NAME

perldos - Perl under DOS, W31, W95.

SYNOPSIS

These are instructions for building Perl under DOS (or w??), using DJGPP v2.03 or later. Under w95 long filenames are supported.

DESCRIPTION

Before you start, you should glance through the README file found in the top-level directory where the Perl distribution was extracted. Make sure you read and understand the terms under which this software is being distributed.

This port currently supports MakeMaker (the set of modules that is used to build extensions to perl). Therefore, you should be able to build and install most extensions found in the CPAN sites.

Detailed instructions on how to build and install perl extension modules, including XS-type modules, is included. See 'BUILDING AND INSTALLING MODULES'.

Prerequisites for Compiling Perl on DOS

DJGPP

DJGPP is a port of GNU C/C++ compiler and development tools to 32-bit, protected-mode environment on Intel 32-bit CPUs running MS-DOS and compatible operating systems, by DJ Delorie <dj@delorie.com> and friends.

For more details (FAQ), check out the home of DJGPP at:

http://www.delorie.com/djgpp/

If you have questions about DJGPP, try posting to the DJGPP newsgroup: comp.os.msdos.djgpp, or use the email gateway djgpp@delorie.com.

You can find the full DJGPP distribution on any of the mirrors listed here:

http://www.delorie.com/djgpp/getting.html

You need the following files to build perl (or add new modules):

```
v2/djdev203.zip
v2gnu/bnu2112b.zip
v2gnu/gcc2953b.zip
v2gnu/bsh204b.zip
v2gnu/mak3791b.zip
v2gnu/fil40b.zip
v2gnu/sed3028b.zip
v2gnu/txt20b.zip
v2gnu/dif272b.zip
v2gnu/grep24b.zip
v2gnu/shl20jb.zip
v2gnu/gwk306b.zip
v2misc/csdpmi5b.zip
```

or possibly any newer version.

Pthreads

Thread support is not tested in this version of the djgpp perl.

Shortcomings of Perl under DOS

Perl under DOS lacks some features of perl under UNIX because of deficiencies in the UNIX-emulation, most notably:

- *fork()* and *pipe()*

- some features of the UNIX filesystem regarding link count and file dates

- in-place operation is a little bit broken with short filenames

- sockets

Building Perl on DOS

- Unpack the source package *perl5.8*.tar.gz* with djtarx. If you want to use long file names under w95 and also to get Perl to pass all its tests, don't forget to use

  ```
  set LFN=y
  set FNCASE=y
  ```

  before unpacking the archive.

- Create a "symlink" or copy your bash.exe to sh.exe in your (`$DJDIR`)`/bin` directory.

  ```
  ln -s bash.exe sh.exe
  ```

  [If you have the recommended version of bash for DJGPP, this is already done for you.]

  And make the `SHELL` environment variable point to this *sh.exe*:

  ```
  set SHELL=c:/djgpp/bin/sh.exe (use full path name!)
  ```

  You can do this in *djgpp.env* too. Add this line BEFORE any section definition:

  ```
  +SHELL=%DJDIR%/bin/sh.exe
  ```

- If you have *split.exe* and *gsplit.exe* in your path, then rename *split.exe* to *djsplit.exe*, and *gsplit.exe* to *split.exe*. Copy or link *gecho.exe* to *echo.exe* if you don't have *echo.exe*. Copy or link *gawk.exe* to *awk.exe* if you don't have *awk.exe*.

  [If you have the recommended versions of djdev, shell utilities and gawk, all these are already done for you, and you will not need to do anything.]

- Chdir to the djgpp subdirectory of perl toplevel and type the following commands:

  ```
  set FNCASE=y
  configure.bat
  ```

  This will do some preprocessing then run the Configure script for you. The Configure script is interactive, but in most cases you just need to press ENTER. The "set" command ensures that DJGPP preserves the letter case of file names when reading directories. If you already issued this set command when unpacking the archive, and you are in the same DOS session as when you unpacked the archive, you don't have to issue the set command again. This command is necessary *before* you start to (re)configure or (re)build perl in order to ensure both that perl builds correctly and that building XS-type modules can succeed. See the DJGPP info entry for "_preserve_fncase" for more information:

  ```
  info libc alphabetical _preserve_fncase
  ```

  If the script says that your package is incomplete, and asks whether to continue, just answer with Y (this can only happen if you don't use long filenames or forget to issue "set FNCASE=y" first).

  When Configure asks about the extensions, I suggest IO and Fcntl, and if you want database handling then SDBM_File or GDBM_File (you need to install gdbm for this one). If you want to use the POSIX extension (this is the default), make sure that the stack size of your *cc1.exe* is at least 512kbyte (you can check this with: `stubedit cc1.exe`).

