
 32380 Wed Jan 16 16:29:30 2013
 new/usr/src/cmd/getconf/getconf.c
 452 /usr/!(|xpg4/|xpg6/)|bin/getconf need to support NPROCESSORS_
 Reviewed by: Garrett D'Amore <garrett@damore.org>

 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_DIM_MAX, "POSIX2_BC_DIM_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 _POSIX_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_RT_SIG_MAX, "_POSIX_RT_SIG_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,
182 "POSIX_THREAD_DESTRUCTOR_ITERATIONS", SELFCNF, NOFLAGS,
183 _POSIX_THREAD_KEYS_MAX,
184 "POSIX_THREAD_KEYS_MAX", SELFCNF, NOFLAGS,
185 _POSIX_THREAD_THREADS_MAX,
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,
225     /* UNIX 03 Additions */
227     /* Advisory Information (ADV) */
228     _PC_ALLOC_SIZE_MIN,  "POSIX_ALLOC_SIZE_MIN", PATHCONF, NOFLAGS,
229     _PC_REC_INCR_XFER_SIZE, "POSIX_REC_INCR_XFER_SIZE", PATHCONF, NOFLAGS,
230     _PC_REC_MAX_XFER_SIZE, "POSIX_REC_MAX_XFER_SIZE", PATHCONF, NOFLAGS,
231     _PC_REC_MIN_XFER_SIZE, "POSIX_REC_MIN_XFER_SIZE", PATHCONF, NOFLAGS,
232     _PC_REC_XFER_ALIGN,  "POSIX_REC_XFER_ALIGN", PATHCONF, NOFLAGS,
234     _PC_2_SYMLINKS,     "POSIX2_SYMLINKS",   PATHCONF, NOFLAGS,
235     _PC_SYMLINK_MAX,    "SYMLINK_MAX",       PATHCONF, NOFLAGS,
237     /* UNIX 08 additions */
239     _PC_TIMESTAMP_RESOLUTION, "_POSIX_TIMESTAMP_RESOLUTION",
240     PATHCONF, NOFLAGS,
242     /* confstr() variables */
244     /* POSIX.2-1992 description of getconf utility */
245     _CS_PATH,           "PATH",              CONFSTR, NOFLAGS,
247     /* POSIX.2-1992 table B-18 */
248     _CS_PATH,           "CS_PATH",           CONFSTR, NOFLAGS,
249     _CS_PATH,           "_CS_PATH",          CONFSTR, NOFLAGS,
251     /* command names for large file configuration information */
252     /* large file compilation environment configuration */
254     _CS_LFS_CFLAGS,     "LFS_CFLAGS",       CONFSTR, NOFLAGS,
255     _CS_LFS_LDFLAGS,   "LFS_LDFLAGS",     CONFSTR, NOFLAGS,
256     _CS_LFS_LIBS,      "LFS_LIBS",        CONFSTR, NOFLAGS,
257     _CS_LFS_LINTFLAGS, "LFS_LINTFLAGS",   CONFSTR, NOFLAGS,
259     /* transitional large file interface configuration */
261     _CS_LFS64_CFLAGS,  "LFS64_CFLAGS",    CONFSTR, NOFLAGS,
262     _CS_LFS64_LDFLAGS, "LFS64_LDFLAGS",  CONFSTR, NOFLAGS,
263     _CS_LFS64_LIBS,   "LFS64_LIBS",     CONFSTR, NOFLAGS,
264     _CS_LFS64_LINTFLAGS, "LFS64_LINTFLAGS", CONFSTR, NOFLAGS,
266     /*
267     * UNIX 98:
268     * confstr additions, these support
269     * programming specifications/environments.
270     *
271     * Unfortunately there is some uncertainty with
272     * the standard at this point about the constant
273     * name, thus we need to recognize BOTH with and
274     * without the initial underscore.
275     */
276     _CS_XBS5_ILP32_OFF32_CFLAGS, "XBS5_ILP32_OFF32_CFLAGS",
277     CONFSTR, NOFLAGS,
278     _CS_XBS5_ILP32_OFF32_CFLAGS, "_XBS5_ILP32_OFF32_CFLAGS",
279     CONFSTR, NOFLAGS,

```

```

281     _CS_XBS5_ILP32_OFF32_LDFLAGS, "XBS5_ILP32_OFF32_LDFLAGS",
282     CONFSTR, NOFLAGS,
283     _CS_XBS5_ILP32_OFF32_LDFLAGS, "_XBS5_ILP32_OFF32_LDFLAGS",
284     CONFSTR, NOFLAGS,
286     _CS_XBS5_ILP32_OFF32_LIBS,    "XBS5_ILP32_OFF32_LIBS",
287     CONFSTR, NOFLAGS,
288     _CS_XBS5_ILP32_OFF32_LIBS,    "_XBS5_ILP32_OFF32_LIBS",
289     CONFSTR, NOFLAGS,
291     _CS_XBS5_ILP32_OFF32_LINTFLAGS, "XBS5_ILP32_OFF32_LINTFLAGS",
292     CONFSTR, NOFLAGS,
293     _CS_XBS5_ILP32_OFF32_LINTFLAGS, "_XBS5_ILP32_OFF32_LINTFLAGS",
294     CONFSTR, NOFLAGS,
296     _CS_XBS5_ILP32_OFFBIG_CFLAGS,  "XBS5_ILP32_OFFBIG_CFLAGS",
297     CONFSTR, NOFLAGS,
298     _CS_XBS5_ILP32_OFFBIG_CFLAGS,  "_XBS5_ILP32_OFFBIG_CFLAGS",
299     CONFSTR, NOFLAGS,
301     _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "XBS5_ILP32_OFFBIG_LDFLAGS",
302     CONFSTR, NOFLAGS,
303     _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "_XBS5_ILP32_OFFBIG_LDFLAGS",
304     CONFSTR, NOFLAGS,
306     _CS_XBS5_ILP32_OFFBIG_LIBS,    "XBS5_ILP32_OFFBIG_LIBS",
307     CONFSTR, NOFLAGS,
308     _CS_XBS5_ILP32_OFFBIG_LIBS,    "_XBS5_ILP32_OFFBIG_LIBS",
309     CONFSTR, NOFLAGS,
311     _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "XBS5_ILP32_OFFBIG_LINTFLAGS",
312     CONFSTR, NOFLAGS,
313     _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "_XBS5_ILP32_OFFBIG_LINTFLAGS",
314     CONFSTR, NOFLAGS,
316     _CS_XBS5_LP64_OFF64_CFLAGS,    "XBS5_LP64_OFF64_CFLAGS",
317     CONFSTR, NOFLAGS,
318     _CS_XBS5_LP64_OFF64_CFLAGS,    "_XBS5_LP64_OFF64_CFLAGS",
319     CONFSTR, NOFLAGS,
321     _CS_XBS5_LP64_OFF64_LDFLAGS,   "XBS5_LP64_OFF64_LDFLAGS",
322     CONFSTR, NOFLAGS,
323     _CS_XBS5_LP64_OFF64_LDFLAGS,   "_XBS5_LP64_OFF64_LDFLAGS",
324     CONFSTR, NOFLAGS,
326     _CS_XBS5_LP64_OFF64_LIBS,      "XBS5_LP64_OFF64_LIBS",
327     CONFSTR, NOFLAGS,
328     _CS_XBS5_LP64_OFF64_LIBS,      "_XBS5_LP64_OFF64_LIBS",
329     CONFSTR, NOFLAGS,
331     _CS_XBS5_LP64_OFF64_LINTFLAGS, "XBS5_LP64_OFF64_LINTFLAGS",
332     CONFSTR, NOFLAGS,
333     _CS_XBS5_LP64_OFF64_LINTFLAGS, "_XBS5_LP64_OFF64_LINTFLAGS",
334     CONFSTR, NOFLAGS,
336     _CS_XBS5_LPBIG_OFFBIG_CFLAGS,  "XBS5_LPBIG_OFFBIG_CFLAGS",
337     CONFSTR, NOFLAGS,
338     _CS_XBS5_LPBIG_OFFBIG_CFLAGS,  "_XBS5_LPBIG_OFFBIG_CFLAGS",
339     CONFSTR, NOFLAGS,
341     _CS_XBS5_LPBIG_OFFBIG_LDFLAGS, "XBS5_LPBIG_OFFBIG_LDFLAGS",
342     CONFSTR, NOFLAGS,
343     _CS_XBS5_LPBIG_OFFBIG_LDFLAGS, "_XBS5_LPBIG_OFFBIG_LDFLAGS",
344     CONFSTR, NOFLAGS,

```

