

# Cygwin API Reference

---

# Cygwin API Reference

Copyright © 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014 Red Hat, Inc.

Permission is granted to make and distribute verbatim copies of this documentation provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this documentation under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this documentation into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Free Software Foundation.

---

---

# Table of Contents

1. Compatibility .....	1
System interfaces compatible with the Single Unix Specification, Version 4: .....	2
System interfaces compatible with BSD functions: .....	18
System interfaces compatible with GNU or Linux extensions: .....	20
System interfaces compatible with Solaris or SunOS functions: .....	22
Other UNIX system interfaces, deprecated or not in POSIX.1-2008: .....	24
NOT implemented system interfaces from the Single Unix Specification, Volume 4: .....	25
Implementation Notes .....	28
2. Cygwin Functions .....	29
Path conversion functions .....	30
cygwin_conv_path .....	30
cygwin_conv_path_list .....	31
cygwin_create_path .....	31
cygwin_posix_path_list_p .....	31
cygwin_split_path .....	32
Helper functions to change user context .....	33
cygwin_logon_user .....	33
cygwin_set_impersonation_token .....	33
Miscellaneous functions .....	34
cygwin_attach_handle_to_fd .....	34
cygwin_internal .....	34
cygwin_stackdump .....	34

---

## List of Examples

2.1. Example use of <code>cygwin_conv_path</code> .....	30
2.2. Example use of <code>cygwin_split_path</code> .....	32

---

# Chapter 1. Compatibility

# System interfaces compatible with the Single Unix Specification, Version 4:

Note that the core of the Single Unix Specification, Version 4 is also IEEE Std 1003.1-2008 (POSIX.1-2008).

FD\_CLR  
FD\_ISSET  
FD\_SET  
FD\_ZERO  
\_Exit  
\_exit  
\_longjmp  
\_setjmp  
\_tolower  
\_toupper  
a64l  
abort  
abs  
accept  
access  
acos  
acosf  
acosh  
acoshf  
alarm  
alphasort  
asctime  
asctime\_r  
asin  
asinf  
asinh  
asinhf  
atan  
atan2  
atan2f  
atanf  
atanh  
atanhf  
atexit  
atof  
atoff  
atoi  
atol  
atoll  
basename  
bind  
bsearch  
btowc  
cabs  
cabsf  
cacos  
cacosf  
cacosh  
cacoshf  
calloc  
carg  
cargf  
casin

casinf  
casinh  
casinhf  
casinhl  
catan  
catanf  
catanh  
catanhf  
catclose (available in external "catgets" library)  
catgets (available in external "catgets" library)  
catopen (available in external "catgets" library)  
cbrt  
cbrtf  
ccos  
ccosf  
ccosh  
ccoshf  
ceil  
ceilf  
cexp  
cexpf  
cfgetispeed  
cfgetospeed  
cfsetispeed  
cfsetospeed  
chdir  
chmod  
chown  
cimag  
cimagf  
clearerr  
clock  
clock\_getcpuclockid  
clock\_getres  
clock\_gettime  
clock\_nanosleep (see chapter "Implementation Notes")  
clock\_settime (see chapter "Implementation Notes")  
clog  
clogf  
close  
closedir  
closelog  
confstr  
conj  
conjf  
connect  
copysign  
copysignf  
cos  
cosf  
cosh  
coshf  
cpow  
cpowf  
cproj  
cprojf  
creal  
crealf  
creat  
crypt (available in external "crypt" library)  
csin  
csinf  
csinh

csinhf  
csqrt  
csqrtf  
ctan  
ctanf  
ctanh  
ctanhf  
ctermid  
ctime  
ctime\_r  
daylight  
dbm\_clearerr (available in external "libgdbm" library)  
dbm\_close (available in external "libgdbm" library)  
dbm\_delete (available in external "libgdbm" library)  
dbm\_error (available in external "libgdbm" library)  
dbm\_fetch (available in external "libgdbm" library)  
dbm\_firstkey (available in external "libgdbm" library)  
dbm\_nextkey (available in external "libgdbm" library)  
dbm\_open (available in external "libgdbm" library)  
dbm\_store (available in external "libgdbm" library)  
difftime  
dirfd  
dirname  
div  
dlclose  
dlerror  
dlopen  
dlsym  
dprintf  
drand48  
dup  
dup2  
encrypt (available in external "crypt" library)  
endgrent  
endhostent  
endprotoent  
endpwent  
endservent  
endutxent  
environ  
erand48  
erf  
erfc  
erfcf  
erff  
errno  
execl  
execle  
execlp  
execv  
execve  
execvp  
exit  
exp  
exp2  
exp2f  
expf  
expm1  
expm1f  
fabs  
fabsf  
faccessat  
fchdir



