

new/usr/src/head/iso/stdio_iso.h

1

```
*****
11166 Sun Jun 16 11:04:39 2013
new/usr/src/head/iso/stdio_iso.h
Fix clang errors
Used Albert's patch to stdio_iso.h
Formatting issues
3781 gcc4.7 __cplusplus change incompatibility
*****
1 /*
2 * CDDL HEADER START
3 *
4 * The contents of this file are subject to the terms of the
5 * Common Development and Distribution License, Version 1.0 only
6 * (the "License"). You may not use this file except in compliance
7 * with the License.
8 *
9 * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 */

27 /*      Copyright (c) 1988 AT&T */
28 /*      All Rights Reserved      */

30 /*
31 * An application should not include this header directly. Instead it
32 * should be included only through the inclusion of other Sun headers.
33 *
34 * The contents of this header is limited to identifiers specified in the
35 * C Standard. Any new identifiers specified in future amendments to the
36 * C Standard must be placed in this header. If these new identifiers
37 * are required to also be in the C++ Standard "std" namespace, then for
38 * anything other than macro definitions, corresponding "using" directives
39 * must also be added to <stdio.h>.
40 */

42 /*
43 * User-visible pieces of the ANSI C standard I/O package.
44 */

46 #ifndef _ISO_STDIO_ISO_H
47 #define _ISO_STDIO_ISO_H

49 #pragma ident      "%Z%M% %I%      %E% SMI"

49 #include <sys/feature_tests.h>
50 #include <sys/va_list.h>
51 #include <stdio_tag.h>
52 #include <stdio_impl.h>

54 /*
55 * If feature test macros are set that enable interfaces that use types
56 * defined in <sys/types.h>, get those types by doing the include.
```

new/usr/src/head/iso/stdio_iso.h

2

```
57 *
58 * Note that in asking for the interfaces associated with this feature test
59 * macro one also asks for definitions of the POSIX types.
60 */

62 #ifdef __cplusplus
63 extern "C" {
64 #endif

66 #if !defined(_LP64) && (_FILE_OFFSET_BITS == 64 || defined(_LARGEFILE64_SOURCE))
67 /*
68 * The following typedefs are adopted from ones in <sys/types.h> (with leading
69 * underscores added to avoid polluting the ANSI C name space). See the
70 * commentary there for further explanation.
71 */
72 #if defined(_LONGLONG_TYPE)
73 typedef long long      __longlong_t;
74 #else
75 /* used to reserve space and generate alignment */
76 typedef union {
77     double   _d;
78     int      _l[2];
79 } __longlong_t;
__unchanged_portion_omitted_
328 #endif /* end of namespace std */

330 #if !defined(__lint)

332 #if      !defined(_REENTRANT) && !defined(_LP64)
334 #if      !defined(_REENTRANT) && !defined(_LP64) && !defined(_STRICT_STDC)

334 #ifdef __STDC__
335 #if __cplusplus >= 199711L
336 namespace std {
337 inline int getc(FILE *_p) {
338     return (--p->_cnt < 0 ? __filbuf(_p) : (int)*p->_ptr++); }
339 inline int putc(int _x, FILE *_p) {
340     return (--p->_cnt < 0 ? __flsbuf(_x, _p)
341         : (int)(*p->_ptr++ = (unsigned char) _x)); }
342 }
343 #else /* __cplusplus >= 199711L */
344 #define getc(p)      (--(p)->_cnt < 0 ? __filbuf(p) : (int)*(p)->_ptr++)
345 #define putc(x, p)  (--(p)->_cnt < 0 ? __flsbuf((x), (p)) \
346                     : (int)*(p)->_ptr++ = (unsigned char) (x))
347 #endif /* __cplusplus >= 199711L */
348 #else /* __STDC__ */
349 #define getc(p)      (--(p)->_cnt < 0 ? __filbuf(p) : (int)*(p)->_ptr++)
350 #define putc(x, p)  (--(p)->_cnt < 0 ? __flsbuf((x), (p)) : \
351                     (int)*(p)->_ptr++ = (unsigned char) (x))
352 #endif /* __STDC__ */

354 #endif /* !defined(_REENTRANT) && !defined(_LP64) */
356 #endif /* !defined(_REENTRANT) && !defined(_LP64) && !defined(_STRICT_STDC) */

356 #ifndef _REENTRANT

358 #if __cplusplus >= 199711L
359 namespace std {
360 inline int getchar() { return getc(stdin); }
361 inline int putchar(int _x) { return putc(_x, stdout); }
362 }
__unchanged_portion_omitted_
```