```

346     _CS_XBS5_LPBIG_OFFBIG_LIBS,      "XBS5_LPBIG_OFFBIG_LIBS",
347         CONFSTR,      NOFLAGS,
348     _CS_XBS5_LPBIG_OFFBIG_LIBS,      "_XBS5_LPBIG_OFFBIG_LIBS",
349         CONFSTR,      NOFLAGS,
350
351     _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "XBS5_LPBIG_OFFBIG_LINTFLAGS",
352         CONFSTR,      NOFLAGS,
353     _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "_XBS5_LPBIG_OFFBIG_LINTFLAGS",
354         CONFSTR,      NOFLAGS,
355
356 /*
357  * UNIX 03 confstr() additions for compilation environments
358  */
359     _CS_POSIX_V6_ILP32_OFF32_CFLAGS, "POSIX_V6_ILP32_OFF32_CFLAGS",
360         CONFSTR,      NOFLAGS,
361     _CS_POSIX_V6_ILP32_OFF32_LDFLAGS, "POSIX_V6_ILP32_OFF32_LDFLAGS",
362         CONFSTR,      NOFLAGS,
363     _CS_POSIX_V6_ILP32_OFF32_LIBS,   "POSIX_V6_ILP32_OFF32_LIBS",
364         CONFSTR,      NOFLAGS,
365     _CS_POSIX_V6_ILP32_OFFBIG_CFLAGS, "POSIX_V6_ILP32_OFFBIG_CFLAGS",
366         CONFSTR,      NOFLAGS,
367     _CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS, "POSIX_V6_ILP32_OFFBIG_LDFLAGS",
368         CONFSTR,      NOFLAGS,
369     _CS_POSIX_V6_ILP32_OFFBIG_LIBS,  "POSIX_V6_ILP32_OFFBIG_LIBS",
370         CONFSTR,      NOFLAGS,
371     _CS_POSIX_V6_LP64_OFF64_CFLAGS,  "POSIX_V6_LP64_OFF64_CFLAGS",
372         CONFSTR,      NOFLAGS,
373     _CS_POSIX_V6_LP64_OFF64_LDFLAGS, "POSIX_V6_LP64_OFF64_LDFLAGS",
374         CONFSTR,      NOFLAGS,
375     _CS_POSIX_V6_LP64_OFF64_LIBS,    "POSIX_V6_LP64_OFF64_LIBS",
376         CONFSTR,      NOFLAGS,
377     _CS_POSIX_V6_LPBIG_OFFBIG_CFLAGS, "POSIX_V6_LPBIG_OFFBIG_CFLAGS",
378         CONFSTR,      NOFLAGS,
379     _CS_POSIX_V6_LPBIG_OFFBIG_LDFLAGS, "POSIX_V6_LPBIG_OFFBIG_LDFLAGS",
380         CONFSTR,      NOFLAGS,
381     _CS_POSIX_V6_LPBIG_OFFBIG_LIBS,  "POSIX_V6_LPBIG_OFFBIG_LIBS",
382         CONFSTR,      NOFLAGS,
383     _CS_POSIX_V6_WIDTH_RESTRICTED_ENVS, "POSIX_V6_WIDTH_RESTRICTED_ENVS",
384         CONFSTR,      NOFLAGS,
385 /* End of UNIX 03 compilation environments */
386
387 #ifdef _SC_2_C_VERSION      /* POSIX.2 table B-4 */
388     _SC_2_C_VERSION,      "_POSIX2_C_VERSION",      SYSCONF,      NOFLAGS,
389 #else
390     _POSIX2_C_VERSION,    "_POSIX2_C_VERSION",      SELFCONF,      NOFLAGS,
391 #endif
392
393 /*
394  * XPG4 support BEGINS
395  */
396
397 #ifndef CHARCLASS_NAME_MAX
398 #define CHARCLASS_NAME_MAX    -1
399 #endif
400 #ifndef CHAR_BIT
401 #define CHAR_BIT              -1
402 #endif
403 #ifndef CHAR_MAX
404 #define CHAR_MAX              -1
405 #endif
406 #ifndef CHAR_MIN
407 #define CHAR_MIN              -1
408 #endif
409 #ifndef INT_MAX
410 #define INT_MAX               -1
411 #endif

```

```

412 #ifndef INT_MIN
413 #define INT_MIN               -1
414 #endif
415 #ifndef LONG_BIT
416 #define LONG_BIT              -1
417 #endif
418 #ifndef LONG_MAX
419 #define LONG_MAX              -1
420 #endif
421 #ifndef LONG_MIN
422 #define LONG_MIN              -1
423 #endif
424 #ifndef MB_LEN_MAX
425 #define MB_LEN_MAX            -1
426 #endif
427 #ifndef NL_NMAX
428 #define NL_NMAX                -1
429 #endif
430 #ifndef NL_ARGMAX
431 #define NL_ARGMAX              -1
432 #endif
433 #ifndef NL_LANGMAX
434 #define NL_LANGMAX            -1
435 #endif
436 #ifndef NL_MSGMAX
437 #define NL_MSGMAX              -1
438 #endif
439 #ifndef NL_SETMAX
440 #define NL_SETMAX              -1
441 #endif
442 #ifndef NL_TEXTMAX
443 #define NL_TEXTMAX            -1
444 #endif
445 #ifndef NZERO
446 #define NZERO                  -1
447 #endif
448 #ifndef SCHAR_MAX
449 #define SCHAR_MAX              -1
450 #endif
451 #ifndef SCHAR_MIN
452 #define SCHAR_MIN              -1
453 #endif
454 #ifndef SHRT_MAX
455 #define SHRT_MAX               -1
456 #endif
457 #ifndef SHRT_MIN
458 #define SHRT_MIN               -1
459 #endif
460 #ifndef TMP_MAX
461 #define TMP_MAX                 -1
462 #endif
463 #ifndef WORD_BIT
464 #define WORD_BIT                -1
465 #endif
466
467 #ifndef _XOPEN_XPG2
468 #define _XOPEN_XPG2            -1
469 #endif
470 #ifndef _XOPEN_XPG3
471 #define _XOPEN_XPG3            -1
472 #endif
473 #ifndef _XOPEN_XPG4
474 #define _XOPEN_XPG4            -1
475 #endif
476 #endif

```