fchmod  
fchmodat  
fchown  
fchownat  
fclose  
fcntl (see chapter "Implementation Notes")  
fdatasync  
fdim  
fdimf  
fdopen  
fdopendir  
feclearexcept  
fegetenv  
fegetexceptflag  
fegetround  
feholdexcept  
feof  
feraiseexcept  
ferror  
fesetenv  
fesetexceptflag  
fesetround  
fetestexcept  
feupdateenv  
fexecve  
fflush  
ffs  
fgetc  
fgetpos  
fgets  
fgetwc  
fgetws  
fileno  
flockfile  
floor  
floorf  
fma  
fmaf  
fmax  
fmaxf  
fmemopen  
fmin  
fminf  
fmod  
fmodf  
fnmatch  
fopen  
fork  
fpathconf  
fpclassify (see chapter "Implementation Notes")  
fprintf  
fputc  
fputs  
fputwc  
fputws  
fread  
free  
freeaddrinfo  
freopen  
frexp  
frexpf  
fscanf  
fseek

fseeko  
fsetpos  
fstat  
fstatat  
fstatvfs  
fsync  
ftell  
ftello  
ftok  
ftruncate  
ftrylockfile  
ftw  
funlockfile  
futimens  
fwide  
fwprintf  
fwrite  
fwscanf  
gai\_strerror  
getaddrinfo  
getc  
getc\_unlocked  
getchar  
getchar\_unlocked  
getcwd  
getdelim  
getdomainname  
getegid  
getenv  
geteuid  
getgid  
getgrent  
getgrgid  
getgrgid\_r  
getgrnam  
getgrnam\_r  
getgroups  
gethostid  
gethostname  
getitimer (see chapter "Implementation Notes")  
getline  
getlogin  
getlogin\_r  
getnameinfo  
getopt  
getpeername  
getpgid  
getpgrp  
getpid  
getppid  
getpriority  
getprotobyname  
getprotobyname  
getprotoent  
getpwent  
getpwnam  
getpwnam\_r  
getpwuid  
getpwuid\_r  
getrlimit  
getrusage  
gets  
getservbyname

getservbyport  
getservent  
getsid  
getsockname  
getsockopt  
getsubopt  
gettimeofday  
getuid  
getutxent  
getutxid  
getutxline  
getwc  
getwchar  
glob  
globfree  
gmtime  
gmtime\_r  
grantpt  
hcreate  
hdestroy  
hsearch  
htonl  
htons  
hypot  
hypotf  
iconv (available in external "libiconv" library)  
iconv\_close (available in external "libiconv" library)  
iconv\_open (available in external "libiconv" library)  
if\_freenameindex  
if\_indextoname  
if\_nameindex  
if\_nametoindex  
ilogb  
ilogbf  
imaxabs  
imaxdiv  
inet\_addr  
inet\_ntoa  
inet\_ntop  
inet\_pton  
initstate  
insque  
ioctl  
isalnum  
isalpha  
isascii  
isatty  
isblank  
iscntrl  
isdigit  
isfinite (see chapter "Implementation Notes")  
isgraph  
isgreater (see chapter "Implementation Notes")  
isgreaterequal (see chapter "Implementation Notes")  
isinf (see chapter "Implementation Notes")  
isless  
islessequal (see chapter "Implementation Notes")  
islessgreater (see chapter "Implementation Notes")  
islower  
isnan (see chapter "Implementation Notes")  
isnormal (see chapter "Implementation Notes")  
isprint  
ispunct

isspace  
isunordered (see chapter "Implementation Notes")  
isupper  
iswalnum  
iswalpha  
iswblank  
iswcntrl  
iswctype  
iswdigit  
iswgraph  
iswlower  
iswprint  
iswpunct  
iswspace  
iswupper  
iswxdigit  
isxdigit  
j0  
j1  
jn  
jrand48  
kill  
killpg  
l64a  
labs  
lchown  
lcong48  
ldexp  
ldexpf  
ldiv  
lfind  
lgamma  
lgammaf  
link  
linkat  
listen  
llabs  
lldiv  
llrint  
llrintf  
llrintl  
llround  
llroundf  
localeconv  
localtime  
localtime\_r  
lockf (see chapter "Implementation Notes")  
log  
log10  
log10f  
loglp  
loglpf  
log2  
log2f  
logb  
logbf  
logf  
longjmp  
lrand48  
lrint  
lrintf  
lrintl  
lround

lroundf  
lsearch  
lseek  
lstat  
malloc  
mblen  
mbrlen  
mbrtowc  
mbsinit  
mbsnrtowcs  
mbsrtowcs  
mbstowcs  
mbtowc  
memccpy  
memchr  
memcmp  
memcpy  
memmove  
memset  
mkdir  
mkdirat  
mkdtemp  
mkfifo  
mkfifoat  
mknod  
mknodat  
mkstemp  
mktime  
mlock  
mmap  
modf  
modff  
mprotect  
mq\_close  
mq\_getattr  
mq\_notify  
mq\_open  
mq\_receive  
mq\_send  
mq\_setattr  
mq\_timedreceive  
mq\_timedsend  
mq\_unlink  
rand48  
msgctl (see chapter "Implementation Notes")  
msgget (see chapter "Implementation Notes")  
msgrcv (see chapter "Implementation Notes")  
msgsnd (see chapter "Implementation Notes")  
msync  
munlock  
munmap  
nan  
nanf  
nanosleep  
nearbyint  
nearbyintf  
nextafter  
nextafterf  
nftw  
nice  
nl\_langinfo  
nrand48  
ntohl

