NAME

perlcygwin - Perl for Cygwin

SYNOPSIS

This document will help you configure, make, test and install Perl on Cygwin. This document also describes features of Cygwin that will affect how Perl behaves at runtime.

NOTE: There are pre-built Perl packages available for Cygwin and a version of Perl is provided in the normal Cygwin install. If you do not need to customize the configuration, consider using one of those packages.

PREREQUISITES FOR COMPILING PERL ON CYGWIN

Cygwin = GNU+Cygnus+Windows (Don't leave UNIX without it)

The Cygwin tools are ports of the popular GNU development tools for Win32 platforms. They run thanks to the Cygwin library which provides the UNIX system calls and environment these programs expect. More information about this project can be found at:

<https://www.cygwin.com/>

A recent net or commercial release of Cygwin is required.

At the time this document was last updated, Cygwin 1.7.16 was current.

Cygwin Configuration

While building Perl some changes may be necessary to your Cygwin setup so that Perl builds cleanly. These changes are **not** required for normal Perl usage.

NOTE: The binaries that are built will run on all Win32 versions. They do not depend on your host system (WinXP/Win2K/Win7) or your Cygwin configuration (binary/text mounts, cvgserver). The only dependencies come from hard-coded pathnames like */usr/local*. However, your host system and Cygwin configuration will affect Perl's runtime behavior (see "TEST").

- "PATH"

  Set the "PATH" environment variable so that Configure finds the Cygwin versions of programs. Any not-needed Windows directories should be removed or moved to the end of your "PATH".

- *nroff*

  If you do not have *nroff* (which is part of the *groff* package), Configure will **not** prompt you to

install *man* pages.

CONFIGURE PERL ON CYGWIN

The default options gathered by Configure with the assistance of *hints/cygwin.sh* will build a Perl that supports dynamic loading (which requires a shared *cygperl5_16.dll*).

This will run Configure and keep a record:

```
./Configure 2>&1 | tee log.configure
```

If you are willing to accept all the defaults run Configure with **-de**. However, several useful customizations are available.

Stripping Perl Binaries on Cygwin

It is possible to strip the EXEs and DLLs created by the build process. The resulting binaries will be significantly smaller. If you want the binaries to be stripped, you can either add a **-s** option when Configure prompts you,

```
Any additional ld flags (NOT including libraries)? [none] -s
Any special flags to pass to g++ to create a dynamically loaded
library?
[none] -s
Any special flags to pass to gcc to use dynamic linking? [none] -s
```

or you can edit *hints/cygwin.sh* and uncomment the relevant variables near the end of the file.

Optional Libraries for Perl on Cygwin

Several Perl functions and modules depend on the existence of some optional libraries. Configure will find them if they are installed in one of the directories listed as being used for library searches. Pre-built packages for most of these are available from the Cygwin installer.

- "-lcrypt"

  The crypt package distributed with Cygwin is a Linux compatible 56-bit DES crypt port by Corinna Vinschen.

  Alternatively, the crypt libraries in GNU libc have been ported to Cygwin.

  As of libcrypt 1.3 (March 2016), you will need to install the libcrypt-devel package for Configure to detect **crypt()**.

- "-lgdbm_compat" ("use GDBM_File")

  GDBM is available for Cygwin.

  NOTE: The GDBM library only works on NTFS partitions.

- "-ldb" ("use DB_File")

  BerkeleyDB is available for Cygwin.

  NOTE: The BerkeleyDB library only completely works on NTFS partitions.

- "cygserver" ("use IPC::SysV")

  A port of SysV IPC is available for Cygwin.

  NOTE: This has **not** been extensively tested. In particular, "d_semctl_semun" is undefined because it fails a Configure test and on Win9x the *shm*()* functions seem to hang. It also creates a compile time dependency because *perl.h* includes <*sys/ipc.h*> and <*sys/sem.h*> (which will be required in the future when compiling CPAN modules). CURRENTLY NOT SUPPORTED!