```

478 /*
479  * the following are values that we should find in <limits.h>
480  * so we use SELFCNF here.
481  *
482  */

484 /* POSIX.1-1990 */
485 CHARCLASS_NAME_MAX, "CHARCLASS_NAME_MAX", SELFCNF, NOFLAGS,
486 SSIZE_MAX, "SSIZE_MAX", SELFCNF, NOFLAGS,

488 /* POSIX.1-1988 */
489 CHAR_BIT, "CHAR_BIT", SELFCNF, NOFLAGS,
490 CHAR_MAX, "CHAR_MAX", SELFCNF, NOFLAGS,
491 CHAR_MIN, "CHAR_MIN", SELFCNF, NOFLAGS,
492 INT_MAX, "INT_MAX", SELFCNF, NOFLAGS,
493 INT_MIN, "INT_MIN", SELFCNF, NOFLAGS,
494 LONG_BIT, "LONG_BIT", SELFCNF, NOFLAGS,
495 LONG_MAX, "LONG_MAX", SELFCNF, NOFLAGS,
496 LONG_MIN, "LONG_MIN", SELFCNF, NOFLAGS,
497 MB_LEN_MAX, "MB_LEN_MAX", SELFCNF, NOFLAGS,
498 NL_NMAX, "NL_NMAX", SELFCNF, NOFLAGS,
499 NL_ARGMAX, "NL_ARGMAX", SELFCNF, NOFLAGS,
500 NL_LANGMAX, "NL_LANGMAX", SELFCNF, NOFLAGS,
501 NL_MSGMAX, "NL_MSGMAX", SELFCNF, NOFLAGS,
502 NL_SETMAX, "NL_SETMAX", SELFCNF, NOFLAGS,
503 NL_TEXTMAX, "NL_TEXTMAX", SELFCNF, NOFLAGS,
504 NZERO, "NZERO", SELFCNF, NOFLAGS,
505 SCHAR_MAX, "SCHAR_MAX", SELFCNF, NOFLAGS,
506 SCHAR_MIN, "SCHAR_MIN", SELFCNF, NOFLAGS,
507 SHRT_MAX, "SHRT_MAX", SELFCNF, NOFLAGS,
508 SHRT_MIN, "SHRT_MIN", SELFCNF, NOFLAGS,
509 TMP_MAX, "TMP_MAX", SELFCNF, NOFLAGS,

511 /*
512  * for the unsigned maximums, we cannot rely on the value -1
513  * to indicate "undefined".
514  */
515 #ifndef UCHAR_MAX
516 0, "UCHAR_MAX", SELFCNF, UNDEFINED,
517 #else
518 (long)UCHAR_MAX, "UCHAR_MAX", SELFCNF, UNSIGNED_VALUE,
519 #endif /* UCHAR_MAX */
520 #ifndef UINT_MAX
521 0, "UINT_MAX", SELFCNF, UNDEFINED,
522 #else
523 (long)UINT_MAX, "UINT_MAX", SELFCNF, UNSIGNED_VALUE,
524 #endif /* UINT_MAX */
525 #ifndef ULONG_MAX
526 0, "ULONG_MAX", SELFCNF, UNDEFINED,
527 #else
528 (long)ULONG_MAX, "ULONG_MAX", SELFCNF, UNSIGNED_VALUE,
529 #endif /* ULONG_MAX */
530 #ifndef USHRT_MAX
531 0, "USHRT_MAX", SELFCNF, UNDEFINED,
532 #else
533 (long)USHRT_MAX, "USHRT_MAX", SELFCNF, UNSIGNED_VALUE,
534 #endif /* USHRT_MAX */
535 #ifndef SIZE_MAX
536 0, "SIZE_MAX", SELFCNF, UNDEFINED,
537 #else
538 (long)SIZE_MAX, "SIZE_MAX", SELFCNF, UNSIGNED_VALUE,
539 #endif /* SIZE_MAX */

541 WORD_BIT, "WORD_BIT", SELFCNF, NOFLAGS,

543 /* XPG4 Version 2 */

```

```

544 _XOPEN_IOV_MAX, "_XOPEN_IOV_MAX", SELFCNF, NOFLAGS,

546 /* UNIX 03 */
547 _XOPEN_NAME_MAX, "_XOPEN_NAME_MAX", SELFCNF, NOFLAGS,
548 _XOPEN_PATH_MAX, "_XOPEN_PATH_MAX", SELFCNF, NOFLAGS,

550 /*
551  * The following are defined via sysconf(). These are considered
552  * an extension to sysconf().
553  * The -0 is needed if _XOPEN_XPG? is defined to be an empty string.
554  */
555 _XOPEN_XPG2-0, "_XOPEN_XPG2", SELFCNF, NOFLAGS,
556 _XOPEN_XPG3-0, "_XOPEN_XPG3", SELFCNF, NOFLAGS,
557 _XOPEN_XPG4-0, "_XOPEN_XPG4", SELFCNF, NOFLAGS,

559 /*
560  * The following should be provided by sysconf() (e.g use SYSCONF),
561  * so we look for the appropriate _SC_* symbol in <unistd.h>.
562  * If it is not defined, then we use SELFCNF with the value of -1.
563  */
564 _SC_XOPEN_VERSION, "_XOPEN_VERSION", SYSCONF, NOFLAGS,
565 _SC_XOPEN_XCU_VERSION, "_XOPEN_XCU_VERSION", SYSCONF, NOFLAGS,
566 _SC_XOPEN_CRYPT, "_XOPEN_CRYPT", SYSCONF, NOFLAGS,
567 _SC_XOPEN_ENH_I18N, "_XOPEN_ENH_I18N", SYSCONF, NOFLAGS,
568 _SC_XOPEN_SHM, "_XOPEN_SHM", SYSCONF, NOFLAGS,

