
32607 Fri Sep 27 10:14:07 2013
new/usr/src/cmd/getconf/getconf.c
4146 getconf(1) does not support XATTR_ENABLED, XATTR_EXISTS, SATTR_ENABLED, SAT

```
unchanged_portion_omitted
89 /*
90 * Some entries in this structure have both leading _ and non leading _
91 * versions for compatibility with various versions of the specifications.
92 * It is not expected that both forms will be required for new entries.
93 */

95 /* POSIX.2-1992 table 2-17 */
96 _POSIX2_BC_BASE_MAX, "POSIX2_BC_BASE_MAX", SELFCNF, NOFLAGS,
97 _POSIX2_BC_SCALE_MAX, "POSIX2_BC_SCALE_MAX", SELFCNF, NOFLAGS,
98 _POSIX2_BC_DIM_MAX, "POSIX2_BC_DIM_MAX", SELFCNF, NOFLAGS,
99 _POSIX2_BC_DIM_MAX, "POSIX2_BC_DIM_MAX", SELFCNF, NOFLAGS,
100 _POSIX2_BC_SCALE_MAX, "POSIX2_BC_SCALE_MAX", SELFCNF, NOFLAGS,
101 _POSIX2_BC_SCALE_MAX, "POSIX2_BC_SCALE_MAX", SELFCNF, NOFLAGS,
102 _POSIX2_BC_STRING_MAX, "POSIX2_BC_STRING_MAX", SELFCNF, NOFLAGS,
103 _POSIX2_BC_STRING_MAX, "POSIX2_BC_STRING_MAX", SELFCNF, NOFLAGS,
104 _POSIX2_COLL_WEIGHTS_MAX, "POSIX2_COLL_WEIGHTS_MAX", SELFCNF, NOFLAGS,
105 _POSIX2_COLL_WEIGHTS_MAX, "POSIX2_COLL_WEIGHTS_MAX", SELFCNF, NOFLAGS,
106 _POSIX2_EXPR_NEST_MAX, "POSIX2_EXPR_NEST_MAX", SELFCNF, NOFLAGS,
107 _POSIX2_EXPR_NEST_MAX, "POSIX2_EXPR_NEST_MAX", SELFCNF, NOFLAGS,
108 _POSIX2_LINE_MAX, "POSIX2_LINE_MAX", SELFCNF, NOFLAGS,
109 _POSIX2_LINE_MAX, "POSIX2_LINE_MAX", SELFCNF, NOFLAGS,
110 _POSIX2_RE_DUP_MAX, "POSIX2_RE_DUP_MAX", SELFCNF, NOFLAGS,
111 _POSIX2_RE_DUP_MAX, "POSIX2_RE_DUP_MAX", SELFCNF, NOFLAGS,
112 /* See limits.h section below for _POSIX2_RE_DUP_MAX */

114 /* UNIX 98 */
115 _POSIX_AIO_LISTIO_MAX, "_POSIX_AIO_LISTIO_MAX", SELFCNF, NOFLAGS,
116 _POSIX_AIO_MAX, "_POSIX_AIO_MAX", SELFCNF, NOFLAGS,
117 _POSIX_DELAYTIMER_MAX, "POSIX_DELAYTIMER_MAX", SELFCNF, NOFLAGS,
118 _POSIX_LOGIN_NAME_MAX, "_POSIX_LOGIN_NAME_MAX", SELFCNF, NOFLAGS,
119 _POSIX_MQ_OPEN_MAX, "POSIX_MQ_OPEN_MAX", SELFCNF, NOFLAGS,
120 _POSIX_MQ_PRIO_MAX, "POSIX_MQ_PRIO_MAX", SELFCNF, NOFLAGS,
121 _POSIX_RTSIG_MAX, "POSIX_RTSIG_MAX", SELFCNF, NOFLAGS,
122 _POSIX_SEM_NSEMS_MAX, "POSIX_SEM_NSEMS_MAX", SELFCNF, NOFLAGS,
123 _POSIX_SEM_VALUE_MAX, "POSIX_SEM_VALUE_MAX", SELFCNF, NOFLAGS,
124 _POSIX_SIGQUEUE_MAX, "POSIX_SIGQUEUE_MAX", SELFCNF, NOFLAGS,
125 _POSIX_TTY_NAME_MAX, "_POSIX_TTY_NAME_MAX", SELFCNF, NOFLAGS,

127 /* UNIX 03 */
128 _POSIX_HOST_NAME_MAX, "_POSIX_HOST_NAME_MAX", SELFCNF, NOFLAGS,
129 _POSIX_RE_DUP_MAX, "POSIX_RE_DUP_MAX", SELFCNF, NOFLAGS,
130 _POSIX_CLOCKRES_MIN, "POSIX_CLOCKRES_MIN", SELFCNF, NOFLAGS,
131 _POSIX_SYMLINK_MAX, "POSIX_SYMLINK_MAX", SELFCNF, NOFLAGS,
132 _POSIX_SYMLINK_MAX, "POSIX_SYMLINK_MAX", SELFCNF, NOFLAGS,
133 CHARCLASS_NAME_MAX, "_POSIX2_CHARCLASS_NAME_MAX", SELFCNF, NOFLAGS,

135 /* POSIX.2-1992 table 2-17 */
136 _SC_2_VERSION, "POSIX2_VERSION", SYSCNF, NOFLAGS,
137 _SC_2_VERSION, "_POSIX2_VERSION", SYSCNF, NOFLAGS,

139 /* POSIX.2-1992 table 2-18 */
140 _SC_BC_BASE_MAX, "BC_BASE_MAX", SYSCNF, NOFLAGS,
141 _SC_BC_DIM_MAX, "BC_DIM_MAX", SYSCNF, NOFLAGS,
142 _SC_BC_SCALE_MAX, "BC_SCALE_MAX", SYSCNF, NOFLAGS,
143 _SC_BC_STRING_MAX, "BC_STRING_MAX", SYSCNF, NOFLAGS,
144 _SC_COLL_WEIGHTS_MAX, "COLL_WEIGHTS_MAX", SYSCNF, NOFLAGS,
145 _SC_EXPR_NEST_MAX, "EXPR_NEST_MAX", SYSCNF, NOFLAGS,
146 _SC_LINE_MAX, "LINE_MAX", SYSCNF, NOFLAGS,
147 _SC_RE_DUP_MAX, "RE_DUP_MAX", SYSCNF, NOFLAGS,
```

```
149 /* POSIX.2-1992 table 2-19 */
150 _SC_2_C_BIND, "POSIX2_C_BIND", SYSCNF, NOFLAGS,
151 _SC_2_C_BIND, "_POSIX2_C_BIND", SYSCNF, NOFLAGS,
152 _SC_2_C_DEV, "POSIX2_C_DEV", SYSCNF, NOFLAGS,
153 _SC_2_C_DEV, "_POSIX2_C_DEV", SYSCNF, NOFLAGS,
154 _SC_2_CHAR_TERM, "POSIX2_CHAR_TERM", SYSCNF, NOFLAGS,
155 _SC_2_CHAR_TERM, "_POSIX2_CHAR_TERM", SYSCNF, NOFLAGS,
156 _SC_2_FORT_DEV, "POSIX2_FORT_DEV", SYSCNF, NOFLAGS,
157 _SC_2_FORT_DEV, "_POSIX2_FORT_DEV", SYSCNF, NOFLAGS,
158 _SC_2_FORT_RUN, "POSIX2_FORT_RUN", SYSCNF, NOFLAGS,
159 _SC_2_FORT_RUN, "_POSIX2_FORT_RUN", SYSCNF, NOFLAGS,
160 _SC_2_LOCALEDEF, "POSIX2_LOCALEDEF", SYSCNF, NOFLAGS,
161 _SC_2_LOCALEDEF, "_POSIX2_LOCALEDEF", SYSCNF, NOFLAGS,
162 _SC_2_SW_DEV, "POSIX2_SW_DEV", SYSCNF, NOFLAGS,
163 _SC_2_SW_DEV, "_POSIX2_SW_DEV", SYSCNF, NOFLAGS,
164 _SC_2_UPE, "POSIX2_UPE", SYSCNF, NOFLAGS,
165 _SC_2_UPE, "_POSIX2_UPE", SYSCNF, NOFLAGS,

167 /* POSIX.1-1990 table 2-3 */
168 _POSIX_ARG_MAX, "POSIX_ARG_MAX", SELFCNF, NOFLAGS,
169 _POSIX_CHILD_MAX, "POSIX_CHILD_MAX", SELFCNF, NOFLAGS,
170 _POSIX_LINK_MAX, "POSIX_LINK_MAX", SELFCNF, NOFLAGS,
171 _POSIX_MAX_CANON, "POSIX_MAX_CANON", SELFCNF, NOFLAGS,
172 _POSIX_MAX_INPUT, "POSIX_MAX_INPUT", SELFCNF, NOFLAGS,
173 _POSIX_NAME_MAX, "POSIX_NAME_MAX", SELFCNF, NOFLAGS,
174 _POSIX_NGROUPS_MAX, "POSIX_NGROUPS_MAX", SELFCNF, NOFLAGS,
175 _POSIX_OPEN_MAX, "POSIX_OPEN_MAX", SELFCNF, NOFLAGS,
176 _POSIX_PATH_MAX, "POSIX_PATH_MAX", SELFCNF, NOFLAGS,
177 _POSIX_PIPE_BUF, "POSIX_PIPE_BUF", SELFCNF, NOFLAGS,
178 _POSIX_SSIZE_MAX, "POSIX_SSIZE_MAX", SELFCNF, NOFLAGS,
179 _POSIX_STREAM_MAX, "POSIX_STREAM_MAX", SELFCNF, NOFLAGS,
180 _POSIX_TIMER_MAX, "POSIX_TIMER_MAX", SELFCNF, NOFLAGS,
181 _POSIX_THREAD_DESTRUCTOR_ITERATIONS, "POSIX_THREAD_DESTRUCTOR_ITERATIONS", SELFCNF, NOFLAGS,
182 "POSIX_THREAD_DESTRUCTOR_ITERATIONS", SELFCNF, NOFLAGS,
183 _POSIX_THREAD_KEYS_MAX, "POSIX_THREAD_KEYS_MAX", SELFCNF, NOFLAGS,
184 "POSIX_THREAD_KEYS_MAX", SELFCNF, NOFLAGS,
185 _POSIX_THREAD_THREADS_MAX, "POSIX_THREAD_THREADS_MAX", SELFCNF, NOFLAGS,
186 "POSIX_THREAD_THREADS_MAX", SELFCNF, NOFLAGS,
187 _POSIX_TZNAME_MAX, "_POSIX_TZNAME_MAX", SELFCNF, NOFLAGS,

189 /* POSIX.1-1990 table 4-2 */
190 _SC_ARG_MAX, "ARG_MAX", SYSCNF, NOFLAGS,
191 _SC_CHILD_MAX, "CHILD_MAX", SYSCNF, NOFLAGS,
192 _SC_CLK_TCK, "CLK_TCK", SYSCNF, NOFLAGS,
193 _SC_JOB_CONTROL, "POSIX_JOB_CONTROL", SYSCNF, NOFLAGS,
194 _SC_NGROUPS_MAX, "NGROUPS_MAX", SYSCNF, NOFLAGS,
195 _SC_OPEN_MAX, "OPEN_MAX", SYSCNF, NOFLAGS,
196 _SC_SAVED_IDS, "POSIX_SAVED_IDS", SYSCNF, NOFLAGS,
197 _SC_STREAM_MAX, "STREAM_MAX", SYSCNF, NOFLAGS,
198 _SC_TZNAME_MAX, "TZNAME_MAX", SYSCNF, NOFLAGS,
199 _SC_VERSION, "_POSIX_VERSION", SYSCNF, NOFLAGS,

201 /* pathconf() variables */
203 /* POSIX.1-1990 table 5-2 */

205 _PC_LINK_MAX, "LINK_MAX", PATHCONF, NOFLAGS,
206 _PC_MAX_CANON, "MAX_CANON", PATHCONF, NOFLAGS,
207 _PC_MAX_INPUT, "MAX_INPUT", PATHCONF, NOFLAGS,
208 _PC_NAME_MAX, "NAME_MAX", PATHCONF, NOFLAGS,
209 _PC_PATH_MAX, "PATH_MAX", PATHCONF, NOFLAGS,
210 _PC_PIPE_BUF, "PIPE_BUF", PATHCONF, NOFLAGS,
211 _PC_CHOWN_RESTRICTED, "POSIX_CHOWN_RESTRICTED", PATHCONF, NOFLAGS,
212 _PC_NO_TRUNC, "POSIX_NO_TRUNC", PATHCONF, NOFLAGS,
213 _PC_VDISABLE, "_POSIX_VDISABLE", PATHCONF, NOFLAGS,
```

```

215      /* UNIX 98 additions */
217      _PC_ASYNC_IO,          "_POSIX_ASYNC_IO",      PATHCONF, NOFLAGS,
218      _PC_PRIO_IO,          "_POSIX_PRIO_IO",      PATHCONF, NOFLAGS,
219      _PC_SYNC_IO,          "_POSIX_SYNC_IO",      PATHCONF, NOFLAGS,
221      /* Large File Summit name (UNIX 98) */
223      _PC_FILESIZEBITS,     "FILESIZEBITS",        PATHCONF, NOFLAGS,
224      _PC_XATTR_ENABLED,    "XATTR_ENABLED",      PATHCONF, NOFLAGS,
225      _PC_XATTR_EXISTS,     "XATTR_EXISTS",       PATHCONF, NOFLAGS,
226 #endif /* ! codereview */
228      /* UNIX 03 Additions */
230      /* Advisory Information (ADV) */
231      _PC_ALLOC_SIZE_MIN,   "POSIX_ALLOC_SIZE_MIN", PATHCONF, NOFLAGS,
232      _PC_REC_INCR_XFER_SIZE, "POSIX_REC_INCR_XFER_SIZE", PATHCONF, NOFLAGS,
233      _PC_REC_MAX_XFER_SIZE, "POSIX_REC_MAX_XFER_SIZE", PATHCONF, NOFLAGS,
234      _PC_REC_MIN_XFER_SIZE, "POSIX_REC_MIN_XFER_SIZE", PATHCONF, NOFLAGS,
235      _PC_REC_XFER_ALIGN,   "POSIX_REC_XFER_ALIGN", PATHCONF, NOFLAGS,
237      _PC_2_SYMLINKS,       "POSIX2_SYMLINKS",     PATHCONF, NOFLAGS,
238      _PC_SYMLINK_MAX,      "SYMLINK_MAX",         PATHCONF, NOFLAGS,
240      _PC_SATTR_ENABLED,    "SATTR_ENABLED",       PATHCONF, NOFLAGS,
241      _PC_SATTR_EXISTS,     "SATTR_EXISTS",       PATHCONF, NOFLAGS,
243 #endif /* ! codereview */
244      /* UNIX 08 additions */
246      _PC_TIMESTAMP_RESOLUTION, "_POSIX_TIMESTAMP_RESOLUTION",
247      PATHCONF, NOFLAGS,
249      /* confstr() variables */
251      /* POSIX.2-1992 description of getconf utility */
252      _CS_PATH,             "PATH",                CONFSTR, NOFLAGS,
254      /* POSIX.2-1992 table B-18 */
255      _CS_PATH,             "CS_PATH",             CONFSTR, NOFLAGS,
256      _CS_PATH,             "CS_PATH",             CONFSTR, NOFLAGS,
258      /* command names for large file configuration information */
259      /* large file compilation environment configuration */
261      _CS_LFS_CFLAGS,       "LFS_CFLAGS",         CONFSTR, NOFLAGS,
262      _CS_LFS_LDFLAGS,     "LFS_LDFLAGS",       CONFSTR, NOFLAGS,
263      _CS_LFS_LIBS,        "LFS_LIBS",          CONFSTR, NOFLAGS,
264      _CS_LFS_LINTFLAGS,   "LFS_LINTFLAGS",     CONFSTR, NOFLAGS,
266      /* transitional large file interface configuration */
268      _CS_LFS64_CFLAGS,    "LFS64_CFLAGS",      CONFSTR, NOFLAGS,
269      _CS_LFS64_LDFLAGS,   "LFS64_LDFLAGS",     CONFSTR, NOFLAGS,
270      _CS_LFS64_LIBS,      "LFS64_LIBS",        CONFSTR, NOFLAGS,
271      _CS_LFS64_LINTFLAGS, "LFS64_LINTFLAGS",   CONFSTR, NOFLAGS,
273      /*
274      * UNIX 98:
275      * confstr additions, these support
276      * programming specifications/environments.
277      *
278      * Unfortunately there is some uncertainty with
279      * the standard at this point about the constant
280      * name, thus we need to recognize BOTH with and