- "-lutil"

  Included with the standard Cygwin netrelease is the inetutils package which includes libutil.a.

Configure-time Options for Perl on Cygwin

The *INSTALL* document describes several Configure-time options. Some of these will work with Cygwin, others are not yet possible. Also, some of these are experimental. You can either select an option when Configure prompts you or you can define (undefine) symbols on the command line.

- "-Uusedl"

  Undefining this symbol forces Perl to be compiled statically.

- "-Dusemymalloc"

  By default Perl does not use the "malloc()" included with the Perl source, because it was slower and not entirely thread-safe. If you want to force Perl to build with the old -Dusemymalloc define this.

- "-Uuseperlio"

  Undefining this symbol disables the PerlIO abstraction. PerlIO is now the default; it is not recommended to disable PerlIO.

- "-Dusemultiplicity"

  Multiplicity is required when embedding Perl in a C program and using more than one interpreter instance. This is only required when you build a not-threaded perl with "-Uuseithreads".

- "-Uuse64bitint"

  By default Perl uses 64 bit integers. If you want to use smaller 32 bit integers, define this symbol.

- "-Duselongdouble"

  gcc supports long doubles (12 bytes). However, several additional long double math functions are necessary to use them within Perl (*{atan2, cos, exp, floor, fmod, frexp, isnan, log, modf, pow, sin, sqrt}l, strtold*). These are **not** yet available with newlib, the Cygwin libc.

- "-Uuseithreads"

  Define this symbol if you want not-threaded faster perl.

- "-Duselargefiles"

  Cygwin uses 64-bit integers for internal size and position calculations, this will be correctly detected and defined by Configure.

- "-Dmksymlinks"

  Use this to build perl outside of the source tree. Details can be found in the *INSTALL* document. This is the recommended way to build perl from sources.

Suspicious Warnings on Cygwin

You may see some messages during Configure that seem suspicious.

- Win9x and "d_eofnblk"

  Win9x does not correctly report "EOF" with a non-blocking read on a closed pipe. You will see

the following messages:

```
But it also returns -1 to signal EOF, so be careful!
WARNING: you can't distinguish between EOF and no data!

*** WHOA THERE!!! ***
    The recommended value for $d_eofnblk on this machine was
    "define"!
    Keep the recommended value? [y]
```

At least for consistency with WinNT, you should keep the recommended value.

- Compiler/Preprocessor defines

  The following error occurs because of the Cygwin "#define" of "_LONG_DOUBLE":

```
Guessing which symbols your C compiler and preprocessor define...
try.c:<line#>: missing binary operator
```

  This failure does not seem to cause any problems. With older gcc versions, "parse error" is reported instead of "missing binary operator".

MAKE ON CYGWIN

Simply run *make* and wait:

```
make 2>&1 | tee log.make
```

TEST ON CYGWIN

There are two steps to running the test suite:

```
make test 2>&1 | tee log.make-test

cd t; ./perl harness 2>&1 | tee ../log.harness
```

The same tests are run both times, but more information is provided when running as "./perl harness".

Test results vary depending on your host system and your Cygwin configuration. If a test can pass in some Cygwin setup, it is always attempted and explainable test failures are documented. It is possible for Perl to pass all the tests, but it is more likely that some tests will fail for one of the reasons listed below.

File Permissions on Cygwin

UNIX file permissions are based on sets of mode bits for {read,write,execute} for each {user,group,other}. By default Cygwin only tracks the Win32 read-only attribute represented as the UNIX file user write bit (files are always readable, files are executable if they have a *.{com,bat,exe}* extension or begin with "#!", directories are always readable and executable). On WinNT with the *ntea* "CYGWIN" setting, the additional mode bits are stored as extended file attributes. On WinNT with the default *ntsec* "CYGWIN" setting, permissions use the standard WinNT security descriptors and access control lists. Without one of these options, these tests will fail (listing not updated yet):

