

```

*****
32718 Fri Sep 27 14:13:12 2013
new/usr/src/cmd/getconf/getconf.c
4146 getconf(1) does not support XATTR_ENABLED, XATTR_EXISTS, SATTR_ENABLED, SAT
Reviewed by: Cedric Blancher <cedric.blancher@gmail.com>
*****
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, "POSIX_THREAD_DESTRUCTOR_ITERATIONS", SELFCNF, NOFLAGS,
182 _POSIX_THREAD_DESTRUCTOR_ITERATIONS, "POSIX_THREAD_DESTRUCTOR_ITERATIONS", SELFCNF, NOFLAGS,
183 _POSIX_THREAD_KEYS_MAX, "POSIX_THREAD_KEYS_MAX", SELFCNF, NOFLAGS,
184 _POSIX_THREAD_KEYS_MAX, "POSIX_THREAD_KEYS_MAX", SELFCNF, NOFLAGS,
185 _POSIX_THREAD_THREADS_MAX, "POSIX_THREAD_THREADS_MAX", SELFCNF, NOFLAGS,
186 _POSIX_THREAD_THREADS_MAX, "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_ACL_ENABLED,     "ACL_ENABLED",       PATHCONF, NOFLAGS,
241     _PC_MIN_HOLE_SIZE,   "MIN_HOLE_SIZE",     PATHCONF, NOFLAGS,
242     _PC_SATTR_ENABLED,   "SATTR_ENABLED",     PATHCONF, NOFLAGS,
243     _PC_SATTR_EXISTS,    "SATTR_EXISTS",     PATHCONF, NOFLAGS,
245 #endif /* ! codereview */
246     /* UNIX 08 additions */
248     _PC_TIMESTAMP_RESOLUTION, "POSIX_TIMESTAMP_RESOLUTION",
249     PATHCONF, NOFLAGS,
251     /* confstr() variables */
253     /* POSIX.2-1992 description of getconf utility */
254     _CS_PATH,            "PATH",              CONFSTR, NOFLAGS,
256     /* POSIX.2-1992 table B-18 */
257     _CS_PATH,            "CS_PATH",           CONFSTR, NOFLAGS,
258     _CS_PATH,            "_CS_PATH",           CONFSTR, NOFLAGS,
260     /* command names for large file configuration information */
261     /* large file compilation environment configuration */
263     _CS_LFS_CFLAGS,      "LFS_CFLAGS",        CONFSTR, NOFLAGS,
264     _CS_LFS_LDFLAGS,     "LFS_LDFLAGS",       CONFSTR, NOFLAGS,
265     _CS_LFS_LIBS,        "LFS_LIBS",          CONFSTR, NOFLAGS,
266     _CS_LFS_LINTFLAGS,   "LFS_LINTFLAGS",     CONFSTR, NOFLAGS,
268     /* transitional large file interface configuration */
270     _CS_LFS64_CFLAGS,    "LFS64_CFLAGS",      CONFSTR, NOFLAGS,
271     _CS_LFS64_LDFLAGS,   "LFS64_LDFLAGS",     CONFSTR, NOFLAGS,
272     _CS_LFS64_LIBS,      "LFS64_LIBS",        CONFSTR, NOFLAGS,
273     _CS_LFS64_LINTFLAGS, "LFS64_LINTFLAGS",   CONFSTR, NOFLAGS,
275     /*
276     * UNIX 98:
277     * confstr additions, these support
278     * programming specifications/environments.
279     */

```

```

280     * Unfortunately there is some uncertainty with
281     * the standard at this point about the constant
282     * name, thus we need to recognize BOTH with and
283     * without the initial underscore.