ntohs  
open  
open\_memstream  
open\_wmemstream  
openat  
opendir  
openlog  
optarg  
opterr  
optind  
optopt  
pathconf  
pause  
pclose  
perror  
pipe  
poll  
popen  
posix\_fadvise  
posix\_fallocate  
posix\_madvise  
posix\_memalign  
posix\_openpt  
posix\_spawn  
posix\_spawnattr\_destroy  
posix\_spawnattr\_init  
posix\_spawnattr\_getflags  
posix\_spawnattr\_getpgroup  
posix\_spawnattr\_getschedparam  
posix\_spawnattr\_getschedpolicy  
posix\_spawnattr\_getsigdefault  
posix\_spawnattr\_getsigmask  
posix\_spawnattr\_setflags  
posix\_spawnattr\_setpgroup  
posix\_spawnattr\_setschedparam  
posix\_spawnattr\_setschedpolicy  
posix\_spawnattr\_setsigdefault  
posix\_spawnattr\_setsigmask  
posix\_spawnnp  
posix\_spawn\_file\_actions\_destroy  
posix\_spawn\_file\_actions\_init  
posix\_spawn\_file\_actions\_addclose  
posix\_spawn\_file\_actions\_adddup2  
posix\_spawn\_file\_actions\_addopen  
pow  
powf  
pread  
printf  
pselect  
psiginfo  
psignal  
pthread\_atfork  
pthread\_attr\_destroy  
pthread\_attr\_getdetachstate  
pthread\_attr\_getguardsize  
pthread\_attr\_getinheritsched  
pthread\_attr\_getschedparam  
pthread\_attr\_getschedpolicy  
pthread\_attr\_getscope  
pthread\_attr\_getstack  
pthread\_attr\_getstacksize  
pthread\_attr\_init  
pthread\_attr\_setdetachstate

pthread\_attr\_setguardsize  
pthread\_attr\_setinheritsched  
pthread\_attr\_setschedparam  
pthread\_attr\_setschedpolicy  
pthread\_attr\_setscope  
pthread\_attr\_setstack  
pthread\_attr\_setstacksize  
pthread\_cancel  
pthread\_cond\_broadcast  
pthread\_cond\_destroy  
pthread\_cond\_init  
pthread\_cond\_signal  
pthread\_cond\_timedwait  
pthread\_cond\_wait  
pthread\_condattr\_destroy  
pthread\_condattr\_getclock  
pthread\_condattr\_getpshared  
pthread\_condattr\_init  
pthread\_condattr\_setclock  
pthread\_condattr\_setpshared  
pthread\_create  
pthread\_detach  
pthread\_equal  
pthread\_exit  
pthread\_getconcurrency  
pthread\_getcpuclockid  
pthread\_getschedparam  
pthread\_getspecific  
pthread\_join  
pthread\_key\_create  
pthread\_key\_delete  
pthread\_kill  
pthread\_mutex\_destroy  
pthread\_mutex\_getprioceiling  
pthread\_mutex\_init  
pthread\_mutex\_lock  
pthread\_mutex\_setprioceiling  
pthread\_mutex\_trylock  
pthread\_mutex\_unlock  
pthread\_mutexattr\_destroy  
pthread\_mutexattr\_getprioceiling  
pthread\_mutexattr\_getprotocol  
pthread\_mutexattr\_getpshared  
pthread\_mutexattr\_gettype  
pthread\_mutexattr\_init  
pthread\_mutexattr\_setprioceiling  
pthread\_mutexattr\_setprotocol  
pthread\_mutexattr\_setpshared  
pthread\_mutexattr\_settype  
pthread\_once  
pthread\_rwlock\_destroy  
pthread\_rwlock\_init  
pthread\_rwlock\_rdlock  
pthread\_rwlock\_tryrdlock  
pthread\_rwlock\_trywrlock  
pthread\_rwlock\_unlock  
pthread\_rwlock\_wrlock  
pthread\_rwlockattr\_destroy  
pthread\_rwlockattr\_getpshared  
pthread\_rwlockattr\_init  
pthread\_rwlockattr\_setpshared  
pthread\_self  
pthread\_setcancelstate

pthread\_setcanceltype  
pthread\_setconcurrency  
pthread\_setschedparam  
pthread\_setschedprio  
pthread\_setspecific  
pthread\_sigmask  
pthread\_spin\_destroy  
pthread\_spin\_init  
pthread\_spin\_lock  
pthread\_spin\_trylock  
pthread\_spin\_unlock  
pthread\_testcancel  
ptsname  
putc  
putc\_unlocked  
putchar  
putchar\_unlocked  
putenv  
puts  
pututxline  
putwc  
putwchar  
pwrite  
qsort  
raise  
rand  
rand\_r  
random  
read  
readdir  
readdir\_r  
readlink  
readlinkat  
readv  
realloc  
realpath  
recv  
recvfrom  
recvmsg  
regcomp  
regerror  
regexec  
regfree  
remainder  
remainderf  
remove  
remque  
remquo  
remquoof  
rename  
renameat  
rewind  
rewinddir  
rint  
rintf  
rintl  
rmdir  
round  
roundf  
scalbln  
scalblnf  
scalbn  
scalbnf