Failed Test	List of failed
io/fs.t	5, 7, 9-10
lib/anydbm.t	2
lib/db-btree.t	20
lib/db-hash.t	16
lib/db-recno.t	18
lib/gdbm.t	2
lib/ndbm.t	2
lib/odbm.t	2
lib/sdbm.t	2
op/stat.t	9, 20 (.tmp not an executable extension)

NDBM_File and ODBM_File do not work on FAT filesystems

Do not use NDBM_File or ODBM_File on FAT filesystem. They can be built on a FAT filesystem, but many tests will fail:

```
../ext/NDBM_File/ndbm.t       13  3328    71   59  83.10%  1-2 4 16-71
../ext/ODBM_File/odbm.t      255 65280    ??   ??       %  ??
../lib/AnyDBM_File.t           2   512    12    2  16.67%  1 4
../lib/Memoize/t/errors.t      0   139    11    5  45.45%  7-11
../lib/Memoize/t/tie_ndbm.t   13  3328     4    4 100.00%  1-4
run/fresh_perl.t                     97    1   1.03%  91
```

If you intend to run only on FAT (or if using AnyDBM_File on FAT), run Configure with the -Ui_ndbm and -Ui_dbm options to prevent NDBM_File and ODBM_File being built.

With NTFS (and no CYGWIN=nontsec), there should be no problems even if perl was built on FAT.

"fork()" failures in io_* tests

A "fork()" failure may result in the following tests failing:

ext/IO/lib/IO/t/io_multihomed.t
ext/IO/lib/IO/t/io_sock.t
ext/IO/lib/IO/t/io_unix.t

See comment on fork in "Miscellaneous" below.

Specific features of the Cygwin port

Script Portability on Cygwin

Cygwin does an outstanding job of providing UNIX-like semantics on top of Win32 systems. However, in addition to the items noted above, there are some differences that you should know about. This is a very brief guide to portability, more information can be found in the Cygwin documentation.

- Pathnames

  Cygwin pathnames are separated by forward (*/*) slashes, Universal Naming Codes (*//UNC*) are also supported Since cygwin-1.7 non-POSIX pathnames are discouraged. Names may contain all printable characters.

  File names are case insensitive, but case preserving. A pathname that contains a backslash or drive letter is a Win32 pathname, and not subject to the translations applied to POSIX style pathnames, but cygwin will warn you, so better convert them to POSIX.

  For conversion we have "Cygwin::win_to_posix_path()" and "Cygwin::posix_to_win_path()".

  Since cygwin-1.7 pathnames are UTF-8 encoded.

- Text/Binary

  Since cygwin-1.7 textmounts are deprecated and strongly discouraged.

  When a file is opened it is in either text or binary mode. In text mode a file is subject to CR/LF/Ctrl-Z translations. With Cygwin, the default mode for an "open()" is determined by the mode of the mount that underlies the file. See "Cygwin::is_binmount"(). Perl provides a "binmode()" function to set binary mode on files that otherwise would be treated as text. "sysopen()" with the "O_TEXT" flag sets text mode on files that otherwise would be treated as binary:

  ```
  sysopen(FOO, "bar", O_WRONLY|O_CREAT|O_TEXT)
  ```

  "lseek()", "tell()" and "sysseek()" only work with files opened in binary mode.

The text/binary issue is covered at length in the Cygwin documentation.

- PerlIO

  PerlIO overrides the default Cygwin Text/Binary behaviour. A file will always be treated as binary, regardless of the mode of the mount it lives on, just like it is in UNIX. So CR/LF translation needs to be requested in either the "open()" call like this:

  ```
  open(FH, ">:crlf", "out.txt");
  ```

  which will do conversion from LF to CR/LF on the output, or in the environment settings (add this to your .bashrc):

  ```
  export PERLIO=crlf
  ```

  which will pull in the crlf PerlIO layer which does LF -> CRLF conversion on every output generated by perl.