284     */
285     _CS_XBS5_ILP32_OFF32_CFLAGS, "XBS5_ILP32_OFF32_CFLAGS",
286     CONFSTR, NOFLAGS,
287     _CS_XBS5_ILP32_OFF32_CFLAGS, "_XBS5_ILP32_OFF32_CFLAGS",
288     CONFSTR, NOFLAGS,
290     _CS_XBS5_ILP32_OFF32_LDFLAGS, "XBS5_ILP32_OFF32_LDFLAGS",
291     CONFSTR, NOFLAGS,
292     _CS_XBS5_ILP32_OFF32_LDFLAGS, "_XBS5_ILP32_OFF32_LDFLAGS",
293     CONFSTR, NOFLAGS,
295     _CS_XBS5_ILP32_OFF32_LIBS,    "XBS5_ILP32_OFF32_LIBS",
296     CONFSTR, NOFLAGS,
297     _CS_XBS5_ILP32_OFF32_LIBS,    "_XBS5_ILP32_OFF32_LIBS",
298     CONFSTR, NOFLAGS,
300     _CS_XBS5_ILP32_OFF32_LINTFLAGS, "XBS5_ILP32_OFF32_LINTFLAGS",
301     CONFSTR, NOFLAGS,
302     _CS_XBS5_ILP32_OFF32_LINTFLAGS, "_XBS5_ILP32_OFF32_LINTFLAGS",
303     CONFSTR, NOFLAGS,
305     _CS_XBS5_ILP32_OFFBIG_CFLAGS, "XBS5_ILP32_OFFBIG_CFLAGS",
306     CONFSTR, NOFLAGS,
307     _CS_XBS5_ILP32_OFFBIG_CFLAGS, "_XBS5_ILP32_OFFBIG_CFLAGS",
308     CONFSTR, NOFLAGS,
310     _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "XBS5_ILP32_OFFBIG_LDFLAGS",
311     CONFSTR, NOFLAGS,
312     _CS_XBS5_ILP32_OFFBIG_LDFLAGS, "_XBS5_ILP32_OFFBIG_LDFLAGS",
313     CONFSTR, NOFLAGS,
315     _CS_XBS5_ILP32_OFFBIG_LIBS,    "XBS5_ILP32_OFFBIG_LIBS",
316     CONFSTR, NOFLAGS,
317     _CS_XBS5_ILP32_OFFBIG_LIBS,    "_XBS5_ILP32_OFFBIG_LIBS",
318     CONFSTR, NOFLAGS,
320     _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "XBS5_ILP32_OFFBIG_LINTFLAGS",
321     CONFSTR, NOFLAGS,
322     _CS_XBS5_ILP32_OFFBIG_LINTFLAGS, "_XBS5_ILP32_OFFBIG_LINTFLAGS",
323     CONFSTR, NOFLAGS,
325     _CS_XBS5_LP64_OFF64_CFLAGS,    "XBS5_LP64_OFF64_CFLAGS",
326     CONFSTR, NOFLAGS,
327     _CS_XBS5_LP64_OFF64_CFLAGS,    "_XBS5_LP64_OFF64_CFLAGS",
328     CONFSTR, NOFLAGS,
330     _CS_XBS5_LP64_OFF64_LDFLAGS,    "XBS5_LP64_OFF64_LDFLAGS",
331     CONFSTR, NOFLAGS,
332     _CS_XBS5_LP64_OFF64_LDFLAGS,    "_XBS5_LP64_OFF64_LDFLAGS",
333     CONFSTR, NOFLAGS,
335     _CS_XBS5_LP64_OFF64_LIBS,        "XBS5_LP64_OFF64_LIBS",
336     CONFSTR, NOFLAGS,
337     _CS_XBS5_LP64_OFF64_LIBS,        "_XBS5_LP64_OFF64_LIBS",
338     CONFSTR, NOFLAGS,
340     _CS_XBS5_LP64_OFF64_LINTFLAGS,   "XBS5_LP64_OFF64_LINTFLAGS",
341     CONFSTR, NOFLAGS,
342     _CS_XBS5_LP64_OFF64_LINTFLAGS,   "_XBS5_LP64_OFF64_LINTFLAGS",
343     CONFSTR, NOFLAGS,
345     _CS_XBS5_LPBIG_OFFBIG_CFLAGS,    "XBS5_LPBIG_OFFBIG_CFLAGS",

```

```

346     CONFSTR,          NOFLAGS,
347     _CS_XBS5_LPBIG_OFFBIG_CFLAGS,  "_XBS5_LPBIG_OFFBIG_CFLAGS",
348     CONFSTR,          NOFLAGS,

350     _CS_XBS5_LPBIG_OFFBIG_LDFLAGS,  "XBS5_LPBIG_OFFBIG_LDFLAGS",
