NAME

stdlib.h − standard library definitions

SYNOPSIS

#include <stdlib.h>

DESCRIPTION

Some of the functionality described on this reference page extends the ISO C standard. Applications shall define the appropriate feature test macro (see the System Interfaces volume of IEEE Std 1003.1-2001, Section 2.2, The Compilation Environment) to enable the visibility of these symbols in this header.

The <*stdlib.h*> header shall define the following macros:

EXIT_FAILURE
: Unsuccessful termination for *exit*(); evaluates to a non-zero value.

EXIT_SUCCESS
: Successful termination for *exit*(); evaluates to 0.

NULL Null pointer.

{RAND_MAX}
: Maximum value returned by *rand*(); at least 32767.

{MB_CUR_MAX}
: Integer expression whose value is the maximum number of bytes in a character specified by the current locale.

The following data types shall be defined through **typedef**:

div_t Structure type returned by the *div*() function.

ldiv_t Structure type returned by the *ldiv*() function.

lldiv_t Structure type returned by the *lldiv*() function.

size_t As described in <*stddef.h*> .

wchar_t
: As described in <*stddef.h*> .

In addition, the following symbolic names and macros shall be defined as in <*sys/wait.h*>, for use in decoding the return value from *system*():

WNOHANG
WUNTRACED
WEXITSTATUS
WIFEXITED
WIFSIGNALED
WIFSTOPPED
WSTOPSIG
WTERMSIG

The following shall be declared as functions and may also be defined as macros. Function prototypes shall be provided.

void _Exit(int);

long a64l(const char *);

void abort(void);

```
int         abs(int);
int         atexit(void (*)(void));
double      atof(const char *);
int         atoi(const char *);
long        atol(const char *);
long long   atoll(const char *);
void        *bsearch(const void *, const void *, size_t, size_t,
              int (*)(const void *, const void *));
void        *calloc(size_t, size_t);
div_t       div(int, int);

double      drand48(void);
char        *ecvt(double, int, int *restrict, int *restrict); (LEGACY )
double      erand48(unsigned short[3]);

void        exit(int);

char        *fcvt(double, int, int *restrict, int *restrict); (LEGACY )

void        free(void *);

char        *gcvt(double, int, char *); (LEGACY )

char        *getenv(const char *);

int         getsubopt(char **, char *const *, char **);
int         grantpt(int);
char        *initstate(unsigned, char *, size_t);
long        jrand48(unsigned short[3]);
char        *l64a(long);

long        labs(long);

void        lcong48(unsigned short[7]);

ldiv_t      ldiv(long, long);
long long   llabs(long long);
lldiv_t     lldiv(long long, long long);

long        lrand48(void);

void        *malloc(size_t);
int         mblen(const char *, size_t);
size_t      mbstowcs(wchar_t *restrict, const char *restrict, size_t);
int         mbtowc(wchar_t *restrict, const char *restrict, size_t);

char        *mktemp(char *); (LEGACY )
int         mkstemp(char *);
long        mrand48(void);
long        nrand48(unsigned short[3]);


int         posix_memalign(void **, size_t, size_t);
```

```
int          posix_openpt(int);
char         *ptsname(int);
int          putenv(char *);

void         qsort(void *, size_t, size_t, int (*)(const void *,
                 const void *));
int          rand(void);

int          rand_r(unsigned *);


long         random(void);

void         *realloc(void *, size_t);

char         *realpath(const char *restrict, char *restrict);
unsigned short seed48(unsigned short[3]);


int          setenv(const char *, const char *, int);


void         setkey(const char *);
char         *setstate(const char *);

void         srand(unsigned);

void         srand48(long);
void         srandom(unsigned);

double       strtod(const char *restrict, char **restrict);
float        strtof(const char *restrict, char **restrict);
long         strtol(const char *restrict, char **restrict, int);
long double  strtold(const char *restrict, char **restrict);
long long    strtoll(const char *restrict, char **restrict, int);
unsigned long strtoul(const char *restrict, char **restrict, int);
unsigned long long
             strtoull(const char *restrict, char **restrict, int);
int          system(const char *);

int          unlockpt(int);


int          unsetenv(const char *);

size_t       wcstombs(char *restrict, const wchar_t *restrict, size_t);
int          wctomb(char *, wchar_t);
```

Inclusion of the <*stdlib.h*> header may also make visible all symbols from <*stddef.h*>, <*limits.h*>, <*math.h*>, and <*sys/wait.h*>.

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

<limits.h>, *<math.h>*, *<stddef.h>*, *<sys/types.h>*, *<sys/wait.h>*, the System Interfaces volume of IEEE Std 1003.1-2001, *_Exit*(), *a64l*(), *abort*(), *abs*(), *atexit*(), *atof*(), *atoi*(), *atol*(), *atoll*(), *bsearch*(), *calloc*(), *div*(), *drand48*(), *erand48*(), *exit*(), *free*(), *getenv*(), *getsubopt*(), *grantpt*(), *initstate*(), *jrand48*(), *l64a*(), *labs*(), *lcong48*(), *ldiv*(), *llabs*(), *lldiv*(), *lrand48*(), *malloc*(), *mblen*(), *mbstowcs*(), *mbtowc*(), *mkstemp*(), *mrand48*(), *nrand48*(), *posix_memalign*(), *ptsname*(), *putenv*(), *qsort*(), *rand*(), *realloc*(), *realpath*(), *setstate*(), *srand*(), *srand48*(), *srandom*(), *strtod*(), *strtof*(), *strtol*(), *strtold*(), *strtoll*(), *strtoul*(), *strtoull*(), *unlockpt*(), *wcstombs*(), *wctomb*()

COPYRIGHT