new/usr/src/head/iso/stdlib_iso.h

1

```
*****
6010 Sun Jun 16 11:04:42 2013
new/usr/src/head/iso/stdlib_iso.h
Fix clang errors
3781 gcc4.7 __cplusplus change incompatibility
3781 gcc4.7 __cplusplus change incompatibility
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*
23 * Copyright 2005 Sun Microsystems, Inc. All rights reserved.
24 * Use is subject to license terms.
25 */

27 /*      Copyright (c) 1988 AT&T */
28 /*      All Rights Reserved */

31 /*
32 * An application should not include this header directly. Instead it
33 * should be included only through the inclusion of other Sun headers.
34 *
35 * The contents of this header is limited to identifiers specified in the
36 * C Standard. Any new identifiers specified in future amendments to the
37 * C Standard must be placed in this header. If these new identifiers
38 * are required to also be in the C++ Standard "std" namespace, then for
39 * anything other than macro definitions, corresponding "using" directives
40 * must also be added to <locale.h>.
41 */

43 #ifndef _ISO_STDLIB_ISO_H
44 #define _ISO_STDLIB_ISO_H

46 #pragma ident      "%Z%M% %I%      %E% SMI"

46 #include <sys/feature_tests.h>

48 #ifdef __cplusplus
49 extern "C" {
50 #endif

52 #if defined(__STDC__)
53 extern unsigned char      _ctype[];
54 #define MB_CUR_MAX      _ctype[520]
55 #else
56 extern unsigned char      _ctype[];
57 #define MB_CUR_MAX      _ctype[520]