351     CONFSTR,          NOFLAGS,
352     _CS_XBS5_LPBIG_OFFBIG_LDFLAGS,  "_XBS5_LPBIG_OFFBIG_LDFLAGS",
353     CONFSTR,          NOFLAGS,

355     _CS_XBS5_LPBIG_OFFBIG_LIBS,     "XBS5_LPBIG_OFFBIG_LIBS",
356     CONFSTR,          NOFLAGS,
357     _CS_XBS5_LPBIG_OFFBIG_LIBS,     "_XBS5_LPBIG_OFFBIG_LIBS",
358     CONFSTR,          NOFLAGS,

360     _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "XBS5_LPBIG_OFFBIG_LINTFLAGS",
361     CONFSTR,          NOFLAGS,
362     _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS, "_XBS5_LPBIG_OFFBIG_LINTFLAGS",
363     CONFSTR,          NOFLAGS,

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

396 #ifndef _SC_2_C_VERSION      /* POSIX.2 table B-4 */
397     _SC_2_C_VERSION,        "_POSIX2_C_VERSION",    SYSCONF,    NOFLAGS,
398 #else
399     _POSIX2_C_VERSION,      "_POSIX2_C_VERSION",    SELFCONF,  NOFLAGS,
400 #endif

402 /*
403 * XPG4 support BEGINS
404 */

406 #ifndef CHARCLASS_NAME_MAX
407 #define CHARCLASS_NAME_MAX -1
408 #endif
409 #ifndef CHAR_BIT
410 #define CHAR_BIT -1
411 #endif

```

```

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

477 #ifndef _XOPEN_XPG2

```

```

478 #define _XOPEN_XPG2      -1
479 #endif
480 #ifndef _XOPEN_XPG3
481 #define _XOPEN_XPG3      -1
482 #endif
483 #ifndef _XOPEN_XPG4
484 #define _XOPEN_XPG4      -1
485 #endif

487 /*
488  * the following are values that we should find in <limits.h>
489  * so we use SELFCNF here.
490  *
491  */

493 /* POSIX.1-1990 */
494 CHARCLASS_NAME_MAX,    "CHARCLASS_NAME_MAX",    SELFCNF,    NOFLAGS,
495 SSIZE_MAX,             "SSIZE_MAX",             SELFCNF,    NOFLAGS,

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

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

```

```

544 #ifndef SIZE_MAX
545 0,                      "SIZE_MAX",              SELFCNF,    UNDEFINED,
546 #else
547 (long)SIZE_MAX,         "SIZE_MAX",              SELFCNF,    UNSIGNED_VALUE,
548 #endif /* SIZE_MAX */

550 WORD_BIT,               "WORD_BIT",              SELFCNF,    NOFLAGS,

552 /* XPG4 Version 2 */
553 _XOPEN_IOV_MAX,         "_XOPEN_IOV_MAX",        SELFCNF,    NOFLAGS,

555 /* UNIX 03 */
556 _XOPEN_NAME_MAX,        "_XOPEN_NAME_MAX",        SELFCNF,    NOFLAGS,
557 _XOPEN_PATH_MAX,        "_XOPEN_PATH_MAX",        SELFCNF,    NOFLAGS,

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

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

579 /*
580  * XPG4 support ends
581  */

583 /* XPG4, Version 2 support */