scandir  
scanf  
sched\_get\_priority\_max  
sched\_get\_priority\_min  
sched\_getparam  
sched\_getscheduler  
sched\_rr\_get\_interval  
sched\_setparam  
sched\_setscheduler  
sched\_yield  
seed48  
seekdir  
select  
sem\_close  
sem\_destroy  
sem\_getvalue  
sem\_init  
sem\_open  
sem\_post  
sem\_timedwait  
sem\_trywait  
sem\_unlink  
sem\_wait  
semctl (see chapter "Implementation Notes")  
semget (see chapter "Implementation Notes")  
semop (see chapter "Implementation Notes")  
send  
sendmsg  
sendto  
setbuf  
setegid  
setenv  
seteuid  
setgid  
setgrent  
sethostent  
setitimer (see chapter "Implementation Notes")  
setjmp  
setkey (available in external "crypt" library)  
setlocale  
setlogmask  
setpgid  
setpgrp  
setpriority  
setprotoent  
setpwent  
setregid  
setreuid  
setrlimit  
setservent  
setsid  
setsockopt  
setstate  
setuid  
setutxent  
setvbuf  
shm\_open  
shm\_unlink  
shmat (see chapter "Implementation Notes")  
shmctl (see chapter "Implementation Notes")  
shmdt (see chapter "Implementation Notes")  
shmget (see chapter "Implementation Notes")  
shutdown

sigaction  
sigaddset  
sigdelset  
sigemptyset  
sigfillset  
sighold  
sigignore  
siginterrupt  
sigismember  
siglongjmp  
signal  
signbit (see chapter "Implementation Notes")  
signgam  
sigpause  
sigpending  
sigprocmask  
sigqueue  
sigrelse  
sigset  
sigsetjmp  
sigsuspend  
sigwait  
sigwaitinfo  
sin  
sinf  
sinh  
sinhf  
sleep  
snprintf  
socket  
socketpair  
sprintf  
sqrt  
sqrtf  
srand  
srand48  
srandom  
sscanf  
stat  
statvfs  
stderr  
stdin  
stdout  
stpcpy  
stpncpy  
strcasecmp  
strcat  
strchr  
strcmp  
strcoll  
strcpy  
strcspn  
strdup  
strerror  
strerror\_r  
strfmon  
strftime  
strlen  
strncasecmp  
strncat  
strncmp  
strncpy  
strndup

strnlen  
strpbrk  
strptime  
strrchr  
strsignal  
strspn  
strstr  
strtod  
strtof  
strtoimax  
strtok  
strtok\_r  
strtol  
strtoll  
strtoul  
strtoull  
strtoumax  
strxfrm  
swab  
swprintf  
swscanf  
symlink  
symlinkat  
sync  
sysconf  
syslog  
system  
tan  
tanh  
tanhf  
tcdrain  
tcflow  
tcflush  
tcgetattr  
tcgetpgrp  
tcsendbreak  
tcsetattr  
tcsetpgrp  
tdelete  
telldir  
tempnam  
tfind  
tgamma  
tgammaf  
time  
timer\_create (see chapter "Implementation Notes")  
timer\_delete  
timer\_gettime  
timer\_settime  
times  
timezone  
tmpfile  
tmpnam  
toascii  
tolower  
toupper  
towctrans  
towlower  
towupper  
trunc  
truncate  
truncf

tsearch  
ttyname  
ttyname\_r  
twalk  
tzname  
tzset  
umask  
uname  
ungetc  
ungetwc  
unlink  
unlinkat  
unlockpt  
unsetenv  
utime  
utimensat  
utimes  
va\_arg  
va\_copy  
va\_end  
va\_start  
vdprintf  
vfprintf  
vfprintf  
vfwprintf  
vfwscanf  
vprintf  
vscanf  
vsnprintf  
vsprintf  
vsscanf  
vswprintf  
vswscanf  
vwprintf  
vwscanf  
wait  
waitpid  
wcpncpy  
wcpncpy  
wcrtoomb  
wcscasecmp  
wcscat  
wcschr  
wcscmp  
wcscoll  
wcscpy  
wcscspn  
wcsdup  
wcsftime  
wcslen  
wcsncasecmp  
wcsncat  
wcsncmp  
wcsncpy  
wcsnlen  
wcsnrtombs  
wcpbrk  
wcsrchr  
wcsrtombs  
wcssp  
wcstr  
wcstod  
wcstof

wcstoimax  
wcstok  
wcstol  
wcstoll  
wcstombs  
wcstoul  
wcstoull  
wcstoumax  
wcswidth  
wcsxfrm  
wctob  
wctomb  
wctrans  
wctype  
wcwidth  
wmemchr  
wmemcmp  
wmemcpy  
wmemmove  
wmemset  
wordexp  
wordfree  
wprintf  
write  
writev  
wscanf  
y0  
y1  
yn

