

```
/*
 * Copyright (c) 1991, 1993
 * The Regents of the University of California. All rights reserved.
 *
 * By using this file, you agree to the terms and conditions set
 * forth in the LICENSE file which can be found at the top level of
 * the sendmail distribution.
 *
 * $Id: cdefs.h,v 1.2 2003/10/20 15:03:16 chrisgreen Exp $
 * @(#)cdefs.h 8.8 (Berkeley) 1/9/95
 */

#ifndef _CDEFS_H_
# define _CDEFS_H_

# if defined(__cplusplus)
# define __BEGIN_DECLS extern "C" {
# define __END_DECLS };
# else /* defined(__cplusplus) */
# define __BEGIN_DECLS
# define __END_DECLS
# endif /* defined(__cplusplus) */

/*
 * The __CONCAT macro is used to concatenate parts of symbol names, e.g.
 * with "#define OLD(foo) __CONCAT(old,foo)", OLD(foo) produces oldfoo.
 * The __CONCAT macro is a bit tricky -- make sure you don't put spaces
 * in between its arguments. __CONCAT can also concatenate double-quoted
 * strings produced by the __STRING macro, but this only works with ANSI C.
 */
# if defined(__STDC__) || defined(__cplusplus)
# define __P(protos) protos /* full-blown ANSI C */
# ifndef __CONCAT
# define __CONCAT(x,y) x ## y
# endif /* !__CONCAT */
# define __STRING(x) #x

# ifndef __const
# define __const const /* define reserved names to standard */
# endif /* !__const */
# define __signed signed
# define __volatile volatile
# if defined(__cplusplus)
# define __inline inline /* convert to C++ keyword */
# else /* defined(__cplusplus) */
# ifndef __GNUC__
# define __inline /* delete GCC keyword */
# endif /* !__GNUC__ */
# endif /* defined(__cplusplus) */

# else /* defined(__STDC__) || defined(__cplusplus) */
# define __P(protos) () /* traditional C preprocessor */
# ifndef __CONCAT
# define __CONCAT(x,y) x/**/y
# endif /* !__CONCAT */
# define __STRING(x) "x"

# ifndef __GNUC__
# define __const /* delete pseudo-ANSI C keywords */
# define __inline
# define __signed
# define __volatile
*/
 * In non-ANSI C environments, new programs will want ANSI-only C keywords
 * deleted from the program and old programs will want them left alone.
 * When using a compiler other than gcc, programs using the ANSI C keywords
 * const, inline etc. as normal identifiers should define -DNO_ANSI_KEYWORDS.
 * When using "gcc -traditional", we assume that this is the intent; if
 * __GNUC__ is defined but __STDC__ is not, we leave the new keywords alone.
 */
# ifndef NO_ANSI_KEYWORDS
# define const /* delete ANSI C keywords */
# define inline
```

```
#   define signed
#   define volatile
#   endif /* ! NO_ANSI_KEYWORDS */
#   endif /* ! __GNUC__ */
# endif /* defined(__STDC__) || defined(__cplusplus) */

/*
 * GCC1 and some versions of GCC2 declare dead (non-returning) and
 * pure (no side effects) functions using "volatile" and "const";
 * unfortunately, these then cause warnings under "-ansi -pedantic".
 * GCC2 uses a new, peculiar __attribute__((attrs)) style. All of
 * these work for GNU C++ (modulo a slight glitch in the C++ grammar
 * in the distribution version of 2.5.5).
 */
# if !defined(__GNUC__) || __GNUC__ < 2 || \
    (__GNUC__ == 2 && __GNUC_MINOR__ < 5)
#   define __attribute__(x) /* delete __attribute__ if non-gcc or gcc1 */
#   if defined(__GNUC__) && !defined(__STRICT_ANSI__)
#     define __dead      __volatile
#     define __pure      __const
#   endif /* defined(__GNUC__) && !defined(__STRICT_ANSI__) */
# endif /* !defined(__GNUC__) || __GNUC__ < 2 || \ */

/* Delete pseudo-keywords wherever they are not available or needed. */
# ifndef __dead
#   define __dead
#   define __pure
# endif /* ! __dead */

#endif /* ! _CDEFS_H_ */
```