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

591 /*
592  * UNIX 98 - XPG5 additions
593  */
594 _SC_XOPEN_LEGACY,       "_XOPEN_LEGACY",          SYSCONF,    NOFLAGS,
595 _SC_PASS_MAX,           "PASS_MAX",               SYSCONF,    NOFLAGS,
596 _SC_AIO_LISTIO_MAX,     "AIO_LISTIO_MAX",        SYSCONF,    NOFLAGS,
597 _SC_AIO_MAX,            "AIO_MAX",                SYSCONF,    NOFLAGS,
598 _SC_AIO_Prio_DELTA_MAX, "AIO_Prio_DELTA_MAX",    SYSCONF,    NOFLAGS,
599 _SC_DELAYTIMER_MAX,     "DELAYTIMER_MAX",         SYSCONF,    NOFLAGS,
600 _SC_MQ_OPEN_MAX,        "MQ_OPEN_MAX",            SYSCONF,    NOFLAGS,
601 _SC_MQ_Prio_MAX,        "MQ_Prio_MAX",            SYSCONF,    NOFLAGS,
602 _SC_RTsig_MAX,          "RTsig_MAX",              SYSCONF,    NOFLAGS,
603 _SC_SEM_NSEMS_MAX,      "SEM_NSEMS_MAX",          SYSCONF,    NOFLAGS,
604 _SC_SEM_VALUE_MAX,      "SEM_VALUE_MAX",          SYSCONF,    NOFLAGS,
605 _SC_SIGQUEUE_MAX,       "SIGQUEUE_MAX",           SYSCONF,    NOFLAGS,
606 _SC_TIMER_MAX,          "TIMER_MAX",              SYSCONF,    NOFLAGS,

608 _SC_ASYNCHRONOUS_IO,    "_POSIX_ASYNCHRONOUS_IO", SYSCONF,    NOFLAGS,
609 _SC_FSYNC,              "_POSIX_FSYNC",           SYSCONF,    NOFLAGS,

```

```

610 _SC_MAPPED_FILES,      "_POSIX_MAPPED_FILES",  SYSCONF,  NOFLAGS,
611 _SC_MEMLOCK,          " _POSIX_MEMLOCK",     SYSCONF,  NOFLAGS,
612 _SC_MEMLOCK_RANGE,   " _POSIX_MEMLOCK_RANGE", SYSCONF,  NOFLAGS,
613 _SC_MEMORY_PROTECTION, " _POSIX_MEMORY_PROTECTION", SYSCONF, NOFLAGS,
614 _SC_MESSAGE_PASSING, " _POSIX_MESSAGE_PASSING", SYSCONF, NOFLAGS,
615 _SC_PRIORITIZED_IO,  " _POSIX_PRIORITIZED_IO", SYSCONF, NOFLAGS,
616 _SC_PRIORITY_SCHEDULING, " _POSIX_PRIORITY_SCHEDULING", SYSCONF, NOFLAGS,
617 _SC_REALTIME_SIGNALS, " _POSIX_REALTIME_SIGNALS", SYSCONF, NOFLAGS,
618 _SC_SEMAPHORES,      " _POSIX_SEMAPHORES",   SYSCONF,  NOFLAGS,

620 _SC_SHARED_MEMORY_OBJECTS, " _POSIX_SHARED_MEMORY_OBJECTS",
621     SYSCONF,  NOFLAGS,

623 _SC_SYNCHRONIZED_IO,  " _POSIX_SYNCHRONIZED_IO", SYSCONF,  NOFLAGS,
624 _SC_TIMERS,           " _POSIX_TIMERS",         SYSCONF,  NOFLAGS,

626 _SC_LOGIN_NAME_MAX,   "LOGIN_NAME_MAX",       SYSCONF,  NOFLAGS,

628 _SC_THREAD_DESTRUCTOR_ITERATIONS, "PTHREAD_DESTRUCTOR_ITERATIONS",
629     SYSCONF,  NOFLAGS,

631 _SC_THREAD_KEYS_MAX,  "PTHREAD_KEYS_MAX",     SYSCONF,  NOFLAGS,
632 _SC_THREAD_STACK_MIN, "PTHREAD_STACK_MIN",   SYSCONF,  NOFLAGS,
633 _SC_THREAD_THREADS_MAX, "PTHREAD_THREADS_MAX", SYSCONF,  NOFLAGS,

635 _SC_THREADS,          " _POSIX_THREADS",      SYSCONF,  NOFLAGS,