## System interfaces compatible with BSD functions:

```
__b64_ntop
__b64_pton
arc4random
arc4random_addrandom
arc4random_buf
arc4random_stir
arc4random_uniform
bindresvport
bindresvport_sa
cfmakeraw
cfsetspeed
daemon
dn_comp
dn_expand
dn_skipname
drem
eaccess
endusershell
err
errx
finite
finitef
fiprintf
flock    (see chapter "Implementation Notes")
forkpty
fpurge
freeifaddrs
fstatfs
fts_children
fts_close
fts_get_clientptr
fts_get_stream
fts_open
fts_read
fts_set
fts_set_clientptr
funopen
futimes
gamma
gamma_r
gammaf
gammaf_r
getdtablesize
getgrouplist
getifaddrs
getpagesize
getpeereid
getprogname
getusershell
herror
hstrerror
inet_aton
inet_makeaddr
inet_netof
inet_network
initgroups
```

iruserok  
iruserok\_sa  
login  
login\_tty  
logout  
logwtmp  
madvise  
mkstemp  
openpty  
rcmd  
rcmd\_af  
reallocf  
res\_close  
res\_init  
res\_mkquery  
res\_nclose  
res\_ninit  
res\_nmkquery  
res\_nquery  
res\_nquerydomain  
res\_nsearch  
res\_nsend  
res\_query  
res\_querydomain  
res\_search  
res\_send  
revoke  
rexec  
rresvport  
rresvport\_af  
ruserok  
sbrk  
setbuffer  
setgroups  
setlinebuf  
setpassent  
setprogname  
settimeofday  
setusershell  
statfs  
strcasestr  
strlcat  
strlcpy  
strsep  
updwtmp  
valloc  
verr  
verrx  
vhangup (see chapter "Implementation Notes")  
vsyslog  
vwarn  
vwarnx  
wait3  
wait4  
warn  
warnx  
wcslcat  
wcslcpy

## System interfaces compatible with GNU or Linux extensions:

```
accept4
argz_add
argz_add_sep
argz_append
argz_count
argz_create
argz_create_sep
argz_delete
argz_extract
argz_insert
argz_next
argz_replace
argz_stringify
asnprintf
asprintf
asprintf_r
canonicalize_file_name
dremf
dup3
envz_add
envz_entry
envz_get
envz_merge
envz_remove
envz_strip
error
error_at_line
euidaccess
execvpe
expl0
expl0f
fcloseall
fcloseall_r
fegetprec
fesetprec
feenableexcept
fedisableexcept
fegetexcept
ffsl
ffsll
fgetxattr
flistxattr
fopencookie
fremovexattr
fsetxattr
get_avphys_pages
get_current_dir_name
get_phys_pages
get_nprocs
get_nprocs_conf
getmntent_r
getopt_long
getopt_long_only
getpt
getxattr
lgetxattr
```



listxattr  
llistxattr  
lremovexattr  
lsetxattr  
memmem  
mempcpy  
memrchr  
mkostemp  
mkostemps  
pipe2  
powl0  
powl0f  
ppoll  
pthread\_getattr\_np  
pthread\_sigqueue  
ptsname\_r  
quotactl  
rawmemchr  
removexattr  
scandirat  
setxattr  
strchrnul  
sysinfo  
tdestroy  
timegm  
timelocal  
updwtmpx  
utmpxname  
vasnprintf  
vasprintf  
vasprintf\_r

## System interfaces compatible with Solaris or SunOS functions:

```
__fpurge
acl
aclcheck
aclfrommode
aclfrompbits
aclfromtext
aclsort
acltomode
acltopbits
acltotext
endmntent
facl
futimesat
getmntent
memalign
setmntent
xdr_array
xdr_bool
xdr_bytes
xdr_char
xdr_double
xdr_enum
xdr_float
xdr_free
xdr_hyper
xdr_int
xdr_int16_t
xdr_int32_t
xdr_int64_t
xdr_int8_t
xdr_long
xdr_longlong_t
xdr_netobj
xdr_opaque
xdr_pointer
xdr_reference
xdr_short
xdr_sizeof
xdr_string
xdr_u_char
xdr_u_hyper
xdr_u_int
xdr_u_int16_t
xdr_u_int32_t
xdr_u_int64_t
xdr_u_int8_t
xdr_u_long
xdr_u_longlong_t
xdr_u_short
xdr_uint16_t
xdr_uint32_t
xdr_uint64_t
xdr_uint8_t
xdr_union
xdr_vector
xdr_void
```

xdr\_wrapstring  
xdrmem\_create  
xdrrec\_create  
xdrrec\_endofrecord  
xdrrec\_eof  
xdrrec\_skiprecord  
\_\_xdrrec\_getrec  
\_\_xdrrec\_setnonblock  
xdrstdio\_create

