editor features that could improve the resulting
1022 dynamic object.
1023 .LP
1024 Specific classes of suggestion can be silenced by specifying an optional comma s
1025 list of guidance identifiers.
1026 .LP
1027 The current classes of suggestion provided are:

1029 .sp
1030 .ne 2
1031 .na
1032 Enable use of direct binding
1033 .ad
1034 .sp .6
1035 .RS 4n
1036 Suggests that \fB-z direct\fR or \fB-B direct\fR be present prior to any
1037 specified dependency.  This allows predictable symbol binding at runtime.

1039 Can be disabled with \fB-z guidance=nodirect\fR
1040 .RE

1042 .sp
1043 .ne 2
1044 .na
1045 Enable lazy dependency loading
1046 .ad
1047 .sp .6
1048 .RS 4n
1049 Suggests that \fB-z lazyload\fR be present prior to any specified dependency.
1050 This allows the dynamic object to be loaded more quickly.
```

```
Can be disabled with \fB-z guidance=nolazyload\fR.
.RE

.sp
.ne 2
.na
Shared objects should define all their dependencies.
.ad
.sp .6
.RS 4n
Suggests that \fB-z defs\fR be specified on the link-editor command line.
Shared objects that explicitly state all their dependencies behave more
predictably when used.

Can be be disabled with \fB-z guidance=nodefs\fR
.RE

.sp
.ne 2
.na
Version 2 mapfile syntax
.ad
.sp .6
.RS 4n
Suggests that any specified mapfiles use the more readable version 2 syntax.

Can be disabled with \fB-z guidance=nomapfile\fR.
.RE

.sp
.ne 2
.na
Read-only text segment
.ad
.sp .6
.RS 4n
Should any runtime relocations within the text segment exist, suggests that
the object be compiled with position independent code (PIC).  Keeping large
allocatable sections read-only allows them to be shared between processes
using a given shared object.

Can be disabled with \fB-z guidance=notext\fR
.RE

.sp
.ne 2
.na
No unused dependencies
.ad
.sp .6
.RS 4n
Suggests that any dependency not referenced by the resulting dynamic object be
removed from the link-editor command line.

Can be disabled with \fB-z guidance=nounused\fR.
.RE
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBhelp\fR\fR
.ad
.br
.na
\fB\fB--help\fR\fR
```