637 _SC_THREAD_ATTR_STACKADDR, "POSIX_THREAD_ATTR_STACKADDR",
638     SYSCONF,  NOFLAGS,
639 _SC_THREAD_ATTR_STACKADDR, " _POSIX_THREAD_ATTR_STACKADDR",
640     SYSCONF,  NOFLAGS,

642 _SC_THREAD_ATTR_STACKSIZE, "POSIX_THREAD_ATTR_STACKSIZE",
643     SYSCONF,  NOFLAGS,
644 _SC_THREAD_ATTR_STACKSIZE, " _POSIX_THREAD_ATTR_STACKSIZE",
645     SYSCONF,  NOFLAGS,

647 _SC_THREAD_PRIORITY_SCHEDULING, "POSIX_THREAD_PRIORITY_SCHEDULING",
648     SYSCONF,  NOFLAGS,
649 _SC_THREAD_PRIORITY_SCHEDULING, " _POSIX_THREAD_PRIORITY_SCHEDULING",
650     SYSCONF,  NOFLAGS,

652 _SC_THREAD_PRIO_INHERIT, "POSIX_THREAD_PRIO_INHERIT",
653     SYSCONF,  NOFLAGS,
654 _SC_THREAD_PRIO_INHERIT, " _POSIX_THREAD_PRIO_INHERIT",
655     SYSCONF,  NOFLAGS,

657 _SC_THREAD_PRIO_PROTECT, "POSIX_THREAD_PRIO_PROTECT",
658     SYSCONF,  NOFLAGS,
659 _SC_THREAD_PRIO_PROTECT, " _POSIX_THREAD_PRIO_PROTECT",
660     SYSCONF,  NOFLAGS,

662 _SC_THREAD_PROCESS_SHARED, "POSIX_THREAD_PROCESS_SHARED",
663     SYSCONF,  NOFLAGS,
664 _SC_THREAD_PROCESS_SHARED, " _POSIX_THREAD_PROCESS_SHARED",
665     SYSCONF,  NOFLAGS,

667 _SC_THREAD_SAFE_FUNCTIONS, "POSIX_THREAD_SAFE_FUNCTIONS",
668     SYSCONF,  NOFLAGS,
669 _SC_THREAD_SAFE_FUNCTIONS, " _POSIX_THREAD_SAFE_FUNCTIONS",
670     SYSCONF,  NOFLAGS,

672 _SC_TTY_NAME_MAX,     "TTY_NAME_MAX",        SYSCONF,  NOFLAGS,
673 _SC_XOPEN_REALTIME,   " _XOPEN_REALTIME",    SYSCONF,  NOFLAGS,
674 _SC_XOPEN_REALTIME_THREADS, " _XOPEN_REALTIME_THREADS", SYSCONF, NOFLAGS,

```

```

676 /*
677  * Due to ambiguity in the standard we need
678  * to recognize both the underscored name
679  * and the non-underscored version in the
680  * following variables.
681  */
682 _SC_XBS5_ILP32_OFF32, "XBS5_ILP32_OFF32",   SYSCONF,  NOFLAGS,
683 _SC_XBS5_ILP32_OFF32, " _XBS5_ILP32_OFF32",   SYSCONF,  NOFLAGS,

685 _SC_XBS5_ILP32_OFFBIG, "XBS5_ILP32_OFFBIG",   SYSCONF,  NOFLAGS,
686 _SC_XBS5_ILP32_OFFBIG, " _XBS5_ILP32_OFFBIG",   SYSCONF,  NOFLAGS,

688 _SC_XBS5_LP64_OFF64,  "XBS5_LP64_OFF64",   SYSCONF,  NOFLAGS,
689 _SC_XBS5_LP64_OFF64,  " _XBS5_LP64_OFF64",   SYSCONF,  NOFLAGS,

691 _SC_XBS5_LPBIG_OFFBIG, "XBS5_LPBIG_OFFBIG",   SYSCONF,  NOFLAGS,
692 _SC_XBS5_LPBIG_OFFBIG, " _XBS5_LPBIG_OFFBIG",   SYSCONF,  NOFLAGS,

694 /*
695  * UNIX 03 additions for sysconf() names, and names that are not in interfaces
696  */
697 _SC_REGEXP,           " _POSIX_REGEXP",      SYSCONF,  NOFLAGS,
698 _SC_SHELL,            " _POSIX_SHELL",       SYSCONF,  NOFLAGS,
699 _SC_HOST_NAME_MAX,   "HOST_NAME_MAX",     SYSCONF,  NOFLAGS,