## Other UNIX system interfaces, deprecated or not in POSIX.1-2008:

```
bcmp      (POSIX.1-2001, SUSv3)
bcopy     (SUSv3)
bzero     (SUSv3)
chroot    (SUSv2) (see chapter "Implementation Notes")
clock_setres (QNX, VxWorks) (see chapter "Implementation Notes")
cuserid   (POSIX.1-1988, SUSv2)
ecvt      (SUSv3)
endutent  (XPG2)
fcvt      (SUSv3)
ftime     (SUSv3)
gcv       (SUSv3)
gethostbyaddr (SUSv3)
gethostbyname (SUSv3)
gethostbyname2 (first defined in BIND 4.9.4)
getpass   (SUSv2)
getutent  (XPG2)
getutid   (XPG2)
getutline (XPG2)
getw      (SVID)
getwd     (SUSv3)
h_errno   (SUSv3)
index     (SUSv3)
mallinfo  (SVID)
mallopt   (SVID)
mktemp    (SUSv3)
on_exit   (SunOS)
pthread_attr_getstackaddr (SUSv3)
pthread_attr_setstackaddr (SUSv3)
pthread_continue (XPG2)
pthread_getsequence_np (Tru64)
pthread_suspend (XPG2)
pthread_yield (POSIX.1c drafts)
pututline (XPG2)
putw      (SVID)
rindex    (SUSv3)
scalb     (SUSv3)
setutent  (XPG2)
stime     (SVID)
sys_errlist (BSD)
sys_nerr   (BSD)
sys_siglist (BSD)
ttyslot   (SUSv2)
ualarm    (SUSv3)
usleep    (SUSv3)
utmpname  (XPG2)
vfork     (SUSv3) (see chapter "Implementation Notes")
```

## NOT implemented system interfaces from the Single Unix Specification, Volume 4:

acoshl  
acosl  
aio\_cancel  
aio\_error  
aio\_fsync  
aio\_read  
aio\_return  
aio\_suspend  
aio\_write  
asinh1  
asinl  
atan2l  
atanhl  
atanl  
cabs1  
cacosh1  
cacosl  
carg1  
casinl  
catanhl  
catanl  
cbrtl  
ccosh1  
ccosl  
ceil1  
cexpl  
cimagl  
clogl  
conj1  
copysign1  
cosh1  
cosl  
cpowl  
cproj1  
creall  
csinh1  
csinl  
csqrt1  
ctanhl  
ctanl  
duplocale  
endnetent  
erfcl  
erfl  
exp2l  
expl  
expml1  
fabs1  
fattach  
fdiml  
floor1  
fmal  
fmaxl  
fminl  
fmodl  
fmtmsg

freelocale  
frexpl  
getdate  
getdate\_err  
gethostent  
getmsg  
getnetbyaddr  
getnetbyname  
getnetent  
getpmsg  
hypotl  
ilogbl  
isalnum\_l  
isalpha\_l  
isastream  
isblank\_l  
iscntrl\_l  
isdigit\_l  
isgraph\_l  
islower\_l  
isprint\_l  
ispunct\_l  
isspace\_l  
isupper\_l  
iswalnum\_l  
iswalpha\_l  
iswblank\_l  
iswcntrl\_l  
iswdigit\_l  
iswgraph\_l  
iswlower\_l  
iswprint\_l  
iswpunct\_l  
iswspace\_l  
iswupper\_l  
iswxdigit\_l  
isxdigit\_l  
ldexpl  
lgammal  
lio\_listio  
llroundl  
log10l  
log1pl  
log2l  
logbl  
logl  
lroundl  
mlockall  
modfl  
munlockall  
nanl  
nearbyintl  
newlocale  
nextafterl  
nexttoward  
nexttowardf  
nexttowardl  
posix\_mem\_offset  
posix\_trace[...]  
posix\_typed[...]  
powl  
pthread\_barrier[...]  
pthread\_mutexattr\_getrobust

pthread\_mutexattr\_setrobust  
pthread\_mutex\_consistent  
pthread\_mutex\_timedlock  
pthread\_rwlock\_timedrdlock  
pthread\_rwlock\_timedwrlock  
putmsg  
reminder1  
remquol  
roundl  
scalblnl  
scalbnl  
setnetent  
sigaltstack  
sigtimedwait  
sinhl  
sinl  
socketmark  
sqrtl  
strcasecmp\_l  
strcoll\_l  
strfmon\_l  
strncasecmp\_l  
strtold  
strxfrm\_l  
tanhl  
tanl  
tcgetsid  
tgammal  
timer\_getoverrun  
tolower\_l  
toupper\_l  
towctrans\_l  
trunc\_l  
ulimit  
uselocale  
waitid  
wcscasecmp\_l  
wcsncasecmp\_l  
wcstold  
wcsxfrm\_l  
wctrans\_l  
wctype\_l

# Implementation Notes

`chroot` only emulates a `chroot` function call by keeping track of the current root and accomodating this in the file related function calls. A real `chroot` functionality is not supported by Windows however.

