/\*

\* jconfig.doc

\*

\* Copyright (C) 1991-1994, Thomas G. Lane.

\* This file is part of the Independent JPEG Group's software.

\* For conditions of distribution and use, see the accompanying README file.

\*

\* This file documents the configuration options that are required to

\* customize the JPEG software for a particular system.

\*

\* The actual configuration options for a particular installation are stored

\* in jconfig.h. On many machines, jconfig.h can be generated automatically

\* or copied from one of the "canned" jconfig files that we supply. But if

\* you need to generate a jconfig.h file by hand, this file tells you how.

\*

\* DO NOT EDIT THIS FILE --- IT WON'T ACCOMPLISH ANYTHING.

\* EDIT A COPY NAMED JCONFIG.H.

\*/

/\*

\* These symbols indicate the properties of your machine or compiler.

\* #define the symbol if yes, #undef it if no.

\*/

/\* Does your compiler support function prototypes?

\* (If not, you also need to use ansi2knr, see install.doc)

\*/

#define HAVE\_PROTOTYPES

/\* Does your compiler support the declaration "unsigned char" ?

\* How about "unsigned short" ?

\*/

#define HAVE\_UNSIGNED\_CHAR

#define HAVE\_UNSIGNED\_SHORT

/\* Define "void" as "char" if your compiler doesn't know about type void.

\* NOTE: be sure to define void such that "void \*" represents the most general

\* pointer type, e.g., that returned by malloc().

\*/

/\* #define void char \*/

/\* Define "const" as empty if your compiler doesn't know the "const" keyword.

\*/

/\* #define const \*/

/\* Define this if an ordinary "char" type is unsigned.

\* If you're not sure, leaving it undefined will work at some cost in speed.

\* If you defined HAVE\_UNSIGNED\_CHAR then the speed difference is minimal.

\*/

#undef CHAR\_IS\_UNSIGNED

/\* Define this if your system has an ANSI-conforming <stddef.h> file.

\*/

#define HAVE\_STDDEF\_H

/\* Define this if your system has an ANSI-conforming <stdlib.h> file.

\*/

#define HAVE\_STDLIB\_H

/\* Define this if your system does not have an ANSI/SysV <string.h>,

\* but does have a BSD-style <strings.h>.

\*/

#undef NEED\_BSD\_STRINGS

/\* Define this if your system does not provide typedef size\_t in any of the

\* ANSI-standard places (stddef.h, stdlib.h, or stdio.h), but places it in

\* <sys/types.h> instead.

\*/

#undef NEED\_SYS\_TYPES\_H

/\* For 80x86 machines, you need to define NEED\_FAR\_POINTERS,

\* unless you are using a large-data memory model or 80386 flat-memory mode.

\* On less brain-damaged CPUs this symbol must not be defined.

\* (Defining this symbol causes large data structures to be referenced through

\* "far" pointers and to be allocated with a special version of malloc.)

\*/

#undef NEED\_FAR\_POINTERS

/\* Define this if your linker needs global names to be unique in less

\* than the first 15 characters.

\*/

#undef NEED\_SHORT\_EXTERNAL\_NAMES

/\* Although a real ANSI C compiler can deal perfectly well with pointers to

\* unspecified structures (see "incomplete types" in the spec), a few pre-ANSI

\* and pseudo-ANSI compilers get confused. To keep one of these bozos happy,

\* define INCOMPLETE\_TYPES\_BROKEN. This is not recommended unless you

\* actually get "missing structure definition" warnings or errors while

\* compiling the JPEG code.

\*/

#undef INCOMPLETE\_TYPES\_BROKEN

/\*

\* The following options affect code selection within the JPEG library,

\* but they don't need to be visible to applications using the library.

\* To minimize application namespace pollution, the symbols won't be

\* defined unless JPEG\_INTERNALS has been defined.

\*/

#ifdef JPEG\_INTERNALS

/\* Define this if your compiler implements ">>" on signed values as a logical

\* (unsigned) shift; leave it undefined if ">>" is a signed (arithmetic) shift,

\* which is the normal and rational definition.

\*/

#undef RIGHT\_SHIFT\_IS\_UNSIGNED

#endif /\* JPEG\_INTERNALS \*/

/\*

\* The remaining options do not affect the JPEG library proper,

\* but only the sample applications cjpeg/djpeg (see cjpeg.c, djpeg.c).

\* Other applications can ignore these.

\*/

#ifdef JPEG\_CJPEG\_DJPEG

/\* These defines indicate which image (non-JPEG) file formats are allowed. \*/

#define BMP\_SUPPORTED /\* BMP image file format \*/

#define GIF\_SUPPORTED /\* GIF image file format \*/

#define PPM\_SUPPORTED /\* PBMPLUS PPM/PGM image file format \*/

#undef RLE\_SUPPORTED /\* Utah RLE image file format \*/

#define TARGA\_SUPPORTED /\* Targa image file format \*/

/\* Define this if you want to name both input and output files on the command

\* line, rather than using stdout and optionally stdin. You MUST do this if

\* your system can't cope with binary I/O to stdin/stdout. See comments at

\* head of cjpeg.c or djpeg.c.

\*/

#undef TWO\_FILE\_COMMANDLINE

/\* Define this if your system needs explicit cleanup of temporary files.

\* This is crucial under MS-DOS, where the temporary "files" may be areas

\* of extended memory; on most other systems it's not as important.

\*/

#undef NEED\_SIGNAL\_CATCHER

/\* By default, we open image files with fopen(...,"rb") or fopen(...,"wb").

\* This is necessary on systems that distinguish text files from binary files,

\* and is harmless on most systems that don't. If you have one of the rare

\* systems that complains about the "b" spec, define this symbol.

\*/

#undef DONT\_USE\_B\_MODE

/\* Define this if you want percent-done progress reports from cjpeg/djpeg.

\*/

#undef PROGRESS\_REPORT

#endif /\* JPEG\_CJPEG\_DJPEG \*/