700 _SC_READER_WRITER_LOCKS, " _POSIX_READER_WRITER_LOCKS", SYSCONF, NOFLAGS,
701 _SC_IPV6,             " _POSIX_IPV6",        SYSCONF,  NOFLAGS,
702 _SC_RAW_SOCKETS,     " _POSIX_RAW_SOCKETS",    SYSCONF,  NOFLAGS,
703 _SC_XOPEN_STREAMS,   " _XOPEN_STREAMS",     SYSCONF,  NOFLAGS,
704 _SC_SYMLINK_MAX,     "SYMLINK_MAX",       SYSCONF,  NOFLAGS,
705 _SC_2_PBS,           " _POSIX2_PBS",      SYSCONF,  NOFLAGS,
706 _SC_2_PBS_ACCOUNTING, " _POSIX2_PBS_ACCOUNTING", SYSCONF, NOFLAGS,
707 _SC_2_PBS_CHECKPOINT, " _POSIX2_PBS_CHECKPOINT", SYSCONF, NOFLAGS,
708 _SC_2_PBS_LOCATE,    " _POSIX2_PBS_LOCATE",   SYSCONF,  NOFLAGS,
709 _SC_2_PBS_MESSAGE,   " _POSIX2_PBS_MESSAGE",  SYSCONF,  NOFLAGS,
710 _SC_2_PBS_TRACK,     " _POSIX2_PBS_TRACK",   SYSCONF,  NOFLAGS,

712 _SC_ADVISORY_INFO,   " _POSIX_ADVISORY_INFO", SYSCONF,  NOFLAGS,
713 _SC_BARRIERS,        " _POSIX_BARRIERS",     SYSCONF,  NOFLAGS,
714 _SC_CLOCK_SELECTION, " _POSIX_CLOCK_SELECTION", SYSCONF, NOFLAGS,
715 _SC_CPUTIME,         " _POSIX_CPUTIME",     SYSCONF,  NOFLAGS,
716 _SC_MONOTONIC_CLOCK, " _POSIX_MONOTONIC_CLOCK", SYSCONF, NOFLAGS,
717 _SC_SPAWN,           " _POSIX_SPAWN",       SYSCONF,  NOFLAGS,
718 _SC_SPIN_LOCKS,      " _POSIX_SPIN_LOCKS",   SYSCONF,  NOFLAGS,
719 _SC_SPORADIC_SERVER, " _POSIX_SPORADIC_SERVER", SYSCONF, NOFLAGS,
720 _SC_SS_REPL_MAX,     " _POSIX_SS_REPL_MAX",   SYSCONF,  NOFLAGS,
721 _SC_THREAD_CPUTIME,  " _POSIX_THREAD_CPUTIME", SYSCONF, NOFLAGS,
722 _SC_THREAD_SPORADIC_SERVER, " _POSIX_THREAD_SPORADIC_SERVER", SYSCONF, NOFLAGS,
723     NOFLAGS,
724 _SC_TIMEOUTS,        " _POSIX_TIMEOUTS",     SYSCONF,  NOFLAGS,
725 _SC_TRACE,           " _POSIX_TRACE",       SYSCONF,  NOFLAGS,
726 _SC_TRACE_EVENT_FILTER, " _POSIX_TRACE_EVENT_FILTER", SYSCONF, NOFLAGS,
727 _SC_TRACE_EVENT_NAME_MAX, " _POSIX_TRACE_EVENT_NAME_MAX", SYSCONF, NOFLAGS,
728     NOFLAGS,
729 _SC_TRACE_INHERIT,   " _POSIX_TRACE_INHERIT", SYSCONF, NOFLAGS,
730 _SC_TRACE_LOG,       " _POSIX_TRACE_LOG",   SYSCONF,  NOFLAGS,
731 _SC_TRACE_NAME_MAX,  " _POSIX_TRACE_NAME_MAX", SYSCONF, NOFLAGS,
732 _SC_TRACE_SYS_MAX,   " _POSIX_TRACE_SYS_MAX",  SYSCONF,  NOFLAGS,
733 _SC_TRACE_USER_EVENT_MAX, " _POSIX_TRACE_USER_EVENT_MAX", SYSCONF, NOFLAGS,
734     NOFLAGS,
735 _SC_TYPED_MEMORY_OBJECTS, " _POSIX_TYPED_MEMORY_OBJECTS", SYSCONF, NOFLAGS,
736     NOFLAGS,

738 /*
739  * Due to an ambiguity in the specification, we need to recognize both
740  * the underscored POSIX name and the non-underscored name. In addition,
741  * we need to permit the use of the _V6_* environment names specified

```

```

742  * in unistd.h.