```

```

281      * without the initial underscore.
282      */
283      _CS_XBS5_ILP32_OFF32_CFLAGS, "XBS5_ILP32_OFF32_CFLAGS",
284      CONFSTR, NOFLAGS,
285      _CS_XBS5_ILP32_OFF32_CFLAGS, "_XBS5_ILP32_OFF32_CFLAGS",
286      CONFSTR, NOFLAGS,
288      _CS_XBS5_ILP32_OFF32_LDFLAGS, "XBS5_ILP32_OFF32_LDFLAGS",
289      CONFSTR, NOFLAGS,
290      _CS_XBS5_ILP32_OFF32_LDFLAGS, "_XBS5_ILP32_OFF32_LDFLAGS",
291      CONFSTR, NOFLAGS,
293      _CS_XBS5_ILP32_OFF32_LIBS,    "XBS5_ILP32_OFF32_LIBS",
294      CONFSTR, NOFLAGS,
295      _CS_XBS5_ILP32_OFF32_LIBS,    "_XBS5_ILP32_OFF32_LIBS",
296      CONFSTR, NOFLAGS,
298      _CS_XBS5_ILP32_OFF32_LINTFLAGS, "XBS5_ILP32_OFF32_LINTFLAGS",
299      CONFSTR, NOFLAGS,
300      _CS_XBS5_ILP32_OFF32_LINTFLAGS, "_XBS5_ILP32_OFF32_LINTFLAGS",
301      CONFSTR, NOFLAGS,
303      _CS_XBS5_ILP32_OFFBIG_CFLAGS, "XBS5_ILP32_OFFBIG_CFLAGS",
304      CONFSTR, NOFLAGS,
305      _CS_XBS5_ILP32_OFFBIG_CFLAGS, "_XBS5_ILP32_OFFBIG_CFLAGS",
306      CONFSTR, NOFLAGS,
308      _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "XBS5_ILP32_OFFBIG_LDFLAGS",
309      CONFSTR, NOFLAGS,
310      _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "_XBS5_ILP32_OFFBIG_LDFLAGS",
311      CONFSTR, NOFLAGS,
313      _CS_XBS5_ILP32_OFFBIG_LIBS,    "XBS5_ILP32_OFFBIG_LIBS",
314      CONFSTR, NOFLAGS,
315      _CS_XBS5_ILP32_OFFBIG_LIBS,    "_XBS5_ILP32_OFFBIG_LIBS",
316      CONFSTR, NOFLAGS,
318      _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "XBS5_ILP32_OFFBIG_LINTFLAGS",
319      CONFSTR, NOFLAGS,
320      _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "_XBS5_ILP32_OFFBIG_LINTFLAGS",
321      CONFSTR, NOFLAGS,
323      _CS_XBS5_LP64_OFF64_CFLAGS,    "XBS5_LP64_OFF64_CFLAGS",
324      CONFSTR, NOFLAGS,
325      _CS_XBS5_LP64_OFF64_CFLAGS,    "_XBS5_LP64_OFF64_CFLAGS",
326      CONFSTR, NOFLAGS,
328      _CS_XBS5_LP64_OFF64_LDFLAGS,    "XBS5_LP64_OFF64_LDFLAGS",
329      CONFSTR, NOFLAGS,
330      _CS_XBS5_LP64_OFF64_LDFLAGS,    "_XBS5_LP64_OFF64_LDFLAGS",
331      CONFSTR, NOFLAGS,
333      _CS_XBS5_LP64_OFF64_LIBS,       "XBS5_LP64_OFF64_LIBS",
334      CONFSTR, NOFLAGS,
335      _CS_XBS5_LP64_OFF64_LIBS,       "_XBS5_LP64_OFF64_LIBS",
336      CONFSTR, NOFLAGS,
338      _CS_XBS5_LP64_OFF64_LINTFLAGS,  "XBS5_LP64_OFF64_LINTFLAGS",
339      CONFSTR, NOFLAGS,
340      _CS_XBS5_LP64_OFF64_LINTFLAGS,  "_XBS5_LP64_OFF64_LINTFLAGS",
341      CONFSTR, NOFLAGS,
343      _CS_XBS5_LPBIG_OFFBIG_CFLAGS,   "XBS5_LPBIG_OFFBIG_CFLAGS",
344      CONFSTR, NOFLAGS,
345      _CS_XBS5_LPBIG_OFFBIG_CFLAGS,   "_XBS5_LPBIG_OFFBIG_CFLAGS",
346      CONFSTR, NOFLAGS,