- *.exe*

  The Cygwin "stat()", "lstat()" and "readlink()" functions make the *.exe* extension transparent by looking for *foo.exe* when you ask for *foo* (unless a *foo* also exists). Cygwin does not require a *.exe* extension, but *gcc* adds it automatically when building a program. However, when accessing an executable as a normal file (e.g., *cp* in a makefile) the *.exe* is not transparent. The *install* program included with Cygwin automatically appends a *.exe* when necessary.

- Cygwin vs. Windows process ids

  Cygwin processes have their own pid, which is different from the underlying windows pid. Most posix compliant Proc functions expect the cygwin pid, but several Win32::Process functions expect the winpid. E.g. $$ is the cygwin pid of */usr/bin/perl*, which is not the winpid. Use "Cygwin::pid_to_winpid()" and "Cygwin::winpid_to_pid()" to translate between them.

- Cygwin vs. Windows errors

  Under Cygwin, $^E is the same as $!. When using Win32 API Functions, use "Win32::GetLastError()" to get the last Windows error.

- rebase errors on fork or system

Using "fork()" or "system()" out to another perl after loading multiple dlls may result on a DLL baseaddress conflict. The internal cygwin error looks like like the following:

```
 0 [main] perl 8916 child_info_fork::abort: data segment start:
 parent (0xC1A000) != child(0xA6A000)
```

or:

```
 183 [main] perl 3588 C:\cygwin\bin\perl.exe: *** fatal error -
 unable to remap C:\cygwin\bin\cygsvn_subr-1-0.dll to same address
 as parent(0x6FB30000) != 0x6FE60000 46 [main] perl 3488 fork: child
 3588 - died waiting for dll loading, errno11
```

See <https://cygwin.com/faq/faq-nochunks.html#faq.using.fixing-fork-failures> It helps if not too many DLLs are loaded in memory so the available address space is larger, e.g. stopping the MS Internet Explorer might help.

Use the perlrebase or rebase utilities to resolve the conflicting dll addresses. The rebase package is included in the Cygwin setup. Use *setup.exe* from <https://cygwin.com/install.html> to install it.

1. kill all perl processes and run "perlrebase" or

2. kill all cygwin processes and services, start dash from cmd.exe and run "rebaseall".

- "chown()"

  On WinNT "chown()" can change a file's user and group IDs. On Win9x "chown()" is a no-op, although this is appropriate since there is no security model.

- Miscellaneous

  File locking using the "F_GETLK" command to "fcntl()" is a stub that returns "ENOSYS".

  Win9x can not "rename()" an open file (although WinNT can).

  The Cygwin "chroot()" implementation has holes (it can not restrict file access by native Win32 programs).

  Inplace editing "perl -i" of files doesn't work without doing a backup of the file being edited "perl

-i.bak" because of windowish restrictions, therefore Perl adds the suffix ".bak" automatically if you use "perl -i" without specifying a backup extension.

Prebuilt methods:

"Cwd::cwd"

Returns the current working directory.

"Cygwin::pid_to_winpid"

Translates a cygwin pid to the corresponding Windows pid (which may or may not be the same).

"Cygwin::winpid_to_pid"

Translates a Windows pid to the corresponding cygwin pid (if any).

"Cygwin::win_to_posix_path"

Translates a Windows path to the corresponding cygwin path respecting the current mount points. With a second non-null argument returns an absolute path. Double-byte characters will not be translated.

"Cygwin::posix_to_win_path"

Translates a cygwin path to the corresponding cygwin path respecting the current mount points. With a second non-null argument returns an absolute path. Double-byte characters will not be translated.