```
.ad
.sp .6
.RS 4n
Print a summary of the command line options on the standard output and exit.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBignore\fR | \fBrecord\fR\fR
.ad
.br
.na
\fB\fB--as-needed\fR | \fB--no-as-needed\fR\fR
.ad
.sp .6
.RS 4n
Ignores, or records, dynamic dependencies that are not referenced as part of
the link-edit. Ignores, or records, unreferenced \fBELF\fR sections from the
relocatable objects that are read as part of the link-edit. By default,
\fB-z\fR \fBrecord\fR is in effect.
.sp
If an \fBELF\fR section is ignored, the section is eliminated from the output
file being generated. A section is ignored when three conditions are true. The
eliminated section must contribute to an allocatable segment. The eliminated
section must provide no global symbols. No other section from any object that
contributes to the link-edit, must reference an eliminated section.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBinitarray=\fR\fIfunction\fR\fR
.ad
.sp .6
.RS 4n
Appends an entry to the \fB\&.initarray\fR section of the object being built.
If no \fB\&.initarray\fR section is present, a section is created. The new
entry is initialized to point to \fIfunction\fR. See \fIInitialization and
Termination Sections\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBinitfirst\fR\fR
.ad
.sp .6
.RS 4n
Marks the object so that its runtime initialization occurs before the runtime
initialization of any other objects brought into the process at the same time.
In addition, the object runtime finalization occurs after the runtime
finalization of any other objects removed from the process at the same time.
This option is only meaningful when building a shared object.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBinterpose\fR\fR
.ad
.sp .6
.RS 4n
Marks the object as an interposer. At runtime, an object is identified as an
explicit interposer if the object has been tagged using the \fB-z interpose\fR
option. An explicit interposer is also established when an object is loaded
```

using the \fBLD_PRELOAD\fR environment variable. Implicit interposition can occur because of the load order of objects, however, this implicit interposition is unknown to the runtime linker. Explicit interposition can ensure that interposition takes place regardless of the order in which objects are loaded. Explicit interposition also ensures that the runtime linker searches for symbols in any explicit interposers when direct bindings are in effect.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBlazyload\fR | \fBnolazyload\fR\fR
.ad
.sp .6
.RS 4n
Enables or disables the marking of dynamic dependencies to be lazily loaded. Dynamic dependencies which are marked \fBlazyload\fR are not loaded at initial process start-up. These dependencies are delayed until the first binding to the object is made. \fBNote:\fR Lazy loading requires the correct declaration of dependencies, together with associated runpaths for each dynamic object used within a process. See \fILazy Loading of Dynamic Dependencies\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBld32\fR=\fIarg1\fR,\fIarg2\fR,...\fR
.ad
.br
.na
\fB\fB-z\fR \fBld64\fR=\fIarg1\fR,\fIarg2\fR,...\fR
.ad
.sp .6
.RS 4n
The class of the link-editor is affected by the class of the output file being created and by the capabilities of the underlying operating system. The \fB-z\fR \fBld\fR[\fB32\fR|\fB64\fR] options provide a means of defining any link-editor argument. The defined argument is only interpreted, respectively, by the 32-bit class or 64-bit class of the link-editor.
.sp
For example, support libraries are class specific, so the correct class of support library can be ensured using:
.sp
.in +2
.nf
\fBld ... -z ld32=-Saudit32.so.1 -z ld64=-Saudit64.so.1 ...\fR
.fi
.in -2
.sp

The class of link-editor that is invoked is determined from the \fBELF\fR class of the first relocatable file that is seen on the command line. This determination is carried out \fBprior\fR to any \fB-z\fR \fBld\fR[\fB32\fR|\fB64\fR] processing.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBloadfltr\fR\fR
.ad
.sp .6
.RS 4n
Marks a filter to indicate that filtees must be processed immediately at

runtime. Normally, filter processing is delayed until a symbol reference is bound to the filter. The runtime processing of an object that contains this flag mimics that which occurs if the \fBLD_LOADFLTR\fR environment variable is in effect. See the \fBld.so.1\fR(1).
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBmuldefs\fR\fR
.ad
.br
.na
\fB\fB--allow-multiple-definition\fR\fR
.ad
.sp .6
.RS 4n
Allows multiple symbol definitions. By default, multiple symbol definitions that occur between relocatable objects result in a fatal error condition. This option, suppresses the error condition, allowing the first symbol definition to be taken.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnocompstrtab\fR\fR
.ad
.sp .6
.RS 4n
Disables the compression of \fBELF\fR string tables. By default, string compression is applied to \fBSHT_STRTAB\fR sections, and to \fBSHT_PROGBITS\fR sections that have their \fBSHF_MERGE\fR and \fBSHF_STRINGS\fR section flags set.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnodefaultlib\fR\fR
.ad
.sp .6
.RS 4n
Marks the object so that the runtime default library search path, used after any \fBLD_LIBRARY_PATH\fR or runpaths, is ignored. This option implies that all dependencies of the object can be satisfied from its runpath.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnodelete\fR\fR
.ad
.sp .6
.RS 4n
Marks the object as non-deletable at runtime. This mode is similar to adding the object to the process by using \fBdlopen\fR(3C) with the \fBRTLD_NODELETE\fR mode.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnodlopen\fR\fR
.ad
.sp .6