```

```

348  _CS_XBS5_LPBIG_OFFBIG_LDFLAGS, "XBS5_LPBIG_OFFBIG_LDFLAGS",
349      CONFSTR,          NOFLAGS,
350  _CS_XBS5_LPBIG_OFFBIG_LDFLAGS, "_XBS5_LPBIG_OFFBIG_LDFLAGS",
351      CONFSTR,          NOFLAGS,

353  _CS_XBS5_LPBIG_OFFBIG_LIBS,    "XBS5_LPBIG_OFFBIG_LIBS",
354      CONFSTR,          NOFLAGS,
355  _CS_XBS5_LPBIG_OFFBIG_LIBS,    "_XBS5_LPBIG_OFFBIG_LIBS",
356      CONFSTR,          NOFLAGS,

358  _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "XBS5_LPBIG_OFFBIG_LINTFLAGS",
359      CONFSTR,          NOFLAGS,
360  _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "_XBS5_LPBIG_OFFBIG_LINTFLAGS",
361      CONFSTR,          NOFLAGS,

363  /*
364  * UNIX 03 confstr() additions for compilation environments
365  */
366  _CS_POSIX_V6_ILP32_OFF32_CFLAGS, "POSIX_V6_ILP32_OFF32_CFLAGS",
367      CONFSTR,          NOFLAGS,
368  _CS_POSIX_V6_ILP32_OFF32_LDFLAGS, "POSIX_V6_ILP32_OFF32_LDFLAGS",
369      CONFSTR,          NOFLAGS,
370  _CS_POSIX_V6_ILP32_OFF32_LIBS,    "POSIX_V6_ILP32_OFF32_LIBS",
371      CONFSTR,          NOFLAGS,
372  _CS_POSIX_V6_ILP32_OFFBIG_CFLAGS, "POSIX_V6_ILP32_OFFBIG_CFLAGS",
373      CONFSTR,          NOFLAGS,
374  _CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS, "POSIX_V6_ILP32_OFFBIG_LDFLAGS",
375      CONFSTR,          NOFLAGS,
376  _CS_POSIX_V6_ILP32_OFFBIG_LIBS,    "POSIX_V6_ILP32_OFFBIG_LIBS",
377      CONFSTR,          NOFLAGS,
378  _CS_POSIX_V6_LP64_OFF64_CFLAGS, "POSIX_V6_LP64_OFF64_CFLAGS",
379      CONFSTR,          NOFLAGS,
380  _CS_POSIX_V6_LP64_OFF64_LDFLAGS, "POSIX_V6_LP64_OFF64_LDFLAGS",
381      CONFSTR,          NOFLAGS,
382  _CS_POSIX_V6_LP64_OFF64_LIBS,    "POSIX_V6_LP64_OFF64_LIBS",
383      CONFSTR,          NOFLAGS,
384  _CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS, "POSIX_V6_LPBIG_OFFBIG_CFLAGS",
385      CONFSTR,          NOFLAGS,
386  _CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS, "POSIX_V6_LPBIG_OFFBIG_LDFLAGS",
387      CONFSTR,          NOFLAGS,
388  _CS_POSIX_V6_LPBIG_OFFBIG_LIBS,    "POSIX_V6_LPBIG_OFFBIG_LIBS",
389      CONFSTR,          NOFLAGS,
390  _CS_POSIX_V6_WIDTH_RESTRICTED_ENVS, "POSIX_V6_WIDTH_RESTRICTED_ENVS",
391      CONFSTR,          NOFLAGS,
392  /* End of UNIX 03 compilation environments */

394 #ifdef  _SC_2_C_VERSION      /* POSIX.2 table B-4 */
395  _SC_2_C_VERSION,          "_POSIX2_C_VERSION",    SYSCONF,    NOFLAGS,
396 #else
397  _POSIX2_C_VERSION,        "_POSIX2_C_VERSION",    SELFCONF,  NOFLAGS,
398 #endif

400 /*
401 * XPG4 support BEGINS
402 */

404 #ifndef CHARCLASS_NAME_MAX
405 #define CHARCLASS_NAME_MAX -1
406 #endif
407 #ifndef CHAR_BIT
408 #define CHAR_BIT -1
409 #endif
410 #ifndef CHAR_MAX
411 #define CHAR_MAX -1
412 #endif

```

```

413 #ifndef CHAR_MIN
414 #define CHAR_MIN -1
415 #endif
416 #ifndef INT_MAX
417 #define INT_MAX -1
418 #endif
419 #ifndef INT_MIN
420 #define INT_MIN -1
421 #endif
422 #ifndef LONG_BIT
423 #define LONG_BIT -1
424 #endif
425 #ifndef LONG_MAX
426 #define LONG_MAX -1
427 #endif
428 #ifndef LONG_MIN
429 #define LONG_MIN -1
430 #endif
431 #ifndef MB_LEN_MAX
432 #define MB_LEN_MAX -1
433 #endif
434 #ifndef NL_NMAX
435 #define NL_NMAX -1
436 #endif
437 #ifndef NL_ARGMAX
438 #define NL_ARGMAX -1
439 #endif
440 #ifndef NL_LANGMAX
441 #define NL_LANGMAX -1
442 #endif
443 #ifndef NL_MSGMAX
444 #define NL_MSGMAX -1
445 #endif
446 #ifndef NL_SETMAX
447 #define NL_SETMAX -1
448 #endif
449 #ifndef NL_TEXTMAX
450 #define NL_TEXTMAX -1
451 #endif
452 #ifndef NZERO
453 #define NZERO -1
454 #endif
455 #ifndef SCHAR_MAX
456 #define SCHAR_MAX -1
457 #endif
458 #ifndef SCHAR_MIN
459 #define SCHAR_MIN -1
460 #endif
461 #ifndef SHRT_MAX
462 #define SHRT_MAX -1
463 #endif
464 #ifndef SHRT_MIN
465 #define SHRT_MIN -1
466 #endif
467 #ifndef TMP_MAX
468 #define TMP_MAX -1
469 #endif
470 #ifndef WORD_BIT
471 #define WORD_BIT -1
472 #endif

475 #ifndef _XOPEN_XPG2
476 #define _XOPEN_XPG2 -1
477 #endif
478 #ifndef _XOPEN_XPG3

```

```

479 #define _XOPEN_XPG3      -1
480 #endif
481 #ifndef _XOPEN_XPG4
482 #define _XOPEN_XPG4      -1
483 #endif

485 /*
486  * the following are values that we should find in <limits.h>
487  * so we use SELFCONF here.
488  *
489  */

491 /* POSIX.1-1990 */
492 CHARCLASS_NAME_MAX,    "CHARCLASS_NAME_MAX",    SELFCONF,    NOFLAGS,
493 SSIZE_MAX,            "SSIZE_MAX",            SELFCONF,    NOFLAGS,

495 /* POSIX.1-1988 */
496 CHAR_BIT,             "CHAR_BIT",             SELFCONF,    NOFLAGS,
497 CHAR_MAX,             "CHAR_MAX",             SELFCONF,    NOFLAGS,
498 CHAR_MIN,             "CHAR_MIN",             SELFCONF,    NOFLAGS,
499 INT_MAX,              "INT_MAX",              SELFCONF,    NOFLAGS,
500 INT_MIN,              "INT_MIN",              SELFCONF,    NOFLAGS,
501 LONG_BIT,            "LONG_BIT",            SELFCONF,    NOFLAGS,
502 LONG_MAX,            "LONG_MAX",            SELFCONF,    NOFLAGS,
503 LONG_MIN,            "LONG_MIN",            SELFCONF,    NOFLAGS,
504 MB_LEN_MAX,          "MB_LEN_MAX",          SELFCONF,    NOFLAGS,
505 NL_NMAX,             "NL_NMAX",             SELFCONF,    NOFLAGS,
506 NL_ARGMAX,           "NL_ARGMAX",           SELFCONF,    NOFLAGS,
507 NL_LANGMAX,          "NL_LANGMAX",          SELFCONF,    NOFLAGS,
508 NL_MSGMAX,           "NL_MSGMAX",           SELFCONF,    NOFLAGS,
509 NL_SETMAX,           "NL_SETMAX",           SELFCONF,    NOFLAGS,
510 NL_TEXTMAX,          "NL_TEXTMAX",          SELFCONF,    NOFLAGS,
511 NZERO,               "NZERO",               SELFCONF,    NOFLAGS,
512 SCHAR_MAX,           "SCHAR_MAX",           SELFCONF,    NOFLAGS,
513 SCHAR_MIN,           "SCHAR_MIN",           SELFCONF,    NOFLAGS,
514 SHRT_MAX,            "SHRT_MAX",            SELFCONF,    NOFLAGS,
515 SHRT_MIN,            "SHRT_MIN",            SELFCONF,    NOFLAGS,
516 TMP_MAX,             "TMP_MAX",             SELFCONF,    NOFLAGS,

518 /*
519  * for the unsigned maximums, we cannot rely on the value -1
520  * to indicate "undefined".
521  */
522 #ifndef UCHAR_MAX
523 0,                    "UCHAR_MAX",          SELFCONF,    UNDEFINED,
524 #else
525 (long)UCHAR_MAX,      "UCHAR_MAX",          SELFCONF,    UNSIGNED_VALUE,
526 #endif /* UCHAR_MAX */
527 #ifndef UINT_MAX
528 0,                    "UINT_MAX",           SELFCONF,    UNDEFINED,
529 #else
530 (long)UINT_MAX,       "UINT_MAX",           SELFCONF,    UNSIGNED_VALUE,
531 #endif /* UINT_MAX */
532 #ifndef ULONG_MAX
533 0,                    "ULONG_MAX",          SELFCONF,    UNDEFINED,
534 #else
535 (long)ULONG_MAX,      "ULONG_MAX",          SELFCONF,    UNSIGNED_VALUE,
536 #endif /* ULONG_MAX */
537 #ifndef USHRT_MAX
538 0,                    "USHRT_MAX",          SELFCONF,    UNDEFINED,
539 #else
540 (long)USHRT_MAX,      "USHRT_MAX",          SELFCONF,    UNSIGNED_VALUE,
541 #endif /* USHRT_MAX */
542 #ifndef SIZE_MAX
543 0,                    "SIZE_MAX",           SELFCONF,    UNDEFINED,
544 #else