"Cygwin::mount_table()"

Returns an array of [mnt_dir, mnt_fsname, mnt_type, mnt_opts].

```
perl -e 'for $i (Cygwin::mount_table) {print join(" ",@$i),"\n";}'
/bin c:\cygwin\bin system binmode,cygexec
/usr/bin c:\cygwin\bin system binmode
/usr/lib c:\cygwin\lib system binmode
/ c:\cygwin system binmode
/cygdrive/c c: system binmode,noumount
/cygdrive/d d: system binmode,noumount
/cygdrive/e e: system binmode,noumount
```

"Cygwin::mount_flags"

Returns the mount type and flags for a specified mount point. A comma-separated string of mntent->mnt_type (always "system" or "user"), then the mntent->mnt_opts, where the first is always "binmode" or "textmode".

```
system|user,binmode|textmode,exec,cygexec,cygdrive,mixed,
notexec,managed,nosuid,devfs,proc,noumount
```

If the argument is "/cygdrive", then just the volume mount settings, and the cygdrive mount prefix are returned.

User mounts override system mounts.

```
$ perl -e 'print Cygwin::mount_flags "/usr/bin"'
system,binmode,cygexec
$ perl -e 'print Cygwin::mount_flags "/cygdrive"'
binmode,cygdrive,/cygdrive
```

"Cygwin::is_binmount"

Returns true if the given cygwin path is binary mounted, false if the path is mounted in textmode.

"Cygwin::sync_winenv"

Cygwin does not initialize all original Win32 environment variables. See the bottom of this page <https://cygwin.com/cygwin-ug-net/setup-env.html> for "Restricted Win32 environment".

Certain Win32 programs called from cygwin programs might need some environment variable, such as e.g. ADODB needs %COMMONPROGRAMFILES%. Call **Cygwin::sync_winenv()** to copy all Win32 environment variables to your process and note that cygwin will warn on every encounter of non-POSIX paths.

INSTALL PERL ON CYGWIN

This will install Perl, including *man* pages.

```
make install 2>&1 | tee log.make-install
```

NOTE: If "STDERR" is redirected "make install" will **not** prompt you to install *perl* into */usr/bin*.

You may need to be *Administrator* to run "make install". If you are not, you must have write access to the directories in question.

Information on installing the Perl documentation in HTML format can be found in the *INSTALL* document.

MANIFEST ON CYGWIN

These are the files in the Perl release that contain references to Cygwin. These very brief notes attempt

to explain the reason for all conditional code. Hopefully, keeping this up to date will allow the Cygwin port to be kept as clean as possible.

Documentation

INSTALL README.cygwin README.win32 MANIFEST
pod/perl.pod pod/perlport.pod pod/perlfaq3.pod
pod/perldelta.pod pod/perl5004delta.pod pod/perl56delta.pod
pod/perl561delta.pod pod/perl570delta.pod pod/perl572delta.pod
pod/perl573delta.pod pod/perl58delta.pod pod/perl581delta.pod
pod/perl590delta.pod pod/perlhist.pod pod/perlmodlib.pod
pod/perltoc.pod Porting/Glossary pod/perlgit.pod
Porting/checkAUTHORS.pl
dist/Cwd/Changes ext/Compress-Raw-Zlib/Changes
dist/Time-HiRes/Changes
ext/Compress-Raw-Zlib/README ext/Compress-Zlib/Changes
ext/DB_File/Changes ext/Encode/Changes ext/Sys-Syslog/Changes
ext/Win32API-File/Changes
lib/ExtUtils/CBuilder/Changes lib/ExtUtils/Changes
lib/ExtUtils/NOTES lib/ExtUtils/PATCHING lib/ExtUtils/README
lib/Net/Ping/Changes lib/Test/Harness/Changes
lib/Term/ANSIColor/ChangeLog lib/Term/ANSIColor/README

Build, Configure, Make, Install

cygwin/Makefile.SHs
ext/IPC/SysV/hints/cygwin.pl
ext/NDBM_File/hints/cygwin.pl
ext/ODBM_File/hints/cygwin.pl
hints/cygwin.sh
Configure - help finding hints from uname,
shared libperl required for dynamic loading
Makefile.SH Cross/Makefile-cross-SH
- linklibperl
Porting/patchls - cygwin in port list
installman - man pages with :: translated to .
installperl - install dll, install to 'pods'
makedepend.SH - uwinfix
regen_lib.pl - file permissions