  You can use the Configure script in non-interactive mode too. When I built my *perl.exe*, I used something like this:

  ```
  configure.bat -des
  ```

  You can find more info about Configure's command line switches in the *INSTALL* file.

  When the script ends, and you want to change some values in the generated *config.sh* file, then run

  ```
  sh Configure -S
  ```

  after you made your modifications.

IMPORTANT: if you use this `-S` switch, be sure to delete the CONFIG environment variable before running the script:

```
set CONFIG=
```

• Now you can compile Perl. Type:

```
make
```

Testing Perl on DOS

Type:

```
make test
```

If you're lucky you should see "All tests successful". But there can be a few failed subtests (less than 5 hopefully) depending on some external conditions (e.g. some subtests fail under linux/dosemu or plain dos with short filenames only).

Installation of Perl on DOS

Type:

```
make install
```

This will copy the newly compiled perl and libraries into your DJGPP directory structure. Perl.exe and the utilities go into (`$DJDIR`)/bin, and the library goes under (`$DJDIR`)/lib/perl5. The pod documentation goes under (`$DJDIR`)/lib/perl5/pod.

BUILDING AND INSTALLING MODULES ON DOS

Building Prerequisites for Perl on DOS

For building and installing non-XS modules, all you need is a working perl under DJGPP. Non-XS modules do not require re-linking the perl binary, and so are simpler to build and install.

XS-type modules do require re-linking the perl binary, because part of an XS module is written in "C", and has to be linked together with the perl binary to be executed. This is required because perl under DJGPP is built with the "static link" option, due to the lack of "dynamic linking" in the DJGPP environment.

Because XS modules require re-linking of the perl binary, you need both the perl binary distribution and the perl source distribution to build an XS extension module. In addition, you will have to have built your perl binary from the source distribution so that all of the components of the perl binary are available for the required link step.

Unpacking CPAN Modules on DOS

First, download the module package from CPAN (e.g., the "Comma Separated Value" text package, Text-CSV-0.01.tar.gz). Then expand the contents of the package into some location on your disk. Most CPAN modules are built with an internal directory structure, so it is usually safe to expand it in the root of your DJGPP installation. Some people prefer to locate source trees under /usr/src (i.e., (`$DJDIR`)/usr/src), but you may put it wherever seems most logical to you, *EXCEPT* under the same directory as your perl source code. There are special rules that apply to modules which live in the perl source tree that do not apply to most of the modules in CPAN.

Unlike other DJGPP packages, which are normal "zip" files, most CPAN module packages are "gzipped tarballs". Recent versions of WinZip will safely unpack and expand them, *UNLESS* they have zero-length files. It is a known WinZip bug (as of v7.0) that it will not extract zero-length files.

From the command line, you can use the djtar utility provided with DJGPP to unpack and expand these files. For example:

```
C:\djgpp>djtarx -v Text-CSV-0.01.tar.gz
```

This will create the new directory (`$DJDIR`)/Text-CSV-0.01, filling it with the source for this module.

Building Non-XS Modules on DOS

To build a non-XS module, you can use the standard module-building instructions distributed with perl modules.

```
perl Makefile.PL
make
make test
make install
```

This is sufficient because non-XS modules install only ''.pm'' files and (sometimes) pod and/or man documentation. No re-linking of the perl binary is needed to build, install or use non-XS modules.

Building XS Modules on DOS

To build an XS module, you must use the standard module-building instructions distributed with perl modules *PLUS* three extra instructions specific to the DJGPP ''static link'' build environment.

```
set FNCASE=y
perl Makefile.PL
make
make perl
make test
make -f Makefile.aperl inst_perl MAP_TARGET=perl.exe
make install
```

The first extra instruction sets DJGPP's FNCASE environment variable so that the new perl binary which you must build for an XS-type module will build correctly. The second extra instruction re-builds the perl binary in your module directory before you run ''make test'', so that you are testing with the new module code you built with ''make''. The third extra instruction installs the perl binary from your module directory into the standard DJGPP binary directory, (`$DJDIR`)/bin, replacing your previous perl binary.

Note that the MAP_TARGET value *must* have the ''.exe'' extension or you will not create a ''perl.exe'' to replace the one in (`$DJDIR`)/bin.

When you are done, the XS-module install process will have added information to your ''perllocal'' information telling that the perl binary has been replaced, and what module was installed. You can view this information at any time by using the command:

```
perl -S perldoc perllocal
```

AUTHOR

Laszlo Molnar, *laszlo.molnar@eth.ericsson.se* [Installing/building perl]

Peter J. Farley III *pjfarley@banet.net* [Building/installing modules]

SEE ALSO

perl(1) .