```

```

545 (long)SIZE_MAX,       "SIZE_MAX",           SELFCONF,    UNSIGNED_VALUE,
546 #endif /* SIZE_MAX */

548 WORD_BIT,            "WORD_BIT",           SELFCONF,    NOFLAGS,

550 /* XPG4 Version 2 */
551 _XOPEN_IOV_MAX,      "_XOPEN_IOV_MAX",     SELFCONF,    NOFLAGS,

553 /* UNIX 03 */
554 _XOPEN_NAME_MAX,     "_XOPEN_NAME_MAX",    SELFCONF,    NOFLAGS,
555 _XOPEN_PATH_MAX,     "_XOPEN_PATH_MAX",    SELFCONF,    NOFLAGS,

557 /*
558  * The following are defined via sysconf(). These are considered
559  * an extension to sysconf().
560  * The -0 is needed if _XOPEN_XPG? is defined to be an empty string.
561  */
562 _XOPEN_XPG2-0,        "_XOPEN_XPG2",        SELFCONF,    NOFLAGS,
563 _XOPEN_XPG3-0,        "_XOPEN_XPG3",        SELFCONF,    NOFLAGS,
564 _XOPEN_XPG4-0,        "_XOPEN_XPG4",        SELFCONF,    NOFLAGS,

566 /*
567  * The following should be provided by sysconf() (e.g use SYSCONF),
568  * so we look for the appropriate _SC_* symbol in <unistd.h>.
569  * If it is not defined, then we use SELFCONF with the value of -1.
570  */
571 _SC_XOPEN_VERSION,    "_XOPEN_VERSION",     SYSCONF,    NOFLAGS,
572 _SC_XOPEN_XCU_VERSION, "_XOPEN_XCU_VERSION", SYSCONF,    NOFLAGS,
573 _SC_XOPEN_CRYPT,      "_XOPEN_CRYPT",       SYSCONF,    NOFLAGS,
574 _SC_XOPEN_ENH_I18N,   "_XOPEN_ENH_I18N",    SYSCONF,    NOFLAGS,
575 _SC_XOPEN_SHM,        "_XOPEN_SHM",         SYSCONF,    NOFLAGS,

577 /*
578  * XPG4 support ends
579  */

581 /* XPG4, Version 2 support */

583 _SC_ATEXIT_MAX,       "ATEXIT_MAX",         SYSCONF,    NOFLAGS,
584 _SC_PAGESIZE,         "PAGESIZE",           SYSCONF,    NOFLAGS,
585 _SC_PAGE_SIZE,        "PAGE_SIZE",          SYSCONF,    NOFLAGS,
586 _SC_XOPEN_UNIX,       "_XOPEN_UNIX",        SYSCONF,    NOFLAGS,
587 _SC_IOV_MAX,          "IOV_MAX",            SYSCONF,    NOFLAGS,

589 /*
590  * UNIX 98 - XPG5 additions
591  */
592 _SC_XOPEN_LEGACY,     "_XOPEN_LEGACY",      SYSCONF,    NOFLAGS,
593 _SC_PASS_MAX,         "PASS_MAX",           SYSCONF,    NOFLAGS,
594 _SC_AIO_LISTIO_MAX,   "AIO_LISTIO_MAX",    SYSCONF,    NOFLAGS,
595 _SC_AIO_MAX,          "AIO_MAX",            SYSCONF,    NOFLAGS,
596 _SC_AIO_PRIO_DELTA_MAX, "AIO_PRIO_DELTA_MAX", SYSCONF,    NOFLAGS,
597 _SC_DELAYTIMER_MAX,   "DELAYTIMER_MAX",     SYSCONF,    NOFLAGS,
598 _SC_MQ_OPEN_MAX,      "MQ_OPEN_MAX",        SYSCONF,    NOFLAGS,
599 _SC_MQ_PRIO_MAX,      "MQ_PRIO_MAX",        SYSCONF,    NOFLAGS,
600 _SC_RTSMAX,           "RTSMAX",             SYSCONF,    NOFLAGS,
601 _SC_SEM_NSEMS_MAX,    "SEM_NSEMS_MAX",      SYSCONF,    NOFLAGS,
602 _SC_SEM_VALUE_MAX,    "SEM_VALUE_MAX",      SYSCONF,    NOFLAGS,
603 _SC_SIGQUEUE_MAX,     "SIGQUEUE_MAX",       SYSCONF,    NOFLAGS,
604 _SC_TIMER_MAX,        "TIMER_MAX",          SYSCONF,    NOFLAGS,

606 _SC_ASYNCHRONOUS_IO,  "_POSIX_ASYNCHRONOUS_IO", SYSCONF,    NOFLAGS,
607 _SC_FSYNC,            "_POSIX_FSYNC",       SYSCONF,    NOFLAGS,
608 _SC_MAPPED_FILES,     "_POSIX_MAPPED_FILES", SYSCONF,    NOFLAGS,
609 _SC_MEMLOCK,          "_POSIX_MEMLOCK",     SYSCONF,    NOFLAGS,
610 _SC_MEMLOCK_RANGE,    "_POSIX_MEMLOCK_RANGE", SYSCONF,    NOFLAGS,

```

```

611 _SC_MEMORY_PROTECTION, "POSIX_MEMORY_PROTECTION", SYSCONF, NOFLAGS,
612 _SC_MESSAGE_PASSING, "POSIX_MESSAGE_PASSING", SYSCONF, NOFLAGS,
613 _SC_PRIORITIZED_IO, "POSIX_PRIORITIZED_IO", SYSCONF, NOFLAGS,
614 _SC_PRIORITY_SCHEDULING, "POSIX_PRIORITY_SCHEDULING", SYSCONF, NOFLAGS,
615 _SC_REALTIME_SIGNALS, "POSIX_REALTIME_SIGNALS", SYSCONF, NOFLAGS,
616 _SC_SEMAPHORES, "POSIX_SEMAPHORES", SYSCONF, NOFLAGS,

618 _SC_SHARED_MEMORY_OBJECTS, "POSIX_SHARED_MEMORY_OBJECTS",
619 SYSCONF, NOFLAGS,

621 _SC_SYNCHRONIZED_IO, "POSIX_SYNCHRONIZED_IO", SYSCONF, NOFLAGS,
622 _SC_TIMERS, "POSIX_TIMERS", SYSCONF, NOFLAGS,

624 _SC_LOGIN_NAME_MAX, "LOGIN_NAME_MAX", SYSCONF, NOFLAGS,

626 _SC_THREAD_DESTRUCTOR_ITERATIONS, "PTHREAD_DESTRUCTOR_ITERATIONS",
627 SYSCONF, NOFLAGS,

629 _SC_THREAD_KEYS_MAX, "PTHREAD_KEYS_MAX", SYSCONF, NOFLAGS,
630 _SC_THREAD_STACK_MIN, "PTHREAD_STACK_MIN", SYSCONF, NOFLAGS,
631 _SC_THREAD_THREADS_MAX, "PTHREAD_THREADS_MAX", SYSCONF, NOFLAGS,

633 _SC_THREADS, "POSIX_THREADS", SYSCONF, NOFLAGS,

635 _SC_THREAD_ATTR_STACKADDR, "POSIX_THREAD_ATTR_STACKADDR",
636 SYSCONF, NOFLAGS,
637 _SC_THREAD_ATTR_STACKADDR, "POSIX_THREAD_ATTR_STACKADDR",
638 SYSCONF, NOFLAGS,

640 _SC_THREAD_ATTR_STACKSIZE, "POSIX_THREAD_ATTR_STACKSIZE",
641 SYSCONF, NOFLAGS,
642 _SC_THREAD_ATTR_STACKSIZE, "POSIX_THREAD_ATTR_STACKSIZE",
643 SYSCONF, NOFLAGS,

645 _SC_THREAD_PRIORITY_SCHEDULING, "POSIX_THREAD_PRIORITY_SCHEDULING",
646 SYSCONF, NOFLAGS,
647 _SC_THREAD_PRIORITY_SCHEDULING, "POSIX_THREAD_PRIORITY_SCHEDULING",
648 SYSCONF, NOFLAGS,

650 _SC_THREAD_PRIO_INHERIT, "POSIX_THREAD_PRIO_INHERIT",
651 SYSCONF, NOFLAGS,
652 _SC_THREAD_PRIO_INHERIT, "POSIX_THREAD_PRIO_INHERIT",
653 SYSCONF, NOFLAGS,

655 _SC_THREAD_PRIO_PROTECT, "POSIX_THREAD_PRIO_PROTECT",
656 SYSCONF, NOFLAGS,
657 _SC_THREAD_PRIO_PROTECT, "POSIX_THREAD_PRIO_PROTECT",
658 SYSCONF, NOFLAGS,

660 _SC_THREAD_PROCESS_SHARED, "POSIX_THREAD_PROCESS_SHARED",
661 SYSCONF, NOFLAGS,
662 _SC_THREAD_PROCESS_SHARED, "POSIX_THREAD_PROCESS_SHARED",
663 SYSCONF, NOFLAGS,

665 _SC_THREAD_SAFE_FUNCTIONS, "POSIX_THREAD_SAFE_FUNCTIONS",
666 SYSCONF, NOFLAGS,
667 _SC_THREAD_SAFE_FUNCTIONS, "POSIX_THREAD_SAFE_FUNCTIONS",
668 SYSCONF, NOFLAGS,

670 _SC_TTY_NAME_MAX, "TTY_NAME_MAX", SYSCONF, NOFLAGS,
671 _SC_XOPEN_REALTIME, "XOPEN_REALTIME", SYSCONF, NOFLAGS,
672 _SC_XOPEN_REALTIME_THREADS, "XOPEN_REALTIME_THREADS", SYSCONF, NOFLAGS,

674 /*
675 * Due to ambiguity in the standard we need
676 * to recognize both the underscored name