```

new/usr/src/head/iso/stdlib_iso.h

2

```
58 #endif

60 #if __cplusplus >= 199711L
61 namespace std {
62 #endif

64 typedef struct {
65     int      quot;
66     int      rem;
67 } div_t;
unchanged_portion_omitted

74 #if !defined(_SIZE_T) || __cplusplus >= 199711L
75 #define _SIZE_T
76 #if defined(_LP64) || defined(_I32LPx)
77 typedef unsigned long      size_t;      /* size of something in bytes */
78 #else
79 typedef unsigned int      size_t;      /* (historical version) */
80 #endif
81 #endif /* !_SIZE_T */

83 #ifndef NULL
84 #if defined(_LP64)
85 #define NULL      0L
86 #else
87 #define NULL      0
88 #endif
89 #endif

91 #define EXIT_FAILURE      1
92 #define EXIT_SUCCESS      0
93 #define RAND_MAX      32767

95 /*
96 * wchar_t is a built-in type in standard C++ and as such is not
97 * defined here when using standard C++. However, the GNU compiler
98 * fixincludes utility nonetheless creates its own version of this
99 * header for use by gcc and g++. In that version it adds a redundant
100 * guard for __cplusplus. To avoid the creation of a gcc/g++ specific
101 * header we need to include the following magic comment:
102 *
103 * we must use the C++ compiler's type
104 *
105 * The above comment should not be removed or changed until GNU
106 * gcc/fixinc/inclhack.def is updated to bypass this header.
107 */
108 #if !defined(__cplusplus) || (__cplusplus < 199711L && !defined(__GNUG__))
109 #ifndef _WCHAR_T
110 #define _WCHAR_T
111 #if defined(_LP64)
112 typedef int      wchar_t;
113 #else
114 typedef long      wchar_t;
115 #endif
116 #endif /* !_WCHAR_T */
117 #endif /* !defined(__cplusplus) ... */

119 #if defined(__STDC__)

121 extern void abort(void) __NORETURN;
122 extern int abs(int);
123 extern int atexit(void (*)(void));
124 extern double atof(const char *);
125 extern int atoi(const char *);
126 extern long int atol(const char *);
127 extern void *bsearch(const void *, const void *, size_t, size_t,
```

```
128     int (*)(const void *, const void *));
129 #if __cplusplus >= 199711L && defined(__SUNPRO_CC)
130 #if __cplusplus >= 199711L
131 extern "C++" {
132     void *bsearch(const void *, const void *, size_t, size_t,
133                 int (*)(const void *, const void *));
134 }
135 #endif /* __cplusplus >= 199711L && defined(__SUNPRO_CC) */
136 #endif /* __cplusplus >= 199711L */
137 extern void *calloc(size_t, size_t);
138 extern div_t div(int, int);
139 extern void exit(int)
140     __NORETURN;
141 extern void free(void *);
142 extern char *getenv(const char *);
143 extern long int labs(long);
144 extern ldiv_t ldiv(long, long);
145 extern void *malloc(size_t);
146 extern int mblen(const char *, size_t);
147 extern int mbtowc(wchar_t *_RESTRIC_KYWD, const char *_RESTRIC_KYWD, size_t);
148 extern void qsort(void *, size_t, size_t, int (*)(const void *, const void *));
149 #if __cplusplus >= 199711L && defined(__SUNPRO_CC)
150 #if __cplusplus >= 199711L
151 extern "C++" {
152     void qsort(void *, size_t, size_t, int (*)(const void *, const void *));
153 }
154 #endif /* __cplusplus >= 199711L && defined(__SUNPRO_CC) */
155 #endif /* __cplusplus >= 199711L */
156 extern int rand(void);
157 extern void *realloc(void *, size_t);
158 extern void srand(unsigned int);
159 extern double strtod(const char *_RESTRIC_KYWD, char **_RESTRIC_KYWD);
160 extern long int strtol(const char *_RESTRIC_KYWD, char **_RESTRIC_KYWD, int);
161 extern unsigned long int strtoul(const char *_RESTRIC_KYWD,
162     char **_RESTRIC_KYWD, int);
163 extern int system(const char *);
164 extern int wctomb(char *, wchar_t);
165 extern size_t wcstombs(char *_RESTRIC_KYWD, const wchar_t *_RESTRIC_KYWD,
166     size_t);
167
168 #if __cplusplus >= 199711L
169 extern "C++" {
170     inline long  abs(long _l) { return labs(_l); }
171     inline ldiv_t div(long _l1, long _l2) { return ldiv(_l1, _l2); }
172 }
173
174 unchanged_portion_omitted
```

```

*****
5747 Sun Jun 16 11:04:43 2013
new/usr/src/head/wchar.h
Fix clang errors
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */

22 /*
23  * Copyright (c) 1993, 2010, Oracle and/or its affiliates. All rights reserved.
24 */

26 #ifndef _WCHAR_H
27 #define _WCHAR_H

29 #include <sys/feature_tests.h>
30 #include <iso/wchar_iso.h>
31 #include <iso/wchar_c99.h>

33 /*
34  * Allow global visibility for symbols defined in
35  * C++ "std" namespace in <iso/wchar_iso.h>.
36  */
37 #if __cplusplus >= 199711L
38 using std::FILE;
39 using std::wint_t;
40 using std::clock_t;
41 using std::size_t;
42 using std::time_t;
43 using std::tm;
44 using std::mbstate_t;
45 using std::fgetwc;
46 using std::fgetws;
47 using std::fputwc;
48 using std::fputws;
49 using std::ungetwc;
50 using std::getwc;
51 using std::getwchar;
52 using std::putwc;
53 using std::putwchar;
54 using std::wcstod;
55 using std::wcstol;
56 using std::wcstoul;
57 using std::wcsat;
58 using std::wcschr;
59 using std::wcsncmp;
60 using std::wscoll;
61 using std::wcsncpy;

```

```

62 using std::wcsncpy;
63 using std::wcslen;
64 using std::wcsnecat;
65 using std::wcsncmp;
66 using std::wcsncpy;
67 using std::wcpbrk;
68 using std::wcsrchr;
69 using std::wcsspn;
70 using std::wcssxfrm;
71 using std::wcstok;
72 using std::wcsftime;
73 /* not XPG4 and not XPG4v2 */
74 #if (!defined(_XPG4) && !defined(_XPG4_2) || defined(_XPG5))
75 using std::btowc;
76 using std::fwprintf;
77 using std::fwscanf;
78 using std::fwide;
79 using std::mbsinit;
80 using std::mbrlen;
81 using std::mbrtowc;
82 using std::mbsrtowcs;
83 using std::swprintf;
84 using std::swscanf;
85 using std::vfwprintf;
86 using std::vwprintf;
87 using std::vswprintf;
88 using std::wcrctomb;
89 using std::wcrctombs;
90 using std::wcsstr;
91 using std::wctob;
92 using std::wmemchr;
93 using std::wmemcmp;
94 using std::wmemcpy;
95 using std::wmemmove;
96 using std::wmemset;
97 using std::wprintf;
98 using std::wscanf;
99 #endif /* not XPG4 and not XPG4v2 */
100 #endif /* __cplusplus >= 199711L */

102 #ifdef __cplusplus
103 extern "C" {
104 #endif

106 #if !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE) || defined(__EXTENSIONS__)
107 #if !defined(_WCTYPE_T) || __cplusplus >= 199711L
108 #define _WCTYPE_T
109 typedef int wctype_t;
110 #endif
111 #endif /* !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE)... */

113 /*
114  * XPG6 requires that va_list be defined as defined in <stdarg.h>,
115  * however, inclusion of <stdarg.h> breaks Standard C namespace.
116  */
117 #if defined(_XPG6) && !defined(_VA_LIST)
118 #define _VA_LIST
119 typedef __va_list va_list;
120 #endif /* defined(_XPG6) && !defined(_VA_LIST) */

122 #ifdef __STDC__

124 #if !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE) || defined(__EXTENSIONS__)
125 #if __cplusplus >= 199711L
126 namespace std {
127 #endif