570 /*
571  * XPG4 support ends
572  */

574 /* XPG4, Version 2 support */

576 _SC_ATEXIT_MAX, "ATEXIT_MAX", SYSCONF, NOFLAGS,
577 _SC_PAGESIZE, "PAGESIZE", SYSCONF, NOFLAGS,
578 _SC_PAGE_SIZE, "PAGE_SIZE", SYSCONF, NOFLAGS,
579 _SC_XOPEN_UNIX, "_XOPEN_UNIX", SYSCONF, NOFLAGS,
580 _SC_IOV_MAX, "IOV_MAX", SYSCONF, NOFLAGS,

582 /*
583  * UNIX 98 - XPG5 additions
584  */
585 _SC_XOPEN_LEGACY, "_XOPEN_LEGACY", SYSCONF, NOFLAGS,
586 _SC_PASS_MAX, "PASS_MAX", SYSCONF, NOFLAGS,
587 _SC_AIO_LISTIO_MAX, "AIO_LISTIO_MAX", SYSCONF, NOFLAGS,
588 _SC_AIO_MAX, "AIO_MAX", SYSCONF, NOFLAGS,
589 _SC_AIO_PRIO_DELTA_MAX, "AIO_PRIO_DELTA_MAX", SYSCONF, NOFLAGS,
590 _SC_DELAYTIMER_MAX, "DELAYTIMER_MAX", SYSCONF, NOFLAGS,
591 _SC_MQ_OPEN_MAX, "MQ_OPEN_MAX", SYSCONF, NOFLAGS,
592 _SC_MQ_PRIO_MAX, "MQ_PRIO_MAX", SYSCONF, NOFLAGS,
593 _SC_RTSIG_MAX, "RTSIG_MAX", SYSCONF, NOFLAGS,
594 _SC_SEM_NSEMS_MAX, "SEM_NSEMS_MAX", SYSCONF, NOFLAGS,
595 _SC_SEM_VALUE_MAX, "SEM_VALUE_MAX", SYSCONF, NOFLAGS,
596 _SC_SIGQUEUE_MAX, "SIGQUEUE_MAX", SYSCONF, NOFLAGS,
597 _SC_TIMER_MAX, "TIMER_MAX", SYSCONF, NOFLAGS,

599 _SC_ASYNCHRONOUS_IO, "_POSIX_ASYNCHRONOUS_IO", SYSCONF, NOFLAGS,
600 _SC_FSYNC, "_POSIX_FSYNC", SYSCONF, NOFLAGS,
601 _SC_MAPPED_FILES, "_POSIX_MAPPED_FILES", SYSCONF, NOFLAGS,
602 _SC_MEMLOCK, "_POSIX_MEMLOCK", SYSCONF, NOFLAGS,
603 _SC_MEMLOCK_RANGE, "_POSIX_MEMLOCK_RANGE", SYSCONF, NOFLAGS,
604 _SC_MEMORY_PROTECTION, "_POSIX_MEMORY_PROTECTION", SYSCONF, NOFLAGS,
605 _SC_MESSAGE_PASSING, "_POSIX_MESSAGE_PASSING", SYSCONF, NOFLAGS,
606 _SC_PRIORITY_IO, "_POSIX_PRIORITY_IO", SYSCONF, NOFLAGS,
607 _SC_PRIORITY_SCHEDULING, "_POSIX_PRIORITY_SCHEDULING", SYSCONF, NOFLAGS,
608 _SC_REALTIME_SIGNALS, "_POSIX_REALTIME_SIGNALS", SYSCONF, NOFLAGS,
609 _SC_SEMAPHORES, "_POSIX_SEMAPHORES", SYSCONF, NOFLAGS,

```

```

611  _SC_SHARED_MEMORY_OBJECTS, "_POSIX_SHARED_MEMORY_OBJECTS",
612      SYSCONF, NOFLAGS,

614  _SC_SYNCHRONIZED_IO,      "_POSIX_SYNCHRONIZED_IO", SYSCONF, NOFLAGS,
615  _SC_TIMERS,                "_POSIX_TIMERS",          SYSCONF, NOFLAGS,

617  _SC_LOGIN_NAME_MAX,       "LOGIN_NAME_MAX",          SYSCONF, NOFLAGS,

619  _SC_THREAD_DESTRUCTOR_ITERATIONS, "PTHREAD_DESTRUCTOR_ITERATIONS",
620      SYSCONF, NOFLAGS,

622  _SC_THREAD_KEYS_MAX,      "PTHREAD_KEYS_MAX",       SYSCONF, NOFLAGS,
623  _SC_THREAD_STACK_MIN,     "PTHREAD_STACK_MIN",      SYSCONF, NOFLAGS,
624  _SC_THREAD_THREADS_MAX,   "PTHREAD_THREADS_MAX",    SYSCONF, NOFLAGS,

626  _SC_THREADS,              "_POSIX_THREADS",         SYSCONF, NOFLAGS,

628  _SC_THREAD_ATTR_STACKADDR, "POSIX_THREAD_ATTR_STACKADDR",
629      SYSCONF, NOFLAGS,
630  _SC_THREAD_ATTR_STACKADDR, "_POSIX_THREAD_ATTR_STACKADDR",
631      SYSCONF, NOFLAGS,

633  _SC_THREAD_ATTR_STACKSIZE, "POSIX_THREAD_ATTR_STACKSIZE",
634      SYSCONF, NOFLAGS,
635  _SC_THREAD_ATTR_STACKSIZE, "_POSIX_THREAD_ATTR_STACKSIZE",
636      SYSCONF, NOFLAGS,

638  _SC_THREAD_PRIORITY_SCHEDULING, "POSIX_THREAD_PRIORITY_SCHEDULING",
639      SYSCONF, NOFLAGS,
640  _SC_THREAD_PRIORITY_SCHEDULING, "_POSIX_THREAD_PRIORITY_SCHEDULING",
641      SYSCONF, NOFLAGS,

643  _SC_THREAD_PRIO_INHERIT, "POSIX_THREAD_PRIO_INHERIT",
644      SYSCONF, NOFLAGS,
645  _SC_THREAD_PRIO_INHERIT, "_POSIX_THREAD_PRIO_INHERIT",
646      SYSCONF, NOFLAGS,

648  _SC_THREAD_PRIO_PROTECT, "POSIX_THREAD_PRIO_PROTECT",
649      SYSCONF, NOFLAGS,
650  _SC_THREAD_PRIO_PROTECT, "_POSIX_THREAD_PRIO_PROTECT",
651      SYSCONF, NOFLAGS,

653  _SC_THREAD_PROCESS_SHARED, "POSIX_THREAD_PROCESS_SHARED",
654      SYSCONF, NOFLAGS,
655  _SC_THREAD_PROCESS_SHARED, "_POSIX_THREAD_PROCESS_SHARED",
656      SYSCONF, NOFLAGS,

658  _SC_THREAD_SAFE_FUNCTIONS, "POSIX_THREAD_SAFE_FUNCTIONS",
659      SYSCONF, NOFLAGS,
660  _SC_THREAD_SAFE_FUNCTIONS, "_POSIX_THREAD_SAFE_FUNCTIONS",
661      SYSCONF, NOFLAGS,

