#ifdef \_WIN32

#ifndef \_\_GETOPT\_H\_\_

/\*\*

\* DISCLAIMER

\* This file is part of the mingw-w64 runtime package.

\*

\* The mingw-w64 runtime package and its code is distributed in the hope that it

\* will be useful but WITHOUT ANY WARRANTY. ALL WARRANTIES, EXPRESSED OR

\* IMPLIED ARE HEREBY DISCLAIMED. This includes but is not limited to

\* warranties of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

\*/

/\*

\* Copyright (c) 2002 Todd C. Miller <Todd.Miller@courtesan.com>

\*

\* Permission to use, copy, modify, and distribute this software for any

\* purpose with or without fee is hereby granted, provided that the above

\* copyright notice and this permission notice appear in all copies.

\*

\* THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES

\* WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF

\* MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR

\* ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

\* WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN

\* ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF

\* OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

\*

\* Sponsored in part by the Defense Advanced Research Projects

\* Agency (DARPA) and Air Force Research Laboratory, Air Force

\* Materiel Command, USAF, under agreement number F39502-99-1-0512.

\*/

/\*-

\* Copyright (c) 2000 The NetBSD Foundation, Inc.

\* All rights reserved.

\*

\* This code is derived from software contributed to The NetBSD Foundation

\* by Dieter Baron and Thomas Klausner.

\*

\* Redistribution and use in source and binary forms, with or without

\* modification, are permitted provided that the following conditions

\* are met:

\* 1. Redistributions of source code must retain the above copyright

\* notice, this list of conditions and the following disclaimer.

\* 2. Redistributions in binary form must reproduce the above copyright

\* notice, this list of conditions and the following disclaimer in the

\* documentation and/or other materials provided with the distribution.

\*

\* THIS SOFTWARE IS PROVIDED BY THE NETBSD FOUNDATION, INC. AND CONTRIBUTORS

\* ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED

\* TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR

\* PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE FOUNDATION OR CONTRIBUTORS

\* BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR

\* CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF

\* SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS

\* INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN

\* CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE)

\* ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

\* POSSIBILITY OF SUCH DAMAGE.

\*/

#define \_\_GETOPT\_H\_\_

/\* All the headers include this file. \*/

#include <crtdefs.h>

#include <errno.h>

#include <stdarg.h>

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define WIN32\_LEAN\_AND\_MEAN

#include <windows.h>

#ifdef \_\_cplusplus

extern "C" {

#endif

#define REPLACE\_GETOPT /\* use this getopt as the system getopt(3) \*/

//extern int optind; /\* index of first non-option in argv \*/

//extern int optopt; /\* single option character, as parsed \*/

//extern int opterr; /\* flag to enable built-in diagnostics... \*/

// /\* (user may set to zero, to suppress) \*/

//

//extern char \*optarg; /\* pointer to argument of current option \*/

#define PRINT\_ERROR ((opterr) && (\*options != ':'))

#define FLAG\_PERMUTE 0x01 /\* permute non-options to the end of argv \*/

#define FLAG\_ALLARGS 0x02 /\* treat non-options as args to option "-1" \*/

#define FLAG\_LONGONLY 0x04 /\* operate as getopt\_long\_only \*/

/\* return values \*/

#define BADCH (int)'?'

#define BADARG ((\*options == ':') ? (int)':' : (int)'?')

#define INORDER (int)1

#ifndef \_\_CYGWIN\_\_

#define \_\_progname \_\_argv[0]

#else

extern char \_\_declspec(dllimport) \* \_\_progname;

#endif

#ifdef \_\_CYGWIN\_\_

static char EMSG[] = "";

#else

#define EMSG ""

#endif

static int getopt\_internal(int, char\* const\*, const char\*,

const struct option\*, int\*, int);

static int parse\_long\_options(char\* const\*, const char\*,

const struct option\*, int\*, int);

static int gcd(int, int);

static void permute\_args(int, int, int, char\* const\*);

static char\* place = EMSG; /\* option letter processing \*/

/\* XXX: set optreset to 1 rather than these two \*/

static int nonopt\_start = -1; /\* first non option argument (for permute) \*/

static int nonopt\_end = -1; /\* first option after non options (for permute) \*/