743  */
744  _SC_V6_ILP32_OFF32,    "_V6_ILP32_OFF32",    SYSCONF,    NOFLAGS,
745  _SC_V6_ILP32_OFF32,    "_POSIX_V6_ILP32_OFF32", SYSCONF,    NOFLAGS,
746  _SC_V6_ILP32_OFF32,    "POSIX_V6_ILP32_OFF32", SYSCONF,    NOFLAGS,

748  _SC_V6_ILP32_OFFBIG,   "_V6_ILP32_OFFBIG",   SYSCONF,    NOFLAGS,
749  _SC_V6_ILP32_OFFBIG,   "_POSIX_V6_ILP32_OFFBIG", SYSCONF,    NOFLAGS,
750  _SC_V6_ILP32_OFFBIG,   "POSIX_V6_ILP32_OFFBIG", SYSCONF,    NOFLAGS,

752  _SC_V6_LP64_OFF64,     "_V6_LP64_OFF64",     SYSCONF,    NOFLAGS,
753  _SC_V6_LP64_OFF64,     "_POSIX_V6_LP64_OFF64", SYSCONF,    NOFLAGS,
754  _SC_V6_LP64_OFF64,     "POSIX_V6_LP64_OFF64", SYSCONF,    NOFLAGS,

756  _SC_V6_LPBIG_OFFBIG,   "_V6_LPBIG_OFFBIG",   SYSCONF,    NOFLAGS,
757  _SC_V6_LPBIG_OFFBIG,   "_POSIX_V6_LPBIG_OFFBIG", SYSCONF,    NOFLAGS,
758  _SC_V6_LPBIG_OFFBIG,   "POSIX_V6_LPBIG_OFFBIG", SYSCONF,    NOFLAGS,
759  /*
760  * System V Interface Definition
761  */
762  _SC_NPROCESSORS_CONF,  "NPROCESSORS_CONF",   SYSCONF,    NOFLAGS,
763  _SC_NPROCESSORS_MAX,   "NPROCESSORS_MAX",    SYSCONF,    NOFLAGS,
764  _SC_NPROCESSORS_ONLN,  "NPROCESSORS_ONLN",   SYSCONF,    NOFLAGS,
765  0,                      NULL,                  0,          0          /* end of table */
766  };

769  /*
770  * Print usage message.
771  */
772  static int
773  usage()
774  {
776  (void) fputs(gettext(
777  "usage:\tgetconf [-v specification] system_var\n"
778  "\tgetconf [-v specification] path_var pathname\n"
779  "\tgetconf -a\n"), stderr);
780  return (2);
781  }

784  static int
785  namecmp(const void *a, const void *b)
786  {
787  return (strcoll(((const struct sctab *)a)->name,
788  ((const struct sctab *)b)->name));
789  }

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

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

```

```

808  if (scp->flag & UNSIGNED_VALUE)
809  (void) printf("%lu\n", scp->value);
810  else
811  (void) printf("%ld\n", scp->value);
812  break;

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

833  case CONFSTR:
834  if (argc > 2)
835  return (usage());
836  errno = 0;
837  len = confstr((int)scp->value, NULL, (size_t)0);
838  if (len == 0)
839  if (errno == EINVAL) {
840  (void) fprintf(stderr, gettext(INVAL_ARG),
841  name);
842  return (1);
843  } else {
844  /* DO NOT TRANSLATE */
845  (void) printf("undefined\n");
846  return (0);
847  }
848  /*
849  * allocate space to store result of constr() into
850  */
851  if ((buffer = malloc(len)) == NULL) {
852  (void) fprintf(stderr,
853  gettext("insufficient memory available"));
854  return (1);
855  }

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

859  (void) printf("%s\n", buffer);
860  free(buffer);
861  break;

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

```

```

874         /* Probably problems with the pathname itself */
875         if (aflag) {
876             (void) printf("%s\n", file);
877         } else {
878             (void) printf("%s", file);
879         }
880         return (1);
881     }
882     }
883     (void) printf("%ld\n", value);
884     break;
885 }
886 return (0);
887 }

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

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

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

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

932 #define TabStop      8
933 #define RightColumn  32
934 #define DefPathName  "."

936         /*
937          * sort the table by the "name" field
938          * so we print it in sorted order
939          */

```

```

940     qsort(&sctab[0], (sizeof (sctab)/sizeof (struct sctab))-1,
941          sizeof (struct sctab), namecmp);

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

949         /*
950          * create 2 columns:
951          *   variable name in the left column,
952          *   value in the right column.