663  _SC_TTY_NAME_MAX,         "TTY_NAME_MAX",          SYSCONF, NOFLAGS,
664  _SC_XOPEN_REALTIME,      "_XOPEN_REALTIME",       SYSCONF, NOFLAGS,
665  _SC_XOPEN_REALTIME_THREADS, "_XOPEN_REALTIME_THREADS", SYSCONF, NOFLAGS,

667  /*
668  * Due to ambiguity in the standard we need
669  * to recognize both the underscored name
670  * and the non-underscored version in the
671  * following variables.
672  */
673  _SC_XBS5_ILP32_OFF32,     "XBS5_ILP32_OFF32",     SYSCONF, NOFLAGS,
674  _SC_XBS5_ILP32_OFF32,     "_XBS5_ILP32_OFF32",     SYSCONF, NOFLAGS,

```

```

676  _SC_XBS5_ILP32_OFFBIG,    "XBS5_ILP32_OFFBIG",    SYSCONF, NOFLAGS,
677  _SC_XBS5_ILP32_OFFBIG,    "_XBS5_ILP32_OFFBIG",    SYSCONF, NOFLAGS,

679  _SC_XBS5_LP64_OFF64,     "XBS5_LP64_OFF64",     SYSCONF, NOFLAGS,
680  _SC_XBS5_LP64_OFF64,     "_XBS5_LP64_OFF64",     SYSCONF, NOFLAGS,

682  _SC_XBS5_LPBIG_OFFBIG,   "XBS5_LPBIG_OFFBIG",   SYSCONF, NOFLAGS,
683  _SC_XBS5_LPBIG_OFFBIG,   "_XBS5_LPBIG_OFFBIG",   SYSCONF, NOFLAGS,

685  /*
686  * UNIX 03 additions for sysconf() names, and names that are not in interfaces
687  */
688  _SC_REGEXP,               "_POSIX_REGEXP",        SYSCONF, NOFLAGS,
689  _SC_SHELL,                "_POSIX_SHELL",         SYSCONF, NOFLAGS,
690  _SC_HOST_NAME_MAX,        "HOST_NAME_MAX",        SYSCONF, NOFLAGS,
691  _SC_READER_WRITER_LOCKS, "_POSIX_READER_WRITER_LOCKS", SYSCONF, NOFLAGS,
692  _SC_IPV6,                 "_POSIX_IPV6",          SYSCONF, NOFLAGS,
693  _SC_RAW_SOCKETS,         "_POSIX_RAW_SOCKETS",   SYSCONF, NOFLAGS,
694  _SC_XOPEN_STREAMS,        "XOPEN_STREAMS",        SYSCONF, NOFLAGS,
695  _SC_SYMLINK_MAX,          "SYMLINK_MAX",          SYSCONF, NOFLAGS,
696  _SC_2_PBS,                "_POSIX2_PBS",          SYSCONF, NOFLAGS,
697  _SC_2_PBS_ACCOUNTING,     "_POSIX2_PBS_ACCOUNTING", SYSCONF, NOFLAGS,
698  _SC_2_PBS_CHECKPOINT,    "_POSIX2_PBS_CHECKPOINT", SYSCONF, NOFLAGS,
699  _SC_2_PBS_LOCATE,         "_POSIX2_PBS_LOCATE",   SYSCONF, NOFLAGS,
700  _SC_2_PBS_MESSAGE,        "_POSIX2_PBS_MESSAGE",   SYSCONF, NOFLAGS,
701  _SC_2_PBS_TRACK,          "_POSIX2_PBS_TRACK",    SYSCONF, NOFLAGS,

703  _SC_ADVISORY_INFO,        "_POSIX_ADVISORY_INFO", SYSCONF, NOFLAGS,
704  _SC_BARRIERS,             "POSIX_BARRIERS",       SYSCONF, NOFLAGS,
705  _SC_CLOCK_SELECTION,      "_POSIX_CLOCK_SELECTION", SYSCONF, NOFLAGS,
706  _SC_CPUTIME,              "POSIX_CPUTIME",        SYSCONF, NOFLAGS,
707  _SC_MONOTONIC_CLOCK,      "_POSIX_MONOTONIC_CLOCK", SYSCONF, NOFLAGS,
708  _SC_SPAWN,                "POSIX_SPAWN",          SYSCONF, NOFLAGS,
709  _SC_SPIN_LOCKS,          "POSIX_SPIN_LOCKS",     SYSCONF, NOFLAGS,
710  _SC_SPORADIC_SERVER,      "POSIX_SPORADIC_SERVER", SYSCONF, NOFLAGS,
711  _SC_SS_REPL_MAX,          "POSIX_SS_REPL_MAX",    SYSCONF, NOFLAGS,
712  _SC_THREAD_CPUTIME,        "POSIX_THREAD_CPUTIME",  SYSCONF, NOFLAGS,
713  _SC_THREAD_SPARADIC_SERVER, "_POSIX_THREAD_SPARADIC_SERVER", SYSCONF, NOFLAGS,
714  _SC_TIMEOUTS,             "POSIX_TIMEOUTS",       SYSCONF, NOFLAGS,
715  _SC_TRACE,                "POSIX_TRACE",          SYSCONF, NOFLAGS,
716  _SC_TRACE_EVENT_FILTER,   "POSIX_TRACE_EVENT_FILTER", SYSCONF, NOFLAGS,
717  _SC_TRACE_EVENT_NAME_MAX, "POSIX_TRACE_EVENT_NAME_MAX", SYSCONF, NOFLAGS,
718  _SC_TRACE_EVENT_NAME_MAX, "POSIX_TRACE_EVENT_NAME_MAX", SYSCONF, NOFLAGS,
719  _SC_TRACE_INHERIT,        "POSIX_TRACE_INHERIT",  SYSCONF, NOFLAGS,
720  _SC_TRACE_LOG,            "POSIX_TRACE_LOG",       SYSCONF, NOFLAGS,
721  _SC_TRACE_NAME_MAX,       "POSIX_TRACE_NAME_MAX",  SYSCONF, NOFLAGS,
722  _SC_TRACE_SYS_MAX,        "POSIX_TRACE_SYS_MAX",   SYSCONF, NOFLAGS,
723  _SC_TRACE_USER_EVENT_MAX, "POSIX_TRACE_USER_EVENT_MAX", SYSCONF, NOFLAGS,
724  _SC_TYPED_MEMORY_OBJECTS, "_POSIX_TYPED_MEMORY_OBJECTS", SYSCONF, NOFLAGS,
725  _SC_TYPED_MEMORY_OBJECTS, "POSIX_TYPED_MEMORY_OBJECTS", SYSCONF, NOFLAGS,
726  _SC_TYPED_MEMORY_OBJECTS, "_POSIX_TYPED_MEMORY_OBJECTS", SYSCONF, NOFLAGS,
727  _SC_TYPED_MEMORY_OBJECTS, "POSIX_TYPED_MEMORY_OBJECTS", SYSCONF, NOFLAGS,