/\* Error messages \*/

static const char recargchar[] = "option requires an argument -- %c";

static const char recargstring[] = "option requires an argument -- %s";

static const char ambig[] = "ambiguous option -- %.\*s";

static const char noarg[] = "option doesn't take an argument -- %.\*s";

static const char illoptchar[] = "unknown option -- %c";

static const char illoptstring[] = "unknown option -- %s";

static void \_vwarnx(const char\* fmt, va\_list ap);

static void warnx(const char\* fmt, ...);

/\*

\* Compute the greatest common divisor of a and b.

\*/

static int gcd(int a, int b);

/\*

\* Exchange the block from nonopt\_start to nonopt\_end with the block

\* from nonopt\_end to opt\_end (keeping the same order of arguments

\* in each block).

\*/

static void permute\_args(int panonopt\_start, int panonopt\_end, int opt\_end, char\* const\* nargv);

#ifdef REPLACE\_GETOPT

/\*

\* getopt --

\* Parse argc/argv argument vector.

\*

\* [eventually this will replace the BSD getopt]

\*/

int getopt(int nargc, char\* const\* nargv, const char\* options);

#endif /\* REPLACE\_GETOPT \*/

//extern int getopt(int nargc, char \* const \*nargv, const char \*options);

#ifdef \_BSD\_SOURCE

/\*

\* BSD adds the non-standard `optreset' feature, for reinitialisation

\* of `getopt' parsing. We support this feature, for applications which

\* proclaim their BSD heritage, before including this header; however,

\* to maintain portability, developers are advised to avoid it.

\*/

#define optreset \_\_mingw\_optreset

extern int optreset;

#endif

#ifdef \_\_cplusplus

}

#endif

/\*

\* POSIX requires the `getopt' API to be specified in `unistd.h';

\* thus, `unistd.h' includes this header. However, we do not want

\* to expose the `getopt\_long' or `getopt\_long\_only' APIs, when

\* included in this manner. Thus, close the standard \_\_GETOPT\_H\_\_

\* declarations block, and open an additional \_\_GETOPT\_LONG\_H\_\_

\* specific block, only when \*not\* \_\_UNISTD\_H\_SOURCED\_\_, in which

\* to declare the extended API.

\*/

#endif /\* !defined(\_\_GETOPT\_H\_\_) \*/

#if !defined(\_\_UNISTD\_H\_SOURCED\_\_) && !defined(\_\_GETOPT\_LONG\_H\_\_)

#define \_\_GETOPT\_LONG\_H\_\_

#ifdef \_\_cplusplus

extern "C" {

#endif

/\*

\* parse\_long\_options --

\* Parse long options in argc/argv argument vector.

\* Returns -1 if short\_too is set and the option does not match long\_options.

\*/

/\* static int parse\_long\_options(char\* const\* nargv, const char\* options, const struct option\* long\_options, int\* idx, int short\_too); \*/

/\*

\* getopt\_internal --

\* Parse argc/argv argument vector. Called by user level routines.

\*/

/\* static int getopt\_internal(int nargc, char\* const\* nargv, const char\* options, const struct option\* long\_options, int\* idx, int flags); \*/

/\*

\* getopt\_long --

\* Parse argc/argv argument vector.

\*/

int getopt\_long(int nargc, char\* const\* nargv, const char\* options, const struct option\* long\_options, int\* idx);

/\*

\* getopt\_long\_only --

\* Parse argc/argv argument vector.

\*/

int getopt\_long\_only(int nargc, char\* const\* nargv, const char\* options, const struct option\* long\_options, int\* idx);

/\*

\* Previous MinGW implementation had...

\*/

#ifndef HAVE\_DECL\_GETOPT

/\*

\* ...for the long form API only; keep this for compatibility.

\*/

#define HAVE\_DECL\_GETOPT 1

#endif

#ifdef \_\_cplusplus

}

#endif

#endif /\* !defined(\_\_UNISTD\_H\_SOURCED\_\_) && !defined(\_\_GETOPT\_LONG\_H\_\_) \*/

#endif