```

```

677 * and the non-underscored version in the
678 * following variables.
679 */
680 _SC_XBS5_ILP32_OFF32, "XBS5_ILP32_OFF32", SYSCONF, NOFLAGS,
681 _SC_XBS5_ILP32_OFF32, "XBS5_ILP32_OFF32", SYSCONF, NOFLAGS,

683 _SC_XBS5_ILP32_OFFBIG, "XBS5_ILP32_OFFBIG", SYSCONF, NOFLAGS,
684 _SC_XBS5_ILP32_OFFBIG, "XBS5_ILP32_OFFBIG", SYSCONF, NOFLAGS,

686 _SC_XBS5_LP64_OFF64, "XBS5_LP64_OFF64", SYSCONF, NOFLAGS,
687 _SC_XBS5_LP64_OFF64, "XBS5_LP64_OFF64", SYSCONF, NOFLAGS,

689 _SC_XBS5_LPBIG_OFFBIG, "XBS5_LPBIG_OFFBIG", SYSCONF, NOFLAGS,
690 _SC_XBS5_LPBIG_OFFBIG, "XBS5_LPBIG_OFFBIG", SYSCONF, NOFLAGS,

692 /*
693 * UNIX 03 additions for sysconf() names, and names that are not in interfaces
694 */
695 _SC_REGEX, "POSIX_REGEX", SYSCONF, NOFLAGS,
696 _SC_SHELL, "POSIX_SHELL", SYSCONF, NOFLAGS,
697 _SC_HOST_NAME_MAX, "HOST_NAME_MAX", SYSCONF, NOFLAGS,
698 _SC_READER_WRITER_LOCKS, "POSIX_READER_WRITER_LOCKS", SYSCONF, NOFLAGS,
699 _SC_IPV6, "POSIX_IPV6", SYSCONF, NOFLAGS,
700 _SC_RAW_SOCKETS, "POSIX_RAW_SOCKETS", SYSCONF, NOFLAGS,
701 _SC_XOPEN_STREAMS, "XOPEN_STREAMS", SYSCONF, NOFLAGS,
702 _SC_SYMLOOP_MAX, "SYMLOOP_MAX", SYSCONF, NOFLAGS,
703 _SC_2_PBS, "POSIX2_PBS", SYSCONF, NOFLAGS,
704 _SC_2_PBS_ACCOUNTING, "POSIX2_PBS_ACCOUNTING", SYSCONF, NOFLAGS,
705 _SC_2_PBS_CHECKPOINT, "POSIX2_PBS_CHECKPOINT", SYSCONF, NOFLAGS,
706 _SC_2_PBS_LOCATE, "POSIX2_PBS_LOCATE", SYSCONF, NOFLAGS,
707 _SC_2_PBS_MESSAGE, "POSIX2_PBS_MESSAGE", SYSCONF, NOFLAGS,
708 _SC_2_PBS_TRACK, "POSIX2_PBS_TRACK", SYSCONF, NOFLAGS,

710 _SC_ADVISORY_INFO, "POSIX_ADVISORY_INFO", SYSCONF, NOFLAGS,
711 _SC_BARRIERS, "POSIX_BARRIERS", SYSCONF, NOFLAGS,
712 _SC_CLOCK_SELECTION, "POSIX_CLOCK_SELECTION", SYSCONF, NOFLAGS,
713 _SC_CPUTIME, "POSIX_CPUTIME", SYSCONF, NOFLAGS,
714 _SC_MONOTONIC_CLOCK, "POSIX_MONOTONIC_CLOCK", SYSCONF, NOFLAGS,
715 _SC_SPAWN, "POSIX_SPAWN", SYSCONF, NOFLAGS,
716 _SC_SPIN_LOCKS, "POSIX_SPIN_LOCKS", SYSCONF, NOFLAGS,
717 _SC_SPORADIC_SERVER, "POSIX_SPORADIC_SERVER", SYSCONF, NOFLAGS,
718 _SC_SS_REPL_MAX, "POSIX_SS_REPL_MAX", SYSCONF, NOFLAGS,
719 _SC_THREAD_CPUTIME, "POSIX_THREAD_CPUTIME", SYSCONF, NOFLAGS,
720 _SC_THREAD_SPORADIC_SERVER, "POSIX_THREAD_SPORADIC_SERVER", SYSCONF,
721 NOFLAGS,
722 _SC_TIMEOUTS, "POSIX_TIMEOUTS", SYSCONF, NOFLAGS,
723 _SC_TRACE, "POSIX_TRACE", SYSCONF, NOFLAGS,
724 _SC_TRACE_EVENT_FILTER, "POSIX_TRACE_EVENT_FILTER", SYSCONF, NOFLAGS,
725 _SC_TRACE_EVENT_NAME_MAX, "POSIX_TRACE_EVENT_NAME_MAX", SYSCONF,
726 NOFLAGS,
727 _SC_TRACE_INHERIT, "POSIX_TRACE_INHERIT", SYSCONF, NOFLAGS,
728 _SC_TRACE_LOG, "POSIX_TRACE_LOG", SYSCONF, NOFLAGS,
729 _SC_TRACE_NAME_MAX, "POSIX_TRACE_NAME_MAX", SYSCONF, NOFLAGS,
730 _SC_TRACE_SYS_MAX, "POSIX_TRACE_SYS_MAX", SYSCONF, NOFLAGS,
731 _SC_TRACE_USER_EVENT_MAX, "POSIX_TRACE_USER_EVENT_MAX", SYSCONF,
732 NOFLAGS,
733 _SC_TYPED_MEMORY_OBJECTS, "POSIX_TYPED_MEMORY_OBJECTS", SYSCONF,
734 NOFLAGS,

736 /*
737 * Due to an ambiguity in the specification, we need to recognize both
738 * the underscored POSIX name and the non-underscored name. In addition,
739 * we need to permit the use of the _V6_* environment names specified
740 * in unistd.h.
741 */
742 _SC_V6_ILP32_OFF32, "V6_ILP32_OFF32", SYSCONF, NOFLAGS,

```

```

743  _SC_V6_ILP32_OFF32,    "_POSIX_V6_ILP32_OFF32", SYSCONF,    NOFLAGS,
744  _SC_V6_ILP32_OFF32,    "POSIX_V6_ILP32_OFF32", SYSCONF,    NOFLAGS,

746  _SC_V6_ILP32_OFFBIG,   "_V6_ILP32_OFFBIG",      SYSCONF,    NOFLAGS,
747  _SC_V6_ILP32_OFFBIG,   "POSIX_V6_ILP32_OFFBIG", SYSCONF,    NOFLAGS,
748  _SC_V6_ILP32_OFFBIG,   "POSIX_V6_ILP32_OFFBIG", SYSCONF,    NOFLAGS,

750  _SC_V6_LP64_OFF64,     "_V6_LP64_OFF64",       SYSCONF,    NOFLAGS,
751  _SC_V6_LP64_OFF64,     "POSIX_V6_LP64_OFF64",   SYSCONF,    NOFLAGS,
752  _SC_V6_LP64_OFF64,     "POSIX_V6_LP64_OFF64",   SYSCONF,    NOFLAGS,

754  _SC_V6_LPBIG_OFFBIG,   "_V6_LPBIG_OFFBIG",     SYSCONF,    NOFLAGS,
755  _SC_V6_LPBIG_OFFBIG,   "POSIX_V6_LPBIG_OFFBIG", SYSCONF,    NOFLAGS,
756  _SC_V6_LPBIG_OFFBIG,   "POSIX_V6_LPBIG_OFFBIG", SYSCONF,    NOFLAGS,
757  /*
758  * System V Interface Definition
759  */
760  _SC_NPROCESSORS_CONF,  "NPROCESSORS_CONF",     SYSCONF,    NOFLAGS,
761  _SC_NPROCESSORS_MAX,   "NPROCESSORS_MAX",      SYSCONF,    NOFLAGS,
762  _SC_NPROCESSORS_ONLN,  "NPROCESSORS_ONLN",     SYSCONF,    NOFLAGS,
763  0,                      NULL,                    0,          0          /* end of table */
764  };

767  /*
768  * Print usage message.
769  */
770  static int
771  usage()
772  {
773
774      (void) fputs(gettext(
775          "usage:\tgetconf [-v specification] system_var\n"
776          "\tgetconf [-v specification] path_var pathname\n"
777          "\tgetconf -a\n"), stderr);
778      return (2);
779  }

782  static int
783  namecmp(const void *a, const void *b)
784  {
785      return (strcoll(((const struct sctab *)a)->name,
786          ((const struct sctab *)b)->name));
787  }

790  static int
791  getconf(struct sctab *scp, int argc, char *name, char *file)
792  {
793      register size_t len;
794      register char *buffer;
795      long value;

797      switch (scp->type) {
798      case SELFCONF:
799          if (argc > 2)
800              return (usage());
801          if (scp->flag & UNDEFINED ||
802              (!(scp->flag&UNSIGNED_VALUE) && scp->value == -1)) {
803              /* DO NOT TRANSLATE */
804              (void) printf("undefined\n");
805          } else
806              if (scp->flag & UNSIGNED_VALUE)
807                  (void) printf("%lu\n", scp->value);
808              else

```

```

809          (void) printf("%ld\n", scp->value);
810          break;

812      case SYSCONF:
813          if (argc > 2)
814              return (usage());
815          errno = 0;
816          if (scp->flag & UNDEFINED ||
817              (value = sysconf((int)scp->value)) == -1) {
818              if (errno == EINVAL) {
819                  (void) fprintf(stderr,
820                      gettext(INVAL_ARG), name);
821                  return (1);
822              } else {
823                  /* DO NOT TRANSLATE */
824                  (void) printf("undefined\n");
825                  return (0);
826              }
827          } else
828              (void) printf("%ld\n", value);
829          break;

831      case CONFSTR:
832          if (argc > 2)
833              return (usage());
834          errno = 0;
835          len = confstr((int)scp->value, NULL, (size_t)0);
836          if (len == 0)
837              if (errno == EINVAL) {
838                  (void) fprintf(stderr, gettext(INVAL_ARG),
839                      name);
840                  return (1);
841              } else {
842                  /* DO NOT TRANSLATE */
843                  (void) printf("undefined\n");
844                  return (0);
845              }
846          /*
847          * allocate space to store result of constr() into
848          */
849          if ((buffer = malloc(len)) == NULL) {
850              (void) fprintf(stderr,
851                  gettext("insufficient memory available"));
852              return (1);
853          }

855          assert(confstr((int)scp->value, buffer, len) != 0);

857          (void) printf("%s\n", buffer);
858          free(buffer);
859          break;

861      case PATHCONF:
862          if (argc != 3)
863              return (usage());
864          errno = 0;
865          if ((value = pathconf(file, (int)scp->value)) == -1) {
866              if (errno == EINVAL || errno == 0) {
867                  /* Does not support the association */
868                  /* DO NOT TRANSLATE */
869                  (void) printf("undefined\n");
870                  return (0);
871              } else {
872                  /* Probably problems with the pathname itself */
873                  if (aflag) {
874                      (void) printf("%s\n", file);