729  /*
730  * Due to an ambiguity in the specification, we need to recognize both
731  * the underscored POSIX name and the non-underscored name. In addition,
732  * we need to permit the use of the _V6_* environment names specified
733  * in unistd.h.
734  */
735  _SC_V6_ILP32_OFF32,       "_V6_ILP32_OFF32",      SYSCONF, NOFLAGS,
736  _SC_V6_ILP32_OFF32,       "_POSIX_V6_ILP32_OFF32", SYSCONF, NOFLAGS,
737  _SC_V6_ILP32_OFF32,       "POSIX_V6_ILP32_OFF32", SYSCONF, NOFLAGS,

739  _SC_V6_ILP32_OFFBIG,      "_V6_ILP32_OFFBIG",     SYSCONF, NOFLAGS,
740  _SC_V6_ILP32_OFFBIG,      "_POSIX_V6_ILP32_OFFBIG", SYSCONF, NOFLAGS,
741  _SC_V6_ILP32_OFFBIG,      "POSIX_V6_ILP32_OFFBIG", SYSCONF, NOFLAGS,

```

```

743     _SC_V6_LP64_OFF64,      "_V6_LP64_OFF64",      SYSCONF,      NOFLAGS,
744     _SC_V6_LP64_OFF64,      "_POSIX_V6_LP64_OFF64", SYSCONF,      NOFLAGS,
745     _SC_V6_LP64_OFF64,      "POSIX_V6_LP64_OFF64", SYSCONF,      NOFLAGS,

747     _SC_V6_LPBIG_OFFBIG,    "_V6_LPBIG_OFFBIG",    SYSCONF,      NOFLAGS,
748     _SC_V6_LPBIG_OFFBIG,    "_POSIX_V6_LPBIG_OFFBIG", SYSCONF,      NOFLAGS,
749     _SC_V6_LPBIG_OFFBIG,    "POSIX_V6_LPBIG_OFFBIG", SYSCONF,      NOFLAGS,
750 /*
751  * System V Interface Definition
752  */
753     _SC_NPROCESSORS_CONF,    "NPROCESSORS_CONF",    SYSCONF,      NOFLAGS,
754     _SC_NPROCESSORS_MAX,     "NPROCESSORS_MAX",     SYSCONF,      NOFLAGS,
755     _SC_NPROCESSORS_ONLN,    "NPROCESSORS_ONLN",    SYSCONF,      NOFLAGS,
756 #endif /* ! codereview */
757     0,                        NULL,      0,          0          /* end of table */
758 };

761 /*
762  * Print usage message.
763  */
764 static int
765 usage()
766 {
768     (void) fputs(gettext(
769         "usage:\tgetconf [-v specification] system_var\n"
770         "\tgetconf [-v specification] path_var pathname\n"
771         "\tgetconf -a\n"), stderr);
772     return (2);
773 }

776 static int
777 namecmp(const void *a, const void *b)
778 {
779     return (strcoll(((const struct sctab *)a)->name,
780         ((const struct sctab *)b)->name));
781 }

784 static int
785 getconf(struct sctab *scp, int argc, char *name, char *file)
786 {
787     register size_t len;
788     register char *buffer;
789     long value;

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

806     case SYSCONF:
807         if (argc > 2)

```

```

808             return (usage());
809         errno = 0;
810         if (scp->flag & UNDEFINED ||
811             (value = sysconf((int)scp->value)) == -1) {
812             if (errno == EINVAL) {
813                 (void) fprintf(stderr,
814                     gettext(INVAL_ARG), name);
815                 return (1);
816             } else {
817                 /* DO NOT TRANSLATE */
818                 (void) printf("undefined\n");
819                 return (0);
820             }
821         } else
822             (void) printf("%ld\n", value);
823         break;

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

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

851         (void) printf("%s\n", buffer);
852         free(buffer);
853         break;

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

```

```

874     }
875     (void) printf("%ld\n", value);
876     break;
877 }
878 return (0);
879 }

881 int
882 main(int argc, char **argv)
883 {
884     register struct sctab *scp;
885     int c;
886     int exstat = 0;

888     (void) setlocale(LC_ALL, "");
889 #if !defined(TEXT_DOMAIN) /* Should be defined by cc -D */
890 #define TEXT_DOMAIN "SYS_TEST"
891 #endif
892     (void) textdomain(TEXT_DOMAIN);

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

920     if ((aflag && argc >= 2) || (!aflag && argc < 2))
921         return (usage());
922     if (aflag) {

924 #define TabStop      8
925 #define RightColumn  32
926 #define DefPathName  "."

928     /*
929      * sort the table by the "name" field
930      * so we print it in sorted order
931      */
932     qsort(&sctab[0], (sizeof (sctab)/sizeof (struct sctab))-1,
933         sizeof (struct sctab), namecmp);

935     /*
936      * print all the known symbols and their values
937      */
938     for (scp = &sctab[0]; scp->name != NULL; ++scp) {
939         int stat;

```

```

941         /*
942          * create 2 columns:
943          * variable name in the left column,
944          * value in the right column.
945          * The right column starts at a tab stop.
946          */
947         (void) printf("%s:\t", scp->name);

949         c = strlen(scp->name) + 1;
950         c = (c+TabStop) / TabStop, c *= TabStop;
951         for (; c < RightColumn; c += TabStop)
952             (void) putchar('\t');

954         /*
955          * for pathconf() related variables that require
956          * a pathname, use "."
957          */
958         stat = getconf(scp, scp->type == PATHCONF ? 3 : 2,
959             scp->name, DefPathName);

961         exstat |= stat;

963         /*
964          * if stat != 0, then an error message was already
965          * printed in getconf(),
966          * so don't need to print one here
967          */
968     } else {

971         /*
972          * find the name of the specified variable (argv[1])
973          * and print its value.
974          */
975         for (scp = &sctab[0]; scp->name != NULL; ++scp)
976             if (strcmp(argv[1], scp->name) == 0) {
977                 exstat = getconf(scp, argc, argv[1], argv[2]);
978                 break;
979             }

981         /*
982          * if at last entry in table, then the user specified
983          * variable is invalid
984          */
985         if (scp->name == NULL) {
986             errno = EINVAL;
987             (void) fprintf(stderr, gettext(INVAL_ARG), argv[1]);
988             return (1);
989         }
990     }
991     return (exstat);
992 }

```

```

*****
17462 Wed Jan 16 16:29:30 2013
new/usr/src/man/man1/getconf.1
452 /usr/!(/xpg4/|xpg6/)bin/getconf need to support NPROCESSORS_*
Reviewed by: Garrett D'Amore <garrett@damore.org>
*****
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 \"Jan 16, 2013\"
14 .TH GETCONF 1 \"Sep 1, 2009\"
15 .SH NAME
16 getconf \- get configuration values
17 .SH SYNOPSIS
18 .SS \"/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf\"
19 .LP
20 .nf
21 \fB/usr/bin/getconf\fR [\fB-v\fR \fIspecification\fR] \fIsystem_var\fR
22 .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\fR\fR\fR\fR\fR\fR\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\fR\fR\fR\fR
187 .ad
188 .RS 24n
189 Display variables with \fBcall\fR prefix that matches \fR\fR. The \fBcall\fR
190 prefixes are:
191 .RS +4

```