`clock_nanosleep` currently supports only `CLOCK_REALTIME` and `CLOCK_MONOTONIC`. `clock_setres`, `clock_settime`, and `timer_create` currently support only `CLOCK_REALTIME`.

POSIX file locks via `fcntl` or `lockf`, as well as BSD `flock` locks are advisory locks. They don't interact with Windows mandatory locks, nor do POSIX `fcntl` locks interfere with BSD `flock` locks or vice versa.

BSD file locks created via `flock` are only propagated to the direct parent process, not to grand parents or sibling processes. The locks are only valid in the creating process, its parent process, and subsequently started child processes sharing the same file descriptor.

In very rare circumstances an application would want to use Windows mandatory locks to interact with non-Cygwin Windows processes accessing the same file (databases, etc). For these purposes, the entire locking mechanism (`fcntl/flock/lockf`) can be switched to Windows mandatory locks on a per-descriptor/per-process basis. For this purpose, use the call

```
fcntl (fd, F_LCK_MANDATORY, 1);
```

After that, all file locks on this descriptor will follow Windows mandatory record locking semantics: Locks are per-descriptor/per-process; locks are not propagated to child processes, not even via `execve`; no atomic replacement of read locks with write locks and vice versa on the same descriptor; locks have to be unlocked exactly as they have been locked.

`fpclassify`, `isfinite`, `isgreater`, `isgreaterequal`, `isinf`, `isless`, `islessequal`, `islessgreater`, `isnan`, `isnormal`, `isunordered`, and `signbit` only support float and double arguments, not long double arguments.

`getitimer` and `setitimer` only support `ITIMER_REAL` for now.

`link` will fail on FAT, FAT32, and other filesystems not supporting hardlinks, just as on Linux.

`lseek` only works properly on files opened in binary mode. On files opened in textmode (via mount mode or explicit open flag) its positioning is potentially unreliable.

`setuid` is only safe against reverting the user switch after a call to one of the `exec(2)` functions took place. Windows doesn't support a non-revertable user switch within the context of Win32 processes.

`vfork` just calls `fork`.

`vhangup` and `revoke` always return -1 and set `errno` to `ENOSYS`. `grantpt` and `unlockpt` always just return 0.

The XSI IPC functions `semctl`, `semget`, `semop`, `shmat`, `shmctl`, `shmdt`, `shmget`, `msgctl`, `msgget`, `msgrcv` and `msgsnd` are only available when `cygserver` is running.

The Linux-specific function `quotactl` only implements what works on Windows: Windows only supports user block quotas on NTFS, no group quotas, no inode quotas, no time constraints.



---

## Chapter 2. Cygwin Functions

## Path conversion functions

These functions are specific to Cygwin itself, and probably won't be found anywhere else.

### cygwin\_conv\_path

```
extern "C" ssize_t cygwin_conv_path(what, from, to, size);

cygwin_conv_path_t what;
const void * from;
void * to;
size_t size;
```

Use this function to convert POSIX paths in *from* to Win32 paths in *to* or, vice versa, Win32 paths in *from* to POSIX paths in *to*. *what* defines the direction of this conversion and can be any of the below values.

```
CCP_POSIX_TO_WIN_A    /* from is char *posix, to is char *win32      */
CCP_POSIX_TO_WIN_W,   /* from is char *posix, to is wchar_t *win32    */
CCP_WIN_A_TO_POSIX,   /* from is char *win32, to is char *posix        */
CCP_WIN_W_TO_POSIX,   /* from is wchar_t *win32, to is char *posix     */
```

You can additionally or the following values to *what*, to define whether you want the resulting path in *to* to be absolute or if you want to keep relative paths in relative notation. Creating absolute paths is the default.

```
CCP_ABSOLUTE = 0,      /* Request absolute path (default).              */
CCP_RELATIVE = 0x100   /* Request to keep path relative.                 */
```

*size* is the size of the buffer pointed to by *to* in bytes. If *size* is 0, *cygwin\_conv\_path* just returns the required buffer size in bytes. Otherwise, it returns 0 on success, or -1 on error and *errno* is set to one of the below values.

```
EINVAL      what has an invalid value or from is NULL.
EFAULT      from or to point into nirvana.
ENAMETOOLONG the resulting path is longer than 32K, or, in case
             of what == CCP_POSIX_TO_WIN_A, longer than MAX_PATH.
ENOSPC      size is less than required for the conversion.
```

#### Example 2.1. Example use of *cygwin\_conv\_path*

```
#include <sys/cygwin.h>

/* Conversion from incoming Win32 path given as wchar_t *win32 to POSIX path.
   If incoming path is a relative path, stick to it. First ask how big
   the output buffer has to be and allocate space dynamically. */
ssize_t size;
char *posix;
size = cygwin_conv_path (CCP_WIN_W_TO_POSIX | CCP_RELATIVE, win32, NULL, 0);
if (size < 0)
    perror ("cygwin_conv_path");
else
{
    posix = (char *) malloc (size);
```

```
if (cygwin_conv_path (CCP_WIN_W_TO_POSIX | CCP_RELATIVE, win32,
                    posix, size))
    perror ("cygwin_conv_path");
}
```

## cygwin\_conv\_path\_list

```
extern "C" ssize_t cygwin_conv_path_list(what, from, to, size);

cygwin_conv_path_t what;
const void * from;
void * to;
size_t size;
```

This is the same as `cygwin_conv_path`, but the input is treated as a path list in `$PATH` or `%PATH` notation.

