Header Files

Last Updated: 2023-03-24

Information that is needed by several different files or functions is collected into a header file. A header file contains C-language definitions and structures. Centralizing information into a header file facilitates the creation and update of programs. Because **#include** statements are used to insert header files into a C-language program, header files are often referred to as include files.

Header files define the following functions:

- Structures of certain files and subroutines
- Type definition (typedef) synonyms for data types
- System parameters or implementation characteristics
- Constants and macros that are substituted during the C language preprocessing phase

By convention, the names of header files end with **.h** (dot h). The **.h** suffix is used by header files that are provided with the operating system; however, the suffix is not required for user-generated header files.

> **Note:** Several of the header files provided with the operating system end with **.inc** (include file).

Additional header files are provided with the operating system. Most of these can be found in either the **/usr/include** directory or the **/usr/include/sys** directory. Use the **pg** command to view the contents of a header file.

More information about the following header files is provided in this documentation:

Item	Description
a.out.h	Defines the structure of the standard **a.out** file.
acct.h	Describes the format of the records in the system accounting files.
ar.h	Describes the format of an archive file.
audit.h	Defines values used by the auditing system as well as the structure of a bin.
core.h	Describes the structures created as a result of a core dump.
ct_ffdc.h	Provides data types, definitions, and interface prototypes for the First Failure Data Capture (FFDC) C language library interfaces.
dirent.h	Describes the format of a file system-independent directory entry.
eucioctl.h	Defines ioctl operations and data types for handling EUC code sets.
fcntl.h	Defines values for the **fcntl** and **open** subroutines.
filsys.h	Contains the format of a file system logical volume.
flock.h	Defines the file control options.
fullstat.h	Describes the data structure returned by the **fullstat** and **ffullstat** subroutines.
iconv.h	Defines types, macros, and subroutines for character code set conversion.
ipc.h	Defines structures used by the subroutines that perform interprocess communications operations.
ldr.h	Describes the **ld_info** data type and loader entry points.
libperfstst.h	Describes the structures and constants used by the **libperfstat.h** API subroutines.
limits.h	Defines implementation limits identified by the IEEE POSIX 1003 standard.
math.h	Defines math subroutines and constants
mode.h	Defines the interpretation of a file mode.
msg.h	Defines structures used by the subroutines that perform message queueing operations.
mtio.h	Defines the magnetic tape user include file.

ICPillar, LLC, Exhibit 2006
Page 2006 - 1
IPR2024-00476, ARM Limited v. ICPillar, LLC

Item	Description
param.h	Defines certain hardware-dependent parameters.
poll.h	Defines the **pollfd** structure used by the **poll** subroutine.
sem.h	Defines the structures that are used by subroutines that perform semaphore operations.
sgtty.h	Defines structures used by the Berkeley terminal interface.
shm.h	Defines structures used by the subroutines that perform shared memory operations.
spc.h	Defines external interfaces provided by the System Resource Controller (SRC) subroutines.
srcobj.h	Defines structures used by the System Resource Controller (SRC) subsystem.
stat.h	Describes the data structure returned by the status subroutines.
statfs.h	Describes the structure of the statistics returned by the status subroutines.
statvfs.h	Describes the structure of the statistics that are returned by the **statvfs** subroutines and **fsatvfs** subroutines.
systemcfg.h	Defines the **_system_configuration** structure.
tar.h	Defines flags used in the **tar** archive header.
termio.h	Defines structures used by the terminal interface for compatibility of Version 2 of the operating system.
termios.h	Defines structures used by the POSIX terminal interface.
termiox.h	Defines the structure of the **termiox** file, which provides the extended terminal interface.
trace.h	Defines implementation trace identified by IEEE POSIX 1003.
types.h	Defines primitive system data types.
unistd.h	Defines POSIX implementation characteristics.
utmp.h	Defines the format of certain user and accounting information files.
values.h	Defines hardware-dependent values.
vmount.h	Describes the structure of a mounted file system.

- List of Major Control Block Header Files
- ct_ffdc.h File
- dirent.h File
- dlfcn.h File
- eucioctl.h File
- fcntl.h File
- filsys.h File
- flock.h File
- fullstat.h File
- grp.h File
- iconv.h File
- inode.h File
- inttypes.h File
- ipc.h File
- iso646.h File
- ldr.h File
- limits.h File
- libperfstat.h File
- math.h File
- mode.h File
- msg.h File
- mtio.h File
- param.h File
- pmapi.h File

- poll.h File
- pollset.h File
- pthread.h File
- pwd.h File
- pwdpolicy.h File
- sem.h File
- sgtty.h File
- shm.h File
- spc.h File
- srcobj.h File
- stat.h File
- statfs.h File
- statvfs.h File
- systemcfg.h File
- tar.h File
- termio.h File
- termios.h File
- termiox.h File
- threads.h File
- trace.h File
- types.h File
- uchar.h File
- unistd.h File
- utmp.h File
- values.h File
- vmount.h File
- wctype.h File
- wlm.h File

Parent topic:

→ Files Reference

Related information

→ pg Command