```

192 .TP
193 .ie t \(\bu
194 .el o
195 \fB\fB-CS\fR
196 .sp
197 \fB\fBconfstr\fR(3C),
198 .RE
199 .RS +4
200 .TP
201 .ie t \(\bu
202 .el o
203 \fB\fBPC\fR
204 .sp
205 \fB\fBpathconf\fR(2)
206 .RE
207 .RS +4
208 .TP
209 .ie t \(\bu
210 .el o
211 \fB\fBSC\fR
212 .sp
213 \fB\fBsysconf\fR(3C)
214 .RE
215 .RS +4
216 .TP
217 .ie t \(\bu
218 .el o
219 \fB\fBСИ\fR
220 .sp
221 \fB\fBsysinfo\fR(2)
222 .RE
223 .RS +4
224 .TP
225 .ie t \(\bu
226 .el o
227 \fB\fBXX\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\fiopath_var\fr\fr
398 .ad
399 .RS 12n
400 A name of a configuration variable whose value is available from the
401 \fb\fbpathconf\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          NAME_MAX          _POSIX_CHOWN_RESTRICTED
412 MAX_CANON         PATH_MAX          _POSIX_NO_TRUNC
413 MAX_INPUT         PIPE_BUF         _POSIX_VDISABLE
414 .TE

416 .sp
417 .ne 2
418 .na
419 \fb\fiopathname\fr\fr
420 .ad
421 .RS 14n
422 A path name for which the variable specified by \fiopath_var\fr is to be
423 determined.
424 .RE

426 .sp
427 .ne 2
428 .na
429 \fb\fisystem_var\fr\fr
430 .ad
431 .RS 14n
432 A name of a configuration variable whose value is available from
433 \fb\fbconfstr\fr(3C) or \fb\fbsysconf\fr(3C). All of the values in the following
434 table are supported:
435 .RE

437 .sp

439 .sp
440 .TS
441 1 1
442 1 1 .
443 ARG_MAX BC_BASE_MAX
444 BC_DIM_MAX BC_SCALE_MAX
445 BC_STRING_MAX CHAR_BIT
446 CHARCLASS_NAME_MAX CHAR_MAX
447 CHAR_MIN CHILD_MAX
448 CLK_TCK COLL_WEIGHTS_MAX
449 CS_PATH EXPR_NEST_MAX
450 HOST_NAME_MAX INT_MAX
451 INT_MIN LFS64_CFLAGS
452 LFS64_LDFLAGS LFS64_LIBS
453 LFS64_LINTFLAGS LFS_CFLAGS
454 LFS_LDFLAGS LFS_LIBS
455 LFS_LINTFLAGS LINE_MAX

```

```

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

```

```

522 _POSIX_THREAD_CPUTIME      _POSIX_THREAD_DESTRUCTOR_ITERATIONS
523 _POSIX_THREAD_KEYS_MAX    _POSIX_THREAD_PRIORITY_SCHEDULING
524 _POSIX_THREAD_PRIO_INHERIT  _POSIX_THREAD_PRIO_PROTECT
525 _POSIX_THREAD_PROCESS_SHARED  _POSIX_THREAD_SAFE_FUNCTIONS
526 _POSIX_THREAD_SPORADIC_SERVER  _POSIX_THREAD_THREADS_MAX
527 _POSIX_TIMEOUTS           _POSIX_TIMER_MAX
528 _POSIX_TRA_POSIX_TIMER_MAXCE  _POSIX_TIMESTAMP_RESOLUTION
461 NL_TEXTMAX                NZERO
462 OPEN_MAX                  POSIX2_BC_BASE_MAX
463 POSIX2_BC_DIM_MAX         POSIX2_BC_SCALE_MAX
464 POSIX2_BC_STRING_MAX     POSIX2_C_BIND
465 POSIX2_C_DEV              POSIX2_CHAR_TERM
466 POSIX2_COLL_WEIGHTS_MAX  POSIX2_C_VERSION
467 POSIX2_EXPR_NEST_MAX     POSIX2_FORT_DEV
468 POSIX2_FORT_RUN          POSIX2_LINE_MAX
469 POSIX2_LOCALEDEF         POSIX2_RE_DUP_MAX
470 POSIX2_SW_DEV            POSIX2_SYMLINKS
471 POSIX2_UPE                POSIX2_VERSION
472 POSIX_ALLOC_SIZE_MIN     POSIX_REC_INCR_XFER_SIZE
473 POSIX_REC_MAX_XFER_SIZE  POSIX_REC_MIN_XFER_SIZE
474 POSIX_REC_XFER_ALIGN     POSIX_V6_ILP32_OFF32
475 POSIX_V6_ILP32_OFF32_CFLAGS  POSIX_V6_ILP32_OFF32_LDFLAGS
476 POSIX_V6_ILP32_OFF32_LIBS   POSIX_V6_ILP32_OFFBIG
477 POSIX_V6_ILP32_OFFBIG_CFLAGS  POSIX_V6_ILP32_OFFBIG_LDFLAGS
478 POSIX_V6_ILP32_OFFBIG_LIBS   POSIX_V6_LP64_OFF64
479 POSIX_V6_LP64_OFF64_CFLAGS  POSIX_V6_LP64_OFF64_LDFLAGS
480 POSIX_V6_LP64_OFF64_LIBS    POSIX_V6_LPBIG_OFFBIG
481 POSIX_V6_LPBIG_OFFBIG_CFLAGS  POSIX_V6_LPBIG_OFFBIG_LDFLAGS
482 POSIX_V6_LPBIG_OFFBIG_LIBS   POSIX_V6_WIDTH_RESTRICTED_ENVS
483 SYMLINK_MAX              SYMLOOP_MAX
484 _POSIX2_BC_BASE_MAX      _POSIX2_BC_DIM_MAX
485 _POSIX2_BC_SCALE_MAX     _POSIX2_BC_STRING_MAX
486 _POSIX2_CHARCLASS_NAME_MAX  _POSIX2_CHAR_TERM
487 _POSIX2_COLL_WEIGHTS_MAX  _POSIX2_C_BIND
488 _POSIX2_C_DEV            _POSIX2_C_VERSION
489 _POSIX2_EXPR_NEST_MAX    _POSIX2_FORT_DEV
490 _POSIX2_FORT_RUN         _POSIX2_LINE_MAX
491 _POSIX2_LOCALEDEF        _POSIX2_PBS
492 _POSIX2_PBS_ACCOUNTING   _POSIX2_PBS_CHECKPOINT
493 _POSIX2_PBS_LOCATE       _POSIX2_PBS_MESSAGE
494 _POSIX2_PBS_TRACK        _POSIX2_RE_DUP_MAX
495 _POSIX2_SW_DEV           _POSIX2_UPE
496 _POSIX2_VERSION          _POSIX2_ADVISORY_INFO
497 _POSIX_AIO_LISTIO_MAX    _POSIX_AIO_MAX
498 _POSIX_ARG_MAX           _POSIX_ASYNC_IO
499 _POSIX_BARRIERS          _POSIX_CHILD_MAX
500 _POSIX_CLOCKRES_MIN      _POSIX_CLOCK_SELECTION
501 _POSIX_CPUTIME           _POSIX_DELAYTIMER_MAX
502 _POSIX_HOST_NAME_MAX     _POSIX_IPV6
503 _POSIX_JOB_CONTROL       _POSIX_LINK_MAX
504 _POSIX_LOGIN_NAME_MAX    _POSIX_MAX_CANON
505 _POSIX_MAX_INPUT         _POSIX_MONOTONIC_CLOCK
506 _POSIX_MQ_OPEN_MAX       _POSIX_MQ_PRIO_MAX
507 _POSIX_NAME_MAX          _POSIX_NGROUPS_MAX
508 _POSIX_OPEN_MAX          _POSIX_PATH_MAX
509 _POSIX_PIPE_BUF          _POSIX_PRIO_IO
510 _POSIX_RAW_SOCKETS       _POSIX_READER_WRITER_LOCKS
511 _POSIX_REGEXP            _POSIX_RE_DUP_MAX
512 _POSIX_RTSIG_MAX         _POSIX_SAVED_IDS
513 _POSIX_SEM_NSEMS_MAX     _POSIX_SEM_VALUE_MAX
514 _POSIX_SHELL             _POSIX_SIGQUEUE_MAX
515 _POSIX_SPAWN             _POSIX_SPIN_LOCKS
516 _POSIX_SPORADIC_SERVER   _POSIX_SSIZE_MAX
517 _POSIX_SS_REPL_MAX       _POSIX_STREAM_MAX
518 _POSIX_SYMLINK_MAX       _POSIX_SYMLOOP_MAX
519 _POSIX_SYNC_IO           _POSIX_THREAD_ATTR_STACKADDR

```