If *what* is `CCP_POSIX_TO_WIN_A` or `CCP_POSIX_TO_WIN_W`, given a POSIX `$PATH`-style string (i.e. `/foo:/bar`) convert it to the equivalent Win32 `%PATH`-style string (i.e. `d:\e:\bar`).

If *what* is `CCP_WIN_A_TO_POSIX` or `CCP_WIN_W_TO_POSIX`, given a Win32 `%PATH`-style string (i.e. `d:\e:\bar`) convert it to the equivalent POSIX `$PATH`-style string (i.e. `/foo:/bar`).

*size* is the size of the buffer pointed to by *to* in bytes.

See also `cygwin_conv_path`

## cygwin\_create\_path

```
extern "C" void * cygwin_create_path(what, from);

cygwin_conv_path_t what;
const void * from;
```

This is equivalent to the `cygwin_conv_path`, except that `cygwin_create_path` does not take a buffer pointer for the result of the conversion as input. Rather it allocates the buffer itself using `malloc(3)` and returns a pointer to this buffer. In case of error it returns `NULL` and sets `errno` to one of the values defined for `cygwin_conv_path`. Additionally `errno` can be set to the below value.

<code>ENOMEM</code>	Insufficient memory was available.
---------------------	------------------------------------

When you don't need the returned buffer anymore, use `free(3)` to deallocate it.

See also `cygwin_conv_path`

## cygwin\_posix\_path\_list\_p

```
extern "C" int cygwin_posix_path_list_p(path);

const char *path;
```

This function tells you if the supplied *path* is a POSIX-style path (i.e. posix names, forward slashes, colon delimiters) or a Win32-style path (drive letters, reverse slashes, semicolon delimiters). The return value is true if the path is a POSIX path. Note that `"_p"` means "predicate", a lisp term meaning that the function tells you something about the parameter.

## cygwin\_split\_path

```
extern "C" void cygwin_split_path (path, dir, file);
```

```
const char * path;  
char * dir;  
char * file;
```

Split a path into the directory and the file portions. Both *dir* and *file* are expected to point to buffers of sufficient size.

### Example 2.2. Example use of `cygwin_split_path`

```
char dir[200], file[100];  
cygwin_split_path("c:/foo/bar.c", dir, file);  
printf("dir=%s, file=%s\n", dir, file);
```

# Helper functions to change user context

## cygwin\_logon\_user

```
extern "C" HANDLE cygwin_logon_user(passwd_entry, password);  
  
const struct passwd *passwd_entry;  
const char *password;
```

Given a pointer to a passwd entry of a user and a cleartext password, returns a HANDLE to an impersonation token for this user which can be used in a subsequent call to `cygwin_set_impersonation_token` to impersonate that user. This function can only be called from a process which has the required NT user rights to perform a logon.

See also the chapter Switching the user context [[../cygwin-ug-net/ntsec.html#ntsec-setuid-overview](http://cygwin-ug-net/ntsec.html#ntsec-setuid-overview)] in the Cygwin User's guide.

See also `cygwin_set_impersonation_token`

## cygwin\_set\_impersonation\_token

```
extern "C" void cygwin_set_impersonation_token(token);  
  
const HANDLE token;
```

Use this function to enable the token given as parameter as impersonation token for the next call to `setuid` or `seteuid`. Use `cygwin_set_impersonation_token` together with `cygwin_logon_user` to impersonate users using password authentication.

See also the chapter Switching the user context [[../cygwin-ug-net/ntsec.html#ntsec-setuid-overview](http://cygwin-ug-net/ntsec.html#ntsec-setuid-overview)] in the Cygwin User's guide.

See also `cygwin_logon_user`

## Miscellaneous functions

### cygwin\_attach\_handle\_to\_fd

```
extern "C" int cygwin_attach_handle_to_fd(name, fd, handle, bin,
access);

char *name;
int fd;
HANDLE handle;
int bin;
int access;
```

This function can be used to turn a Win32 "handle" into a posix-style file handle. *fd* may be -1 to make cygwin allocate a handle; the actual handle is returned in all cases.

Even after using function, Cygwin doesn't know anything about the underlying file or device. It just tries to supply the typical file functions on a "best-effort" basis. Use with care. Don't expect too much.

### cygwin\_internal

```
extern "C" DWORD cygwin_internal(t, ...);

cygwin_getinfo_types t;
...;
```

This function gives you access to various internal data and functions. It takes two arguments. The first argument is a type from the 'cygwin\_getinfo\_types' enum. The second is an optional pointer.

Stay away unless you know what you're doing.

### cygwin\_stackdump

```
extern "C" void cygwin_stackdump();
```

Outputs a stackdump to stderr from the called location.