```

```

128 #endif /* ! codereview */
129 extern int iswalpha(wint_t);
130 extern int iswupper(wint_t);
131 extern int iswlower(wint_t);
132 extern int iswdigit(wint_t);
133 extern int iswxdigit(wint_t);
134 extern int iswalnum(wint_t);
135 extern int iswspace(wint_t);
136 extern int iswpunct(wint_t);
137 extern int iswprint(wint_t);
138 extern int iswgraph(wint_t);
139 extern int iswcntrl(wint_t);
140 extern int iswctype(wint_t, wctype_t);
141 extern wint_t tolower(wint_t);
142 extern wint_t toupper(wint_t);
143 extern wchar_t *wcswcs(const wchar_t *, const wchar_t *);
144 extern int wcswidth(const wchar_t *, size_t);
145 extern int wwidth(wchar_t);
146 extern wctype_t wctype(const char *);
147 #if __cplusplus >= 199711L
148 } /* namespace std */

150 using std::iswalpha;
151 using std::iswupper;
152 using std::iswlower;
153 using std::iswdigit;
154 using std::iswxdigit;
155 using std::iswalnum;
156 using std::iswspace;
157 using std::iswpunct;
158 using std::iswprint;
159 using std::iswgraph;
160 using std::iswcntrl;
161 using std::iswctype;
162 using std::tolower;
163 using std::toupper;
164 using std::wcswcs;
165 using std::wcswidth;
166 using std::wwidth;
167 using std::wctype;
168 #endif
169 #endif /* ! codereview */
170 #endif /* !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE)... */

172 #if defined(__EXTENSIONS__) || \
173     (!defined(_STRICT_STDC) && !defined(_XOPEN_OR_POSIX))
174     /* || defined(_XPG7) */
175 extern wchar_t *wcsdup(const wchar_t *);
176 extern size_t wcsnlen(const wchar_t *, size_t);
177 extern wchar_t *wcpncpy(wchar_t *RESTRICT_KYWD, const wchar_t *RESTRICT_KYWD);
178 extern wchar_t *wcpncpy(wchar_t *RESTRICT_KYWD, const wchar_t *RESTRICT_KYWD,
179     size_t);
180 extern int wcscasecmp(const wchar_t *, const wchar_t *);
181 extern int wcsncasecmp(const wchar_t *, const wchar_t *, size_t);
182 #endif

184 #else /* __STDC__ */

186 #if !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE) || defined(__EXTENSIONS__)
187 extern int iswalpha();
188 extern int iswupper();
189 extern int iswlower();
190 extern int iswdigit();
191 extern int iswxdigit();
192 extern int iswalnum();
193 extern int iswspace();

```

```

194 extern int iswpunct();
195 extern int iswprint();
196 extern int iswgraph();
197 extern int iswcntrl();
198 extern int iswctype();
199 extern wint_t tolower();
200 extern wint_t toupper();
201 extern wchar_t *wcswcs();
202 extern int wcswidth();
203 extern int wwidth();
204 extern wctype_t wctype();
205 #endif /* !defined(_STRICT_STDC) || defined(_XOPEN_SOURCE)... */

207 #if defined(__EXTENSIONS__) || \
208     (!defined(_STRICT_STDC) && !defined(_XOPEN_OR_POSIX))
209     /* || defined(_XPG7) */
210 extern wchar_t *wcsdup();
211 extern size_t wcsnlen();
212 extern wchar_t *wcpncpy();
213 extern wchar_t *wcpncpy();
214 extern int wcscasecmp();
215 extern int wcsncasecmp();
216 #endif

218 #endif /* __STDC__ */

220 #ifdef __cplusplus
221 }
222 #endif

224 #endif /* _WCHAR_H */

```