953          * The right column starts at a tab stop.
954          */
955         (void) printf("%s:\t", scp->name);

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

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

969         exstat |= stat;

971         /*
972          * if stat != 0, then an error message was already
973          * printed in getconf(),
974          * so don't need to print one here
975          */
976     } else {

979         /*
980          * find the name of the specified variable (argv[1])
981          * and print its value.
982          */
983         for (scp = &sctab[0]; scp->name != NULL; ++scp)
984             if (strcmp(argv[1], scp->name) == 0) {
985                 exstat = getconf(scp, argc, argv[1], argv[2]);
986                 break;
987             }

989         /*
990          * if at last entry in table, then the user specified
991          * variable is invalid
992          */
993         if (scp->name == NULL) {
994             errno = EINVAL;
995             (void) fprintf(stderr, gettext(INVAL_ARG), argv[1]);
996             return (1);
997         }

998     }
999     return (exstat);
1000 }

```

```

*****
17542 Fri Sep 27 14:13:12 2013
new/usr/src/man/man1/getconf.1
4146 getconf(1) does not support XATTR_ENABLED, XATTR_EXISTS, SATTR_ENABLED, SAT
Reviewed by: Cedric Blancher <cedric.blancher@gmail.com>
*****
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\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
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-FC\fR
196 .sp
197 \fB\fBconfstr\fR(3C),
198 .RE
199 .RS +4
200 .TP
201 .ie t \(\bu
202 .el o
203 \fB\fBFC\fR
204 .sp
205 \fB\fBpathconf\fR(2)
206 .RE
207 .RS +4
208 .TP
209 .ie t \(\bu
210 .el o
211 \fB\fBFC\fR
212 .sp
213 \fB\fBsysconf\fR(3C)
214 .RE
215 .RS +4
216 .TP
217 .ie t \(\bu
218 .el o
219 \fB\fBFI\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\fIpath_var\fR\fR
398 .ad
399 .RS 12n
400 A name of a configuration variable whose value is available from the
401 \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 ACL_ENABLED      NAME_MAX      XATTR_ENABLED
412 LINK_MAX        PATH_MAX      XATTR_EXISTS
413 MAX_CANON       PIPE_BUF      _POSIX_CHOWN_RESTRICTED
414 MAX_INPUT       SATTR_ENABLED _POSIX_NO_TRUNC
415 MIN_HOLE_SIZE   SATTR_EXISTS _POSIX_VDISABLE
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

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 \fBbconfstr\fR(3C) or \fBsysconf\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_REGEXP
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

583 .sp
584 .in +2

```

```

585 .nf
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 \fBsysconf(_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 `\fBsysconf\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\fR` call to
654 `\fBgetconf\fR` always supplies a value corresponding to conditions when the
655 program is executed.

657 .SH ENVIRONMENT VARIABLES

658 .sp
659 .LP
660 See `\fBenviron\fR(5)` for descriptions of the following environment variables
661 that affect the execution of `\fBgetconf\fR`: `\fBLANG\fR`, `\fBLC_ALL\fR`,
662 `\fBLC_CTYPE\fR`, `\fBLC_MESSAGES\fR`, and `\fBNLSPATH\fR`.

663 .sp
664 .LP
665 The following environment variable is supported by the `\fBksh93\fR` built-in
666 `\fBgetconf\fR` command:

667 .sp
668 .ne 2
669 .na
670 `\fB\FB_AST_FEATURES\fR\fR`

671 .ad
672 .RS 17n
673 Local writable values that are different from the default are stored in the
674 `\fB\FB_AST_FEATURES\fR` environment variable. The `\fB\FB_AST_FEATURES\fR` value is a
675 space-separated list of `\fIname` path value `\fR` 3-tuples, where `\fIname\fR` is the
676 system configuration name, `\fIpath\fR` is the corresponding path, `\fB-\fR` if no
677 path is applicable, and `\fIvalue\fR` 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\fR\fR`
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\fR\fR`
698 .ad
699 .RS 6n
700 An error occurred.
701 .RE

703 .SH ATTRIBUTES

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