NetWare/Makefile
plan9/mkfile

hints/uwin.sh
vms/descrip_mms.template
win32/Makefile

Tests

t/io/fs.t - no file mode checks if not ntsec skip rename() check when not check_case:relaxed
t/io/tell.t - binmode
t/lib/cygwin.t - builtin cygwin function tests
t/op/groups.t - basegroup has ID = 0
t/op/magic.t - $^X/symlink WORKAROUND, s/.exe//
t/op/stat.t - no /dev, skip Win32 ftCreationTime quirk (cache manager sometimes preserves ctime of file previously created and deleted), no -u (setuid)
t/op/taint.t - can't use empty path under Cygwin Perl
t/op/time.t - no tzset()

Compiled Perl Source

EXTERN.h - __declspec(dllimport)
XSUB.h - __declspec(dllexport)
cygwin/cygwin.c - os_extras (getcwd, spawn, and several Cygwin:: functions)
perl.c - os_extras, -i.bak
perl.h - binmode
doio.c - win9x can not rename a file when it is open
pp_sys.c - do not define h_errno, init _pwent_struct.pw_comment
util.c - use setenv
util.h - PERL_FILE_IS_ABSOLUTE macro
pp.c - Comment about Posix vs IEEE math under Cygwin
perlio.c - CR/LF mode
perliol.c - Comment about EXTCONST under Cygwin

Compiled Module Source

ext/Compress-Raw-Zlib/Makefile.PL - Can't install via CPAN shell under Cygwin
ext/Compress-Raw-Zlib/zlib-src/zutil.h

- Cygwin is Unix-like and has vsnprintf
ext/Errno/Errno_pm.PL - Special handling for Win32 Perl under Cygwin
ext/POSIX/POSIX.xs - tzname defined externally
ext/SDBM_File/sdbm/pair.c
- EXTCONST needs to be redefined from EXTERN.h
ext/SDBM_File/sdbm/sdbm.c
- binary open
ext/Sys/Syslog/Syslog.xs
- Cygwin has syslog.h
ext/Sys/Syslog/win32/compile.pl
- Convert paths to Windows paths
ext/Time-HiRes/HiRes.xs
- Various timers not available
ext/Time-HiRes/Makefile.PL
- Find w32api/windows.h
ext/Win32/Makefile.PL - Use various libraries under Cygwin
ext/Win32/Win32.xs - Child dir and child env under Cygwin
ext/Win32API-File/File.xs
- _open_osfhandle not implemented under Cygwin
ext/Win32CORE/Win32CORE.c
- __declspec(dllexport)

Perl Modules/Scripts

ext/B/t/OptreeCheck.pm - Comment about stderr/stdout order under Cygwin
ext/Digest-SHA/bin/shasum
- Use binary mode under Cygwin
ext/Sys/Syslog/win32/Win32.pm
- Convert paths to Windows paths
ext/Time-HiRes/HiRes.pm
- Comment about various timers not available
ext/Win32API-File/File.pm
- _open_osfhandle not implemented under Cygwin
ext/Win32CORE/Win32CORE.pm
- History of Win32CORE under Cygwin
lib/Cwd.pm - hook to internal Cwd::cwd

