Perl also compiles with earlier releases of gcc (2.95.2 and up). See below for notes about using earlier versions of MinGW/gcc.

You also need dmake. See §**??** above on how to get it.

MinGW release 1 with gcc

The MinGW-1.1 bundle comes with gcc-2.95.3.

Make sure you install the binaries that work with MSVCRT.DLL as indicated in the README for the GCC bundle. You may need to set up a few environment variables (usually ran from a batch file).

There are a couple of problems with the version of gcc-2.95.2-msvcrt.exe released 7 November 1999:

- It left out a fix for certain command line quotes. To fix this, be sure to download and install the file fixes/quote-fix-msvcrt.exe from the above ftp location.
- The definition of the fpos_t type in stdio.h may be wrong. If your stdio.h has this problem, you will see an exception when running the test t/lib/io_xs.t. To fix this, change the typedef for fpos_t from "long" to "long long" in the file i386-mingw32msvc/include/stdio.h, and rebuild.

A potentially simpler to install (but probably soon-to-be-outdated) bundle of the above package with the mentioned fixes already applied is available here:

```
http://downloads.ActiveState.com/pub/staff/gsar/gcc-2.95.2-msvcrt.zip
ftp://ftp.ActiveState.com/pub/staff/gsar/gcc-2.95.2-msvcrt.zip
```

127.2.2 Building

- Make sure you are in the "win32" subdirectory under the perl toplevel. This directory contains a "Makefile" that will work with versions of nmake that come with Visual C++ or the Platform SDK, and a dmake "makefile.mk" that will work for all supported compilers. The defaults in the dmake makefile are setup to build using Microsoft Visual C++ 6.0 or newer.

- Edit the makefile.mk (or Makefile, if you're using nmake) and change the values of INST_DRV and INST_TOP. You can also enable various build flags. These are explained in the makefiles.

  Note that it is generally not a good idea to try to build a perl with INST_DRV and INST_TOP set to a path that already exists from a previous build. In particular, this may cause problems with the lib/ExtUtils/t/Embed.t test, which attempts to build a test program and may end up building against the installed perl's lib/CORE directory rather than the one being tested.

  You will have to make sure that CCTYPE is set correctly and that CCHOME points to wherever you installed your compiler.

  The default value for CCHOME in the makefiles for Visual C++ may not be correct for some versions. Make sure the default exists and is valid.

  If you have either the source or a library that contains des_fcrypt(), enable the appropriate option in the makefile. A ready-to-use version of fcrypt.c, based on the version originally written by Eric Young at ftp://ftp.funet.fi/pub/crypt/mirrors/dsi/libdes/, is bundled with the distribution. Set CRYPT_SRC to fcrypt.c to use this version. Alternatively, if you have built a library that contains des_fcrypt(), you can set CRYPT_LIB to point to the library name. Perl will also build without des_fcrypt(), but the crypt() builtin will fail at run time.

  Be sure to read the instructions near the top of the makefiles carefully.

- Type "dmake" (or "nmake" if you are using that make).

  This should build everything. Specifically, it will create perl.exe, perl58.dll at the perl toplevel, and various other extension dll's under the lib\auto directory. If the build fails for any reason, make sure you have done the previous steps correctly.