```
.RS 4n
Marks the object as not available to \fBdlopen\fR(3C), either as the object
specified by the \fBdlopen()\fR, or as any form of dependency required by the
object specified by the \fBdlopen()\fR. This option is only meaningful when
building a shared object.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnodump\fR\fR
.ad
.sp .6
.RS 4n
Marks the object as not available to \fBdldump\fR(3C).
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnoldynsym\fR\fR
.ad
.sp .6
.RS 4n
Prevents the inclusion of a \fB\&.SUNW_ldynsym\fR section in dynamic
executables or sharable libraries. The \fB\&.SUNW_ldynsym\fR section augments
the \fB\&.dynsym\fR section by providing symbols for local functions. Local
function symbols allow debuggers to display local function names in stack
traces from stripped programs. Similarly, \fBdladdr\fR(3C) is able to supply
more accurate results.
.sp
The \fB-z\fR \fBnoldynsym\fR option also prevents the inclusion of the two
symbol sort sections that are related to the \fB\&.SUNW_ldynsym\fR section. The
\fB\&.SUNW_dynsymsort\fR section provides sorted access to regular function and
variable symbols. The \fB\&.SUNW_dyntlssort\fR section provides sorted access
to thread local storage (\fBTLS\fR) variable symbols.
.sp
The \fB\&.SUNW_ldynsym\fR, \fB\&.SUNW_dynsymsort\fR, and
\fB\&.SUNW_dyntlssort\fR sections, which becomes part of the allocable text
segment of the resulting file, cannot be removed by \fBstrip\fR(1). Therefore,
the \fB-z\fR \fBnoldynsym\fR option is the only way to prevent their inclusion.
See the \fB-s\fR and \fB-z\fR \fBredlocsym\fR options.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnopartial\fR\fR
.ad
.sp .6
.RS 4n
Partially initialized symbols, that are defined within relocatable object
files, are expanded in the output file being generated.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBnoversion\fR\fR
.ad
.sp .6
.RS 4n
Does not record any versioning sections. Any version sections or associated
\fB\&.dynamic\fR section entries are not generated in the output image.
.RE
```



```
.sp
.ne 2
.na
\fB\fB-z\fR \fBnow\fR\fR
.ad
.sp .6
.RS 4n
Marks the object as requiring non-lazy runtime binding. This mode is similar to
adding the object to the process by using \fBdlopen\fR(3C) with the
\fBRTLD_NOW\fR mode. This mode is also similar to having the \fBLD_BIND_NOW\fR
environment variable in effect. See \fBld.so.1\fR(1).
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBorigin\fR\fR
.ad
.sp .6
.RS 4n
Marks the object as requiring immediate \fB$ORIGIN\fR processing at runtime.
This option is only maintained for historic compatibility, as the runtime
analysis of objects to provide for \fB$ORIGIN\fR processing is now default.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBpreinitarray=\fR\fIfunction\fR\fR
.ad
.sp .6
.RS 4n
Appends an entry to the \fB\&.preinitarray\fR section of the object being
built. If no \fB\&.preinitarray\fR section is present, a section is created.
The new entry is initialized to point to \fIfunction\fR. See \fIInitialization
and Termination Sections\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBredlocsym\fR\fR
.ad
.sp .6
.RS 4n
Eliminates all local symbols except for the \fISECT\fR symbols from the symbol
table \fBSHT_SYMTAB\fR. All relocations that refer to local symbols are updated
to refer to the corresponding \fISECT\fR symbol. This option allows specialized
objects to greatly reduce their symbol table sizes. Eliminated local symbols
can reduce the \fB\&.stab*\fR debugging information that is generated using the
compiler drivers \fB-g\fR option. See the \fB-s\fR and \fB-z\fR \fBnoldynsym\fR
options.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBrelaxreloc\fR\fR
.ad
.sp .6
.RS 4n
\fBld\fR normally issues a fatal error upon encountering a relocation using a
symbol that references an eliminated COMDAT section. If \fB-z\fR
\fBrelaxreloc\fR is enabled, \fBld\fR instead redirects such relocations to the
equivalent symbol in the COMDAT section that was kept. \fB-z\fR
\fBrelaxreloc\fR is a specialized option, mainly of interest to compiler
```

authors, and is not intended for general use.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBrescan-now\fR\fR
.ad
.br
.na
\fB\fB-z\fR \fBrescan\fR\fR
.ad
.sp .6
.RS 4n
These options rescan the archive files that are provided to the link-edit. By default, archives are processed once as the archives appear on the command line. Archives are traditionally specified at the end of the command line so that their symbol definitions resolve any preceding references. However, specifying archives multiple times to satisfy their own interdependencies can be necessary.
.sp
\fB-z\fR \fBrescan-now\fR is a positional option, and is processed by the link-editor immediately when encountered on the command line. All archives seen on the command line up to that point are immediately reprocessed in an attempt to locate additional archive members that resolve symbol references. This archive rescanning is repeated until a pass over the archives occurs in which no new members are extracted.
.sp
\fB-z\fR \fBrescan\fR is a position independent option. The link-editor defers the rescan operation until after it has processed the entire command line, and then initiates a final rescan operation over all archives seen on the command line. The \fB-z\fR \fBrescan\fR operation can interact incorrectly with objects that contain initialization (.init) or finalization (.fini) sections, preventing the code in those sections from running. For this reason, \fB-z\fR \fBrescan\fR is deprecated, and use of \fB-z\fR \fBrescan-now\fR is advised.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBrescan-start\fR ... \fB-z\fR \fBrescan-end\fR\fR
.ad
.br
.na
\fB\fB--start-group\fR ... \fB--end-group\fR\fR
.ad
.br
.na
\fB\fB-(\fR ... \fB-)\fR\fR
.ad
.sp .6
.RS 4n
Defines an archive rescan group. This is a positional construct, and is processed by the link-editor immediately upon encountering the closing delimiter option. Archives found within the group delimiter options are reprocessed as a group in an attempt to locate additional archive members that resolve symbol references. This archive rescanning is repeated until a pass over the archives On the occurs in which no new members are extracted. Archive rescan groups cannot be nested.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBtarget=sparc|x86\fR \fI\fR\fR

.ad
.sp .6
.RS 4n
Specifies the machine type for the output object. Supported targets are Sparc and x86. The 32-bit machine type for the specified target is used unless the \fB-64\fR option is also present, in which case the corresponding 64-bit machine type is used. By default, the machine type of the object being generated is determined from the first \fBELF\fR object processed from the command line. If no objects are specified, the machine type is determined by the first object encountered within the first archive processed from the command line. If there are no objects or archives, the link-editor assumes the native machine. This option is useful when creating an object directly with \fBld\fR whose input is solely from a \fBmapfile\fR. See the \fB-M\fR option. It can also be useful in the rare case of linking entirely from an archive that contains objects of different machine types for which the first object is not of the desired machine type. See \fIThe 32-bit link-editor and 64-bit link-editor\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBtext\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, forces a fatal error if any relocations against non-writable, allocatable sections remain. For historic reasons, this mode is not the default when building an executable or shared object. However, its use is recommended to ensure that the text segment of the dynamic object being built is shareable between multiple running processes. A shared text segment incurs the least relocation overhead when loaded into memory. See \fIPosition-Independent Code\fR in \fILinker and Libraries Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBtextoff\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, allows relocations against all allocatable sections, including non-writable ones. This mode is the default when building a shared object.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBtextwarn\fR\fR
.ad
.sp .6
.RS 4n
In dynamic mode only, lists a warning if any relocations against non-writable, allocatable sections remain. This mode is the default when building an executable.
.RE

.sp
.ne 2
.na
\fB\fB-z\fR \fBverbose\fR\fR
.ad
.sp .6
.RS 4n