```

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

564 .sp
565 .LP
566 The symbol \fBPATH\fR also is recognized, yielding the same value as the
567 \fBconfstr()\fR name value \fBCS_PATH\fR.
568 .SH USAGE
569 .sp
570 .LP
571 See \fBlargefile\fR(5) for the description of the behavior of
572 \fB/usr/bin/getconf\fR when encountering files greater than or equal to 2 Gbyte
573 ( 2^31 bytes).
574 .SH EXAMPLES
575 .LP
576 \fBExample 1 \fRWriting the Value of a Variable
577 .sp
578 .LP
579 This example illustrates the value of \fB{NGROUPS_MAX}:\fR

581 .sp
582 .in +2
583 .nf
584 example% \fBgetconf NGROUPS_MAX\fR
585 .fi
586 .in -2

```

```

587 .sp
589 .LP
590 \fBExample 2 \fRWriting the Value of a Variable for a Specific Directory
591 .sp
592 .LP
593 This example illustrates the value of \fBNAME_MAX\fR for a specific directory:
595 .sp
596 .in +2
597 .nf
598 example% \fBgetconf NAME_MAX /usr\fR
599 .fi
600 .in -2
601 .sp
603 .LP
604 \fBExample 3 \fRDealing with Unspecified Results
605 .sp
606 .LP
607 This example shows how to deal more carefully with results that might be
608 unspecified:
610 .sp
611 .in +2
612 .nf
613 \fBif value=$(getconf PATH_MAX /usr); then
614 if [ "$value" = "undefined" ]; then
615 echo PATH_MAX in /usr is infinite.
616 else
617 echo PATH_MAX in /usr is $value.
618 fi
619 else
620 echo Error in getconf.
621 fi\fR
622 .fi
623 .in -2
625 .sp
626 .LP
627 For example:
629 .sp
630 .in +2
631 .nf
632 \fBsysconf(_SC_POSIX_C_BIND);\fR
633 .fi
634 .in -2
636 .sp
637 .LP
638 and
640 .sp
641 .in +2
642 .nf
643 \fBsystem("getconf POSIX2_C_BIND);\fR
644 .fi
645 .in -2
647 .sp
648 .LP
649 in a C program could give different answers. The \fBsysconf\fR call supplies a
650 value that corresponds to the conditions when the program was either compiled
651 or executed, depending on the implementation. The \fBsystem\fR call to
652 \fBgetconf\fR always supplies a value corresponding to conditions when the

```

```

653 program is executed.
655 .SH ENVIRONMENT VARIABLES
656 .sp
657 .LP
658 See \fBenviron\fR(5) for descriptions of the following environment variables
659 that affect the execution of \fBgetconf\fR: \fBBLANG\fR, \fBBLC_ALL\fR,
660 \fBBLC_CTYPE\fR, \fBBLC_MESSAGES\fR, and \fBNLSPATH\fR.
661 .sp
662 .LP
663 The following environment variable is supported by the \fBksh93\fR built-in
664 \fBgetconf\fR command:
665 .sp
666 .ne 2
667 .na
668 \fB\fB_AST_FEATURES\fR\fR
669 .ad
670 .RS 17n
671 Local writable values that are different from the default are stored in the
672 \fB_AST_FEATURES\fR environment variable. The \fB_AST_FEATURES\fR value is a
673 space-separated list of \fBfname path value\fR 3-tuples, where \fBfname\fR is the
674 system configuration name, \fBfpath\fR is the corresponding path, \fBfB\fR if no
675 path is applicable, and \fBfvalue\fR is the system configuration value.
676 .RE
678 .SH EXIT STATUS
679 .sp
680 .LP
681 The following exit values are returned:
682 .sp
683 .ne 2
684 .na
685 \fB\fB0\fR\fR
686 .ad
687 .RS 6n
688 The specified variable is valid and information about its current state was
689 written successfully.
690 .RE
692 .sp
693 .ne 2
694 .na
695 \fB\fB>0\fR\fR
696 .ad
697 .RS 6n
698 An error occurred.
699 .RE
701 .SH ATTRIBUTES
702 .sp
703 .LP
704 See \fBattributes\fR(5) for descriptions of the following attributes:
705 .SS "/usr/bin/getconf, /usr/xpg4/bin/getconf, /usr/xpg6/bin/getconf"
706 .sp
708 .sp
709 .TS
710 box;
711 c | c
712 l | l .
713 ATTRIBUTE TYPE ATTRIBUTE VALUE
714 _
715 Interface Stability Committed
716 _
717 Standard See \fBstandards\fR(5).
718 .TE

```

```
720 .SS "ksh93"
721 .sp

723 .sp
724 .TS
725 box;
726 c | c
727 l | l .
728 ATTRIBUTE TYPE  ATTRIBUTE VALUE
729 -
730 Interface Stability    See below.
731 .TE

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