NAME

stdint - standard integer types

SYNOPSIS

#include <stdint.h>

DESCRIPTION

The <*stdint.h*> header provides source-portable integer types of a specific size, smallest memory footprint with a minimum size, fastest access speed with a minimum size, largest integer size, and those capable of storing pointers.

The types *int8_t*, *int16_t*, *int32_t*, and *int64_t* provide a signed integer type of width 8, 16, 32, or 64 bits, respectively. The types *uint8_t*, *uint16_t*, *uint32_t*, and *uint64_t* provide an unsigned integer type of width 8, 16, 32, or 64 bits, respectively. These integer types should be used when a specific size is required.

The types *int_fast8_t*, *int_fast16_t*, *int_fast32_t*, and *int_fast64_t* provide the fastest signed integer type with a width of at least 8, 16, 32, or 64 bits, respectively. The types *uint_fast8_t*, *uint_fast16_t*, *uint_fast32_t*, and *uint_fast64_t* provide the fastest unsigned integer type with a width of at least 8, 16, 32, or 64 bits, respectively. These types should be used when access speed is paramount, and when a specific size is not required.

The types *int_least8_t*, *int_least16_t*, *int_least32_t*, and *int_least64_t* provide the smallest memory footprint signed integer type with a width of at least 8, 16, 32, or 64 bits, respectively. The types *uint_least8_t*, *uint_least16_t*, *uint_least32_t*, and *uint_least64_t* provide the smallest memory footprint unsigned integer type with a width of at least 8, 16, 32, or 64 bits, respectively. These types should be used when memory storage is of concern, and when a specific size is not required.

The type *intmax_t* provides a signed integer type large enough to hold any other signed integer. The type *uintmax_t* provides an unsigned integer type large enough to hold any other unsigned integer. These types are generally the largest signed and unsigned integer types available on a specific architecture.

The type *intptr_t* provides a signed integer type with the ability to hold a pointer to *void*, that can later be converted back to a pointer to *void*.

The type *uintptr_t* provides an unsigned integer type with the ability to hold a pointer to *void*, that can later be converted back to a pointer to *void*.

Additionally, the <*stdint.h*> header defines some macros, but none of them are documented here.

STANDARDS

The <*stdint.h*> header conforms to ISO/IEC 9899:1999 ("ISO C99") and IEEE Std 1003.1-2001 ("POSIX.1").

HISTORY

The <*stdint.h*> header was first introduced in FreeBSD 5.0.