```
1580 This option provides additional warning diagnostics during a link-edit.
1581 Presently, this option conveys suspicious use of displacement relocations. This
1582 option also conveys the restricted use of static \fBTLS\fR relocations when
1583 building shared objects. In future, this option might be enhanced to provide
1584 additional diagnostics that are deemed too noisy to be generated by default.
1585 .RE

1587 .sp
1588 .ne 2
1589 .na
1590 \fB\fB-z\fR\fBwrap=\fR\fIsymbol\fR\fR
1591 .ad
1592 .br
1593 .na
1594 \fB\fB-wrap=\fR \fIsymbol\fR\fR
1595 .ad
1596 .br
1597 .na
1598 \fB\fB--wrap=\fR \fIsymbol\fR\fR
1599 .ad
1600 .sp .6
1601 .RS 4n
1602 Rename undefined references to \fIsymbol\fR in order to allow wrapper code to
1603 be linked into the output object without having to modify source code. When
1604 \fB-z wrap\fR is specified, all undefined references to \fIsymbol\fR are
1605 modified to reference \fB__wrap_\fR\fIsymbol\fR, and all references to
1606 \fB__real_\fR\fIsymbol\fR are modified to reference \fIsymbol\fR. The user is
1607 expected to provide an object containing the \fB__wrap_\fR\fIsymbol\fR
1608 function. This wrapper function can call \fB__real_\fR\fIsymbol\fR in order to
1609 reference the actual function being wrapped.
1610 .sp
1611 The following is an example of a wrapper for the \fBmalloc\fR(3C) function:
1612 .sp
1613 .in +2
1614 .nf
1615 void *
1616 __wrap_malloc(size_t c)
1617 {
1618         (void) printf("malloc called with %zu\en", c);
1619         return (__real_malloc(c));
1620 }
```

_____***unchanged_portion_omitted_***