```

```

875     } else {
876         (void) printf("%s", file);
877     }
878     return (1);
879 }
880 }
881     (void) printf("%ld\n", value);
882     break;
883 }
884 return (0);
885 }

887 int
888 main(int argc, char **argv)
889 {
890     register struct sctab *scp;
891     int c;
892     int exstat = 0;

894     (void) setlocale(LC_ALL, "");
895     #if !defined(TEXT_DOMAIN) /* Should be defined by cc -D */
896     #define TEXT_DOMAIN "SYS_TEST"
897     #endif
898     (void) textdomain(TEXT_DOMAIN);

900     while ((c = getopt(argc, argv, "av:")) != -1)
901         switch (c) {
902             case 'a':
903                 aflag = 1;
904                 break;
905             case 'v':
906                 /*
907                  * Unix98 adds the -v option to allow
908                  * programming 'specifications' to be
909                  * indicated, for present purposes
910                  * the specification isn't really
911                  * doing anything of value, so for
912                  * the moment getopt just processes the
913                  * option value and argv[1] is adjusted.
914                  *
915                  * At some later date we might want to
916                  * do specification processing at this
917                  * point.
918                  */
919                 break;
920             case '?':
921                 return (usage());
922         }
923     argc -= optind-1;
924     argv += optind-1;

926     if ((aflag && argc >= 2) || (!aflag && argc < 2))
927         return (usage());
928     if (aflag) {

930     #define TabStop      8
931     #define RightColumn  32
932     #define DefPathName  "."

934         /*
935          * sort the table by the "name" field
936          * so we print it in sorted order
937          */
938         qsort(&sctab[0], (sizeof (sctab)/sizeof (struct sctab))-1,
939             sizeof (struct sctab), namecmp);

```

```

941     /*
942     * print all the known symbols and their values
943     */
944     for (scp = &sctab[0]; scp->name != NULL; ++scp) {
945         int stat;

947         /*
948         * create 2 columns:
949         * variable name in the left column,
950         * value in the right column.
951         * The right column starts at a tab stop.
952         */
953         (void) printf("%s:\t", scp->name);

955         c = strlen(scp->name) + 1;
956         c = (c+TabStop) / TabStop, c *= TabStop;
957         for (; c < RightColumn; c += TabStop)
958             (void) putchar('\t');

960         /*
961         * for pathconf() related variables that require
962         * a pathname, use "."
963         */
964         stat = getconf(scp, scp->type == PATHCONF ? 3 : 2,
965             scp->name, DefPathName);

967         exstat |= stat;

969         /*
970         * if stat != 0, then an error message was already
971         * printed in getconf(),
972         * so don't need to print one here
973         */
974     } else {

977         /*
978         * find the name of the specified variable (argv[1])
979         * and print its value.
980         */
981         for (scp = &sctab[0]; scp->name != NULL; ++scp)
982             if (strcmp(argv[1], scp->name) == 0) {
983                 exstat = getconf(scp, argc, argv[1], argv[2]);
984                 break;
985             }

987         /*
988         * if at last entry in table, then the user specified
989         * variable is invalid
990         */
991         if (scp->name == NULL) {
992             errno = EINVAL;
993             (void) fprintf(stderr, gettext(INVAL_ARG), argv[1]);
994             return (1);
995         }

996     }
997     return (exstat);
998 }

```

```

*****
17516 Fri Sep 27 10:14:08 2013
new/usr/src/man/man1/getconf.1
4146 getconf(1) does not support XATTR_ENABLED, XATTR_EXISTS, SATTR_ENABLED, SAT
*****
1 \" te
2.\" Copyright (c) 1992, X/Open Company Limited All Rights Reserved
3.\" Portions Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved
4.\" Portions Copyright (c) 1982-2007 AT&T Knowledge Ventures
5.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
6.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
7.\" are reprinted and reproduced in electronic form in the Sun OS Reference Manu
8.\" and Electronics Engineers, Inc and The Open Group. In the event of any discr
9.\" This notice shall appear on any product containing this material.
10.\" The contents of this file are subject to the terms of the Common Development
11.\" See the License for the specific language governing permissions and limitat
12.\" the fields enclosed by brackets \"[]\" replaced with your own identifying info
13.\".TH GETCONF 1 \"Sep 27, 2013\"
13.\".TH GETCONF 1 \"Jan 16, 2013\"
14.SH NAME
15.getconf \- get configuration values
16.SH SYNOPSIS
17.SS \"/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf\"
18.LP
19.nf
20 \fB/usr/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIsystem_var\fR
21 .fi

23.LP
24.nf
25 \fB/usr/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIpath_var\fR \fIpathname
26 .fi

28.LP
29.nf
30 \fB/usr/bin/getconf\fR \fB-a\fR
31 .fi

33.LP
34.nf
35 \fB/usr/xpg4/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIsystem_var\fR
36 .fi

38.LP
39.nf
40 \fB/usr/xpg4/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIpath_var\fR \fIpat
41 .fi

43.LP
44.nf
45 \fB/usr/xpg4/bin/getconf\fR \fB-a\fR
46 .fi

48.LP
49.nf
50 \fB/usr/xpg6/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIsystem_var\fR
51 .fi

53.LP
54.nf
55 \fB/usr/xpg6/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIpath_var\fR \fIpat
56 .fi

58.LP
59.nf
60 \fB/usr/xpg6/bin/getconf\fR \fB-a\fR

```

```

61 .fi

63.SS "ksh93"
64.LP
65.nf
66 \fBgetconf\fR [\fB-abdlpqrtw\fR] [\fB-c\fR RE] [\fB-n\fR RE] [\fB-s\fR RE] [\fB-
67 .fi

69.SH DESCRIPTION
70.SS \"/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf\"
71.sp
72.LP
73 In the first synopsis form, the \fBgetconf\fR utility writes to the standard
74 output the value of the variable specified by \fIsystem_var\fR, in accordance
75 with \fIspecification\fR if the \fB-v\fR option is used.
76.sp
77.LP
78 In the second synopsis form, \fBgetconf\fR writes to the standard output the
79 value of the variable specified by \fIpath_var\fR for the path specified by
80 \fIpathname\fR, in accordance with \fIspecification\fR if the \fB-v\fR option
81 is used.
82.sp
83.LP
84 In the third synopsis form, \fBconfig\fR writes to the standard output the
85 names of the current system configuration variables.
86.sp
87.LP
88 The value of each configuration variable is determined as if it were obtained
89 by calling the function from which it is defined to be available. The value
90 reflects conditions in the current operating environment.
91.SS "ksh93"
92.sp
93.LP
94 The \fBgetconf\fR built-in in \fBksh93\fR is associated with the \fB/bin\fR and
95 \fB/usr/bin\fR paths. It is invoked when \fBgetconf\fR is executed without a
96 pathname prefix and the pathname search finds a \fB/bin/getconf\fR or
97 \fB/usr/bin/getconf\fR executable.
98.sp
99.LP
100 \fBgetconf\fR displays the system configuration value for \fIname\fR. If
101 \fIname\fR is a filesystem specific variable, the value is determined relative
102 to path, or, the current directory, if path is omitted. If \fIvalue\fR is
103 specified, \fBgetconf\fR attempts to change the process local value to
104 \fIvalue\fR. \fB-\fR can be used in place of path when it is not relevant. Only
105 writable variables can be set. \fBreadonly\fR variables cannot be changed.
106.sp
107.LP
108 The current value for \fIname\fR is written to the standard output. If
109 \fIname\fR is valid but undefined, \fBundefined\fR is written to the standard
110 output. If \fIname\fR is invalid or an error occurs in determining its value, a
111 diagnostic is written to standard error and \fBgetconf\fR exits with a
112 \fBnon-zero\fR exit status.
113.sp
114.LP
115 More than one variable can be set or queried by providing the \fIname path
116 value\fR 3-tuple for each variable, specifying \fB-\fR for value when querying.
117.sp
118.LP
119 If no operands are specified, all known variables are written in
120 \fIname\fR \fB=\fR \fIvalue\fR form to the standard output, one per line. Only
121 one of \fB--call\fR, \fB--name\fR or \fB--standard\fR can be specified.
122.SH OPTIONS
123.SS \"/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf\"
124.sp
125.LP
126 The following options are supported by \fB/usr/bin/getconf\fR,

```



```

127 \fB/usr/xpg4/bin/getconf\fR, and \fB/usr/xpg6/bin/getconf\fR
128 .sp
129 .ne 2
130 .na
131 \fB\fB-a\fR\fR
132 .ad
133 .RS 19n
134 Writes the names of the current system configuration variables to the standard
135 output.
136 .RE

138 .sp
139 .ne 2
140 .na
141 \fB\fB-v\fR\fR\fI\spe\fi\fcation\fR\fR
142 .ad
143 .RS 19n
144 Gives the specification which governs the selection of values for configuration
145 variables.
146 .RE

148 .SS "ksh93"
149 .sp
150 .LP
151 The following options are supported by the \fBgetconf\fR built-in command in
152 \fBksh93\fR:
153 .sp
154 .ne 2
155 .na
156 \fB\fB-a\fR\fR
157 .ad
158 .br
159 .na
160 \fB\fB--all\fR\fR
161 .ad
162 .RS 24n
163 Call the native \fBgetconf -a\fR.
164 .RE

166 .sp
167 .ne 2
168 .na
169 \fB\fB-b\fR\fR
170 .ad
171 .br
172 .na
173 \fB\fB--base\fR\fR
174 .ad
175 .RS 24n
176 List base variable name without \fBcall\fR and standard prefixes.
177 .RE

179 .sp
180 .ne 2
181 .na
182 \fB\fB-c\fR\fR
183 .ad
184 .br
185 .na
186 \fB\fB--call=\fR\fIRE\fR\fR
187 .ad
188 .RS 24n
189 Display variables with \fBcall\fR prefix that matches \fR\fIRE\fR. The \fBcall\fR
190 prefixes are:
191 .RS +4
192 .TP