lib/ExtUtils/CBuilder/Platform/cygwin.pm
- use gcc for ld, and link to libperl.dll.a
lib/ExtUtils/CBuilder.pm
- Cygwin is Unix-like
lib/ExtUtils/Install.pm - Install and rename issues under Cygwin
lib/ExtUtils/MM.pm - OS classifications
lib/ExtUtils/MM_Any.pm - Example for Cygwin
lib/ExtUtils/MakeMaker.pm
- require MM_Cygwin.pm
lib/ExtUtils/MM_Cygwin.pm
- canonpath, cflags, manifypods, perl_archive
lib/File/Fetch.pm - Comment about quotes using a Cygwin example
lib/File/Find.pm - on remote drives stat() always sets st_nlink to 1
lib/File/Spec/Cygwin.pm - case_tolerant
lib/File/Spec/Unix.pm - preserve //unc
lib/File/Spec/Win32.pm - References a message on cygwin.com
lib/File/Spec.pm - Pulls in lib/File/Spec/Cygwin.pm
lib/File/Temp.pm - no directory sticky bit
lib/Module/CoreList.pm - List of all module files and versions
lib/Net/Domain.pm - No domainname command under Cygwin
lib/Net/Netrc.pm - Bypass using stat() under Cygwin
lib/Net/Ping.pm - ECONREFUSED is EAGAIN under Cygwin
lib/Pod/Find.pm - Set 'pods' dir
lib/Pod/Perldoc/ToMan.pm - '-c' switch for pod2man
lib/Pod/Perldoc.pm - Use 'less' pager, and use .exe extension
lib/Term/ANSIColor.pm - Cygwin terminal info
lib/perl5db.pl - use stdin not /dev/tty
utils/perlbug.PL - Add CYGWIN environment variable to report

Perl Module Tests

dist/Cwd/t/cwd.t
ext/Compress-Zlib/t/14gzopen.t
ext/DB_File/t/db-btree.t
ext/DB_File/t/db-hash.t
ext/DB_File/t/db-recno.t
ext/DynaLoader/t/DynaLoader.t
ext/File-Glob/t/basic.t
ext/GDBM_File/t/gdbm.t
ext/POSIX/t/sysconf.t

ext/POSIX/t/time.t
ext/SDBM_File/t/sdbm.t
ext/Sys/Syslog/t/syslog.t
ext/Time-HiRes/t/HiRes.t
ext/Win32/t/Unicode.t
ext/Win32API-File/t/file.t
ext/Win32CORE/t/win32core.t
lib/AnyDBM_File.t
lib/Archive/Extract/t/01_Archive-Extract.t
lib/Archive/Tar/t/02_methods.t
lib/ExtUtils/t/Embed.t
lib/ExtUtils/t/eu_command.t
lib/ExtUtils/t/MM_Cygwin.t
lib/ExtUtils/t/MM_Unix.t
lib/File/Compare.t
lib/File/Copy.t
lib/File/Find/t/find.t
lib/File/Path.t
lib/File/Spec/t/crossplatform.t
lib/File/Spec/t/Spec.t
lib/Net/hostent.t
lib/Net/Ping/t/110_icmp_inst.t
lib/Net/Ping/t/500_ping_icmp.t
lib/Net/t/netrc.t
lib/Pod/Simple/t/perlcyg.pod
lib/Pod/Simple/t/perlcygo.txt
lib/Pod/Simple/t/perlfaq.pod
lib/Pod/Simple/t/perlfaqo.txt
lib/User/grent.t
lib/User/pwent.t

BUGS ON CYGWIN

Support for swapping real and effective user and group IDs is incomplete. On WinNT Cygwin provides "setuid()", "seteuid()", "setgid()" and "setegid()". However, additional Cygwin calls for manipulating WinNT access tokens and security contexts are required.

AUTHORS

Charles Wilson <cwilson@ece.gatech.edu>, Eric Fifer <egf7@columbia.edu>, alexander smishlajev <als@turnhere.com>, Steven Morlock <newspost@morlock.net>, Sebastien Barre <Sebastien.Barre@utc.fr>, Teun Burgers <burgers@ecn.nl>, Gerrit P. Haase <gp@familiehaase.de>,

Reini Urban <rurban@cpan.org>, Jan Dubois <jand@activestate.com>, Jerry D. Hedden <jdhedden@cpan.org>.

HISTORY

Last updated: 2012-02-08