```

```

193 .ie t \(\bu
194 .el o
195 \fB\fB-FC\fR
196 .sp
197 \fB\fB-FCstr\fR(3C),
198 .RE
199 .RS +4
200 .TP
201 .ie t \(\bu
202 .el o
203 \fB\fB-PC\fR
204 .sp
205 \fB\fB-pathconf\fR(2)
206 .RE
207 .RS +4
208 .TP
209 .ie t \(\bu
210 .el o
211 \fB\fB-SC\fR
212 .sp
213 \fB\fB-sysconf\fR(3C)
214 .RE
215 .RS +4
216 .TP
217 .ie t \(\bu
218 .el o
219 \fB\fB-SI\fR
220 .sp
221 \fB\fB-sysinfo\fR(2)
222 .RE
223 .RS +4
224 .TP
225 .ie t \(\bu
226 .el o
227 \fB\fB-XX\fR
228 .sp
229 Constant value
230 .RE
231 .RE

233 .sp
234 .ne 2
235 .na
236 \fB\fB-d\fR\fR
237 .ad
238 .br
239 .na
240 \fB\fB--defined\fR\fR
241 .ad
242 .RS 24n
243 Only display defined values when no operands are specified.
244 .RE

246 .sp
247 .ne 2
248 .na
249 \fB\fB-l\fR\fR
250 .ad
251 .br
252 .na
253 \fB\fB--lowercase\fR\fR
254 .ad
255 .RS 24n
256 List variable names in lower case.
257 .RE

```

```

259 .sp
260 .ne 2
261 .na
262 \fB\fB-n\fR\fR
263 .ad
264 .br
265 .na
266 \fB\fB--name=\fR\fIRE\fR\fR
267 .ad
268 .RS 24n
269 Display variables with names that match \fIRE\fR.
270 .RE

272 .sp
273 .ne 2
274 .na
275 \fB\fB-p\fR\fR
276 .ad
277 .br
278 .na
279 \fB\fB--portable\fR\fR
280 .ad
281 .RS 24n
282 Display the named writable variables and values in a form that can be directly
283 executed by \fBksh93\fR(1) to set the values. If \fIname\fR is omitted, lists
284 all writable variables.
285 .RE

287 .sp
288 .ne 2
289 .na
290 \fB\fB-q\fR\fR
291 .ad
292 .br
293 .na
294 \fB\fB--quote\fR\fR
295 .ad
296 .RS 24n
297 Quote values (\fB"..."\fR).
298 .RE

300 .sp
301 .ne 2
302 .na
303 \fB\fB-r\fR\fR
304 .ad
305 .br
306 .na
307 \fB\fB--readonly\fR\fR
308 .ad
309 .RS 24n
310 Display the named \fBreadonly\fR variables in \fIname\fR\fB=\fR\fIvalue\fR
311 form. If \fIname\fR is omitted, lists all \fBreadonly\fR variables.
312 .RE

314 .sp
315 .ne 2
316 .na
317 \fB\fB-s\fR\fR
318 .ad
319 .br
320 .na
321 \fB\fB--standard=\fR\fIRE\fR\fR
322 .ad
323 .RS 24n
324 Display variables with standard prefix that matches \fIRE\fR. Use the

```

```

325 \fB--table\fR option to view all standard prefixes, including local additions.
326 .sp
327 The standard prefixes available on all systems are:
328 .sp
329 .in +2
330 .nf

332 AES
333 AST
334 C
335 GNU
336 POSIX
337 SVID
338 XBS5
339 XOPEN
340 XPG
341 .fi
342 .in -2
343 .sp

345 .RE

347 .sp
348 .ne 2
349 .na
350 \fB\fB-t\fR\fR
351 .ad
352 .br
353 .na
354 \fB\fB--table\fR\fR
355 .ad
356 .RS 24n
357 Display the internal table that contains the name, standard, standard section,
358 and system call symbol prefix for each variable.
359 .RE

361 .sp
362 .ne 2
363 .na
364 \fB\fB-w\fR\fR
365 .ad
366 .br
367 .na
368 \fB\fB--writable\fR\fR
369 .ad
370 .RS 24n
371 Display the named writable variables in \fIname\fR\fB=\fR\fIvalue\fR form. If
372 \fIname\fR is omitted, lists all writable variables.
373 .RE

375 .sp
376 .ne 2
377 .na
378 \fB\fB-v\fR\fR
379 .ad
380 .br
381 .na
382 \fB\fB--specification=name\fR\fR
383 .ad
384 .RS 24n
385 Call the native \fBgetconf -v\fR \fIname\fR.
386 .RE

388 .SH OPERANDS
389 .SS "/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf"
390 .sp

```

```

391 .LP
392 The following operands are supported by \fB/usr/bin/getconf\fR,
393 \fB/usr/xpg4/bin/getconf\fR, and \fB/usr/xpg6/bin/getconf\fR:
394 .sp
395 .ne 2
396 .na
397 \fB\fIpath_var\fR\fR
398 .ad
399 .RS 12n
400 A name of a configuration variable whose value is available from the
401 \fB\bpathconf\fR(2) function. All of the values in the following table are
402 supported:
403 .RE

405 .sp

407 .sp
408 .TS
409 1 1 1
410 1 1 1 .
411 LINK_MAX          PIPE_BUF          _POSIX_CHOWN_RESTRICTED
412 MAX_CANON        SATTR_ENABLED      _POSIX_NO_TRUNC
413 MAX_INPUT        SATTR_EXISTS       _POSIX_VDISABLE
414 NAME_MAX          XATTR_ENABLED
415 PATH_MAX          XATTR_EXISTS
416 LINK_MAX          NAME_MAX           _POSIX_CHOWN_RESTRICTED
417 MAX_CANON        PATH_MAX           _POSIX_NO_TRUNC
418 MAX_INPUT        PIPE_BUF           _POSIX_VDISABLE
419 .TE

418 .sp
419 .ne 2
420 .na
421 \fB\fIpathname\fR\fR
422 .ad
423 .RS 14n
424 A path name for which the variable specified by \fIpath_var\fR is to be
425 determined.
426 .RE

428 .sp
429 .ne 2
430 .na
431 \fB\fIsystem_var\fR\fR
432 .ad
433 .RS 14n
434 A name of a configuration variable whose value is available from
435 \fB\bconfstr\fR(3C) or \fB\bsysconf\fR(3C). All of the values in the following
436 table are supported:
437 .RE

439 .sp

441 .sp
442 .TS
443 1 1
444 1 1 .
445 ARG_MAX BC_BASE_MAX
446 BC_DIM_MAX BC_SCALE_MAX
447 BC_STRING_MAX CHAR_BIT
448 CHARCLASS_NAME_MAX CHAR_MAX
449 CHAR_MIN CHILD_MAX
450 CLK_TCK COLL_WEIGHTS_MAX
451 CS_PATH EXPR_NEST_MAX
452 HOST_NAME_MAX INT_MAX
453 INT_MIN LFS64_CFLAGS

```

```

454 LFS64_LDFLAGS LFS64_LIBS
455 LFS64_LINTFLAGS LFS_CFLAGS
456 LFS_LDFLAGS LFS_LIBS
457 LFS_LINTFLAGS LINE_MAX
458 LONG_BIT LONG_MAX
459 LONG_MIN MB_LEN_MAX
460 NGROUPS_MAX NL_ARGMAX
461 NL_LANGMAX NL_MSGMAX
462 NL_NMAX NL_SETMAX
463 NL_TEXTMAX NPROCESSORS_CONF
464 NPROCESSORS_MAX NPROCESSORS_ONLN
465 NZERO OPEN_MAX
466 POSIX2_BC_BASE_MAX POSIX2_BC_DIM_MAX
467 POSIX2_BC_SCALE_MAX POSIX2_BC_STRING_MAX
468 POSIX2_C_BIND POSIX2_C_DEV
469 POSIX2_CHAR_TERM POSIX2_COLL_WEIGHTS_MAX
470 POSIX2_C_VERSION POSIX2_EXPR_NEST_MAX
471 POSIX2_FORT_DEV POSIX2_FORT_RUN
472 POSIX2_LINE_MAX POSIX2_LOCALEDEF
473 POSIX2_RE_DUP_MAX POSIX2_SW_DEV
474 POSIX2_SYMLINKS POSIX2_UPE
475 POSIX2_VERSION POSIX_ALLOC_SIZE_MIN
476 POSIX_REC_INCR_XFER_SIZE POSIX_REC_MAX_XFER_SIZE
477 POSIX_REC_MIN_XFER_SIZE POSIX_REC_XFER_ALIGN
478 POSIX_V6_ILP32_OFF32 POSIX_V6_ILP32_OFF32_CFLAGS
479 POSIX_V6_ILP32_OFF32_LDFLAGS POSIX_V6_ILP32_OFF32_LIBS
480 POSIX_V6_ILP32_OFFBIG POSIX_V6_ILP32_OFFBIG_CFLAGS
481 POSIX_V6_ILP32_OFFBIG_LDFLAGS POSIX_V6_ILP32_OFFBIG_LIBS
482 POSIX_V6_LP64_OFF64 POSIX_V6_LP64_OFF64_CFLAGS
483 POSIX_V6_LP64_OFF64_LDFLAGS POSIX_V6_LP64_OFF64_LIBS
484 POSIX_V6_LPBIG_OFFBIG POSIX_V6_LPBIG_OFFBIG_CFLAGS
485 POSIX_V6_LPBIG_OFFBIG_LDFLAGS POSIX_V6_LPBIG_OFFBIG_LIBS
486 POSIX_V6_WIDTH_RESTRICTED_ENVS SYMLINK_MAX
487 SYMLINK_MAX _POSIX2_BC_BASE_MAX
488 _POSIX2_BC_DIM_MAX _POSIX2_BC_SCALE_MAX
489 _POSIX2_BC_STRING_MAX _POSIX2_CHARCLASS_NAME_MAX
490 _POSIX2_CHAR_TERM _POSIX2_COLL_WEIGHTS_MAX
491 _POSIX2_C_BIND _POSIX2_C_DEV
492 _POSIX2_C_VERSION _POSIX2_EXPR_NEST_MAX
493 _POSIX2_FORT_DEV _POSIX2_FORT_RUN
494 _POSIX2_LINE_MAX _POSIX2_LOCALEDEF
495 _POSIX2_PBS _POSIX2_PBS_ACCOUNTING
496 _POSIX2_PBS_CHECKPOINT _POSIX2_PBS_LOCATE
497 _POSIX2_PBS_MESSAGE _POSIX2_PBS_TRACK
498 _POSIX2_RE_DUP_MAX _POSIX2_SW_DEV
499 _POSIX2_UPE _POSIX2_VERSION
500 _POSIX_ADVISORY_INFO _POSIX_AIO_LISTIO_MAX
501 _POSIX_AIO_MAX _POSIX_ARG_MAX
502 _POSIX_ASYNC_IO _POSIX_BARRIERS
503 _POSIX_CHILD_MAX _POSIX_CLOCKRES_MIN
504 _POSIX_CLOCK_SELECTION _POSIX_CPUTIME
505 _POSIX_DELAYTIMER_MAX _POSIX_HOST_NAME_MAX
506 _POSIX_IPV6 _POSIX_JOB_CONTROL
507 _POSIX_LINK_MAX _POSIX_LOGIN_NAME_MAX
508 _POSIX_MAX_CANON _POSIX_MAX_INPUT
509 _POSIX_MONOTONIC_CLOCK _POSIX_MQ_OPEN_MAX
510 _POSIX_MQ_PRIO_MAX _POSIX_NAME_MAX
511 _POSIX_NGROUPS_MAX _POSIX_OPEN_MAX
512 _POSIX_PATH_MAX _POSIX_PIPE_BUF
513 _POSIX_PRIO_IO _POSIX_RAW_SOCKETS
514 _POSIX_READER_WRITER_LOCKS _POSIX_REGEX
515 _POSIX_RE_DUP_MAX _POSIX_RTSIG_MAX
516 _POSIX_SAVED_IDS _POSIX_SEM_NSEMS_MAX
517 _POSIX_SEM_VALUE_MAX _POSIX_SHELL
518 _POSIX_SIGQUEUE_MAX _POSIX_SPAWN
519 _POSIX_SPIN_LOCKS _POSIX_SPORADIC_SERVER

```

```

520 _POSIX_SSIZE_MAX      _POSIX_SS_REPL_MAX
521 _POSIX_STREAM_MAX     _POSIX_SYMLINK_MAX
522 _POSIX_SYMLINK_MAX    _POSIX_SYNC_IO
523 _POSIX_THREAD_ATTR_STACKADDR  _POSIX_THREAD_ATTR_STACKSIZE
524 _POSIX_THREAD_CPUTIME  _POSIX_THREAD_DESTRUCTOR_ITERATIONS
525 _POSIX_THREAD_KEYS_MAX  _POSIX_THREAD_PRIORITY_SCHEDULING
526 _POSIX_THREAD_Prio_INHERIT  _POSIX_THREAD_Prio_PROTECT
527 _POSIX_THREAD_PROCESS_SHARED  _POSIX_THREAD_SAFE_FUNCTIONS
528 _POSIX_THREAD_Sporadic_SERVER  _POSIX_THREAD_THREADS_MAX
529 _POSIX_TIMEOUTS _POSIX_TIMER_MAX
530 _POSIX_TRA_POSIX_TIMER_MAXCE  _POSIX_TIMESTAMP_RESOLUTION
531 _POSIX_TRACE_EVENT_FILTER  _POSIX_TRACE_EVENT_NAME_MAX
532 _POSIX_TRACE_INHERIT  _POSIX_TRACE_LOG
533 _POSIX_TRACE_NAME_MAX  _POSIX_TRACE_SYS_MAX
534 _POSIX_TRACE_USER_EVENT_MAX  _POSIX_TTY_NAME_MAX
535 _POSIX_TYPED_MEMORY_OBJECTS  _POSIX_TZNAME_MAX
536 _POSIX_VERSION  _POSIX_V6_ILP32_OFF32
537 _POSIX_V6_ILP32_OFFBIG  _POSIX_V6_LP64_OFF64
538 _POSIX_V6_LPBIG_OFFBIG  _V6_ILP32_OFF32
539 _V6_ILP32_OFFBIG  _V6_LP64_OFF64
540 _V6_LPBIG_OFFBIG  RE_DUP_MAX
541 SCHAR_MAX      SCHAR_MIN
542 SHRT_MAX      SHRT_MIN
543 SSIZE_MAX      STREAM_MAX
544 TMP_MAX  TZNAME_MAX
545 UCHAR_MAX      UINT_MAX
546 ULONG_MAX      USHRT_MAX
547 WORD_BIT      XBS5_ILP32_OFF32
548 XBS5_ILP32_OFF32_CFLAGS  XBS5_ILP32_OFF32_LDFLAGS
549 XBS5_ILP32_OFF32_LIBS  XBS5_ILP32_OFF32_LINTFLAGS
550 XBS5_ILP32_OFFBIG  XBS5_ILP32_OFFBIG_CFLAGS
551 XBS5_ILP32_OFFBIG_LDFLAGS  XBS5_ILP32_OFFBIG_LIBS
552 XBS5_ILP32_OFFBIG_LINTFLAGS  XBS5_LP64_OFF64
553 XBS5_LP64_OFF64_CFLAGS  XBS5_LP64_OFF64_LDFLAGS
554 XBS5_LP64_OFF64_LIBS  XBS5_LP64_OFF64_LINTFLAGS
555 XBS5_LPBIG_OFFBIG  XBS5_LPBIG_OFFBIG_CFLAGS
556 XBS5_LPBIG_OFFBIG_LDFLAGS  XBS5_LPBIG_OFFBIG_LIBS
557 XBS5_LPBIG_OFFBIG_LINTFLAGS  _XOPEN_CRYPT
558 _XOPEN_ENH_I18N  _XOPEN_IOV_MAX
559 _XOPEN_LEGACY  _XOPEN_NAME_MAX
560 _XOPEN_PATH_MAX  _XOPEN_SHM
561 _XOPEN_STREAMS  _XOPEN_VERSION
562 _XOPEN_XCU_VERSION  _XOPEN_XPG2
563 _XOPEN_XPG3  _XOPEN_XPG4
564 .TE

566 .sp
567 .LP
568 The symbol \fBPATH\fR also is recognized, yielding the same value as the
569 \fBconfstr()\fR name value \fBCS_PATH\fR.
570 .SH USAGE
571 .sp
572 .LP
573 See \fBlargefile\fR(5) for the description of the behavior of
574 \fB/usr/bin/getconf\fR when encountering files greater than or equal to 2 Gbyte
575 ( 231 bytes).
576 .SH EXAMPLES
577 .LP
578 \fBExample 1 \fRWriting the Value of a Variable
579 .sp
580 .LP
581 This example illustrates the value of \fB{NGROUPS_MAX}\fR

```

```

586 example% \fBgetconf NGROUPS_MAX\fR
587 .fi
588 .in -2
589 .sp

591 .LP
592 \fBExample 2 \fRWriting the Value of a Variable for a Specific Directory
593 .sp
594 .LP
595 This example illustrates the value of \fBNAME_MAX\fR for a specific directory:

597 .sp
598 .in +2
599 .nf
600 example% \fBgetconf NAME_MAX /usr\fR
601 .fi
602 .in -2
603 .sp

605 .LP
606 \fBExample 3 \fRDealing with Unspecified Results
607 .sp
608 .LP
609 This example shows how to deal more carefully with results that might be
610 unspecified:

612 .sp
613 .in +2
614 .nf
615 \fBif value=$(getconf PATH_MAX /usr); then
616 if [ "$value" = "undefined" ]; then
617 echo PATH_MAX in /usr is infinite.
618 else
619 echo PATH_MAX in /usr is $value.
620 fi
621 else
622 echo Error in getconf.
623 fi\fR
624 .fi
625 .in -2

627 .sp
628 .LP
629 For example:

631 .sp
632 .in +2
633 .nf
634 \fBsystemconf(_SC_POSIX_C_BIND);\fR
635 .fi
636 .in -2

638 .sp
639 .LP
640 and

642 .sp
643 .in +2
644 .nf
645 \fBsystem("getconf POSIX2_C_BIND);\fR
646 .fi
647 .in -2

649 .sp
650 .LP
651 in a C program could give different answers. The \fBsystemconf\fR call supplies a

```

652 value that corresponds to the conditions when the program was either compiled
 653 or executed, depending on the implementation. The `\fBsystem` command
 654 `\fBgetconf` always supplies a value corresponding to conditions when the
 655 program is executed.

657 .SH ENVIRONMENT VARIABLES
 658 .sp
 659 .LP
 660 See `\fBenviron`(5) for descriptions of the following environment variables
 661 that affect the execution of `\fBgetconf`: `\fBBLANG`, `\fBLC_ALL`,
 662 `\fBLC_CTYPE`, `\fBLC_MESSAGES`, and `\fBNLSPATH`.
 663 .sp
 664 .LP
 665 The following environment variable is supported by the `\fBksh93` built-in
 666 `\fBgetconf` command:
 667 .sp
 668 .ne 2
 669 .na
 670 `\fB\fB_AST_FEATURES`
 671 .ad
 672 .RS 17n
 673 Local writable values that are different from the default are stored in the
 674 `\fB_AST_FEATURES` environment variable. The `\fB_AST_FEATURES` value is a
 675 space-separated list of `\fIname` path value 3-tuples, where `\fIname` is the
 676 system configuration name, `\fIpath` is the corresponding path, `\fB` if no
 677 path is applicable, and `\fIvalue` is the system configuration value.
 678 .RE

680 .SH EXIT STATUS
 681 .sp
 682 .LP
 683 The following exit values are returned:
 684 .sp
 685 .ne 2
 686 .na
 687 `\fB\fB0`
 688 .ad
 689 .RS 6n
 690 The specified variable is valid and information about its current state was
 691 written successfully.
 692 .RE

694 .sp
 695 .ne 2
 696 .na
 697 `\fB\fB>0`
 698 .ad
 699 .RS 6n
 700 An error occurred.
 701 .RE

703 .SH ATTRIBUTES
 704 .sp
 705 .LP
 706 See `\fBattributes`(5) for descriptions of the following attributes:
 707 .SS `"/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf"`
 708 .sp

710 .sp
 711 .TS
 712 box;
 713 c | c
 714 l | l .
 715 ATTRIBUTE TYPE ATTRIBUTE VALUE
 716 _
 717 Interface Stability Committed

718 _
 719 Standard See `\fBstandards`(5).
 720 .TE

722 .SS "ksh93"
 723 .sp

725 .sp
 726 .TS
 727 box;
 728 c | c
 729 l | l .
 730 ATTRIBUTE TYPE ATTRIBUTE VALUE
 731 _
 732 Interface Stability See below.
 733 .TE

735 .sp
 736 .LP
 737 The `\fBksh93` built-in binding to `\fB/bin` and `\fB/usr/bin` is Volatile.
 738 The built-in command-line interface is Committed.
 739 .SH SEE ALSO
 740 .sp
 741 .LP
 742 `\fBksh93`(1), `\fBsh`(1), `\fBpathconf`(2), `\fBsysinfo`(2),
 743 `\fBconfstr`(3C), `\fBsysconf`(3C), `\fBattributes`(5), `\fBenviron`(5),
 744 `\fBlargefile`(5), `\fBstandards`(5)