

---

# liba2i

**THE MAN-PAGES BOOK**

**Maintainers:**

Alejandro Colomar <[alx@kernel.org](mailto:alx@kernel.org)> 2024 - present (0.1 - HEAD)

**Name**

a2i – convert a string to an integer

**Library**

String-to-integer library (*liba2i*, *-la2i*)

**Synopsis**

```
#include <a2i/a2i/a2i.h>
```

```
int a2i( TYPE, TYPE *restrict n, QChar *s,  
        QChar **_Nullable restrict endp, int base,  
        TYPE min, TYPE max);
```

**Description**

This macro is like [a2s\(3\)](#) and [a2u\(3\)](#), but works with all integer types.

**See also**

[a2s\(3\)](#), [a2u\(3\)](#), [str2i\(3\)](#), [str2s\(3\)](#), [str2u\(3\)](#)

**Name**

a2s, a2shh, a2sh, a2si, a2sl, a2sll, – convert a string to a signed integer

**Library**

String-to-integer library (*liba2i*, *-la2i*)

**Synopsis**

```
#include <a2i/a2i/a2s.h>
```

**Type-generic macro**

```
int a2s( TYPE, TYPE *restrict n, QChar *s,
        QChar **_Nullable restrict endp, int base,
        TYPE min, TYPE max);
```

**const-generic functions**

```
int a2shh( signed char *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          signed char min, signed char max);
```

```
int a2sh( short *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          short min, short max);
```

```
int a2si( int *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          int min, int max);
```

```
int a2sl( long *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          long min, long max);
```

```
int a2sll( long long *restrict n, QChar *s,
           QChar **_Nullable restrict endp, int base,
           long long min, long long max);
```

**Description**

These functions convert the initial portion of the string pointed to by *s* to a signed integer of base *base*, ensure that the number is in the range [*min*, *max*] and store it in *\*n*.

They are similar to *strtoi(3bsd)*, which itself is similar to *strtol(3)*

The *base* must a value in the range [2, 36], or the special value 0, which means either 8, 10, or 16, depending on the prefix ("0", none, or "0x", respectively).

If *endp* is not a null pointer, these functions store the address of the first invalid character in *\*endp*.

These functions always write to *\*n*.

- The value parsed if it's in the range, or the closest value in the range otherwise.
- 0 if no value was parsed, or the closest value in the range if 0 is not in the range.

The type-generic macro expands to the appropriate function depending on *TYPE*. It's useful for parsing a number into a typedef.

**QChar**

These functions are implemented with const-generic wrapper macros. These decide the const-ness of *endp* depending on the const-ness of the input *s*. So, if *s* is of type *const char \**, *endp* will be of type *const char \*\**; and if *s* is of type *char \**, *endp* will be

of type *char* \*\*.

There are functions named with a *\_nc* suffix, which don't have any const parameters; and functions named with a *\_c* suffix, which use const in both *s* and *endp*.

## Return value

On success, 0 is returned. On error, -1 is returned, and *errno* is set to indicate the error.

## Errors

### EINVAL

The *base* is not in the range [2, 36] nor 0.

### ECANCELED

The string did not contain any characters that were converted.

### ERANGE

The parsed number was out of range.

### ENOTSUP

The string contained non-numeric characters after a valid number.

## Examples

```
if (a2s(pid_t, &pid, s, NULL, 10, 1, type_max(pid_t)) == -1)
    goto err;
```

## See also

[a2i\(3\)](#), [a2u\(3\)](#), [str2i\(3\)](#), [str2s\(3\)](#), [str2u\(3\)](#), [atof\(3\)](#), [atoi\(3\)](#), [strtod\(3\)](#), [strtoimax\(3\)](#), [strtol\(3\)](#), [strtoi\(3\)](#)

**Name**

a2u, a2uhh, a2uh, a2ui, a2ul, a2ull – convert a string to an unsigned integer

**Library**

String-to-integer library (*liba2i*, *-la2i*)

**Synopsis**

```
#include <a2i/a2i/a2u.h>
```

**Type-generic macro**

```
int a2u( TYPE, TYPE *restrict n, QChar *s,
        QChar **_Nullable restrict endp, int base,
        TYPE min, TYPE max);
```

**const-generic functions**

```
int a2uhh( unsigned char *restrict n, QChar *s,
           QChar **_Nullable restrict endp, int base,
           unsigned char min, unsigned char max);
```

```
int a2uh( unsigned short *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          unsigned short min, unsigned short max);
```

```
int a2ui( unsigned int *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          unsigned int min, unsigned int max);
```

```
int a2ul( unsigned long *restrict n, QChar *s,
          QChar **_Nullable restrict endp, int base,
          unsigned long min, unsigned long max);
```

```
int a2ull( unsigned long long *restrict n, QChar *s,
           QChar **_Nullable restrict endp, int base,
           unsigned long long min, unsigned long long max);
```

**Description**

These functions and macros are the unsigned counterparts of [a2s\(3\)](#).

These reject negative values, instead of silently wrapping them around (both [strtoul\(3\)](#) and [strtou\(3bsd\)](#) wrap around negative values before checking for overflow).

**Examples**

```
if (a2u(useconds_t, &us, s, NULL, 10, 0, 1000000) == -1)
    goto err;
```

**See also**

[a2i\(3\)](#), [a2s\(3\)](#), [str2i\(3\)](#), [str2s\(3\)](#), [str2u\(3\)](#), [atof\(3\)](#), [atoi\(3\)](#), [strtod\(3\)](#), [strtoumax\(3\)](#), [strtoul\(3\)](#), [strtou\(3\)](#)

**Name**

str2i – convert a string to an integer

**Library**

String-to-integer library (*liba2i*, *-la2i*)

**Synopsis**

```
#include <a2i/str2i/str2i.h>
```

```
int str2i( TYPE, TYPE *restrict n, const char *s);
```

**Description**

This macro is like [str2s\(3\)](#) and [str2u\(3\)](#), but works with all integer types.

**See also**

[a2i\(3\)](#), [a2s\(3\)](#), [a2u\(3\)](#), [str2s\(3\)](#), [str2u\(3\)](#)

## Name

str2s, str2shh, str2sh, str2si, str2sl, str2sll, – convert a string to a signed integer

## Library

String-to-integer library (*liba2i*, *-la2i*)

## Synopsis

```
#include <a2i/str2i/str2s.h>
```

```
int str2s( TYPE, TYPE *restrict n, const char *s);
```

```
int str2shh( signed char *restrict n, const char *s);
```

```
int str2sh( short *restrict n, const char *s);
```

```
int str2si( int *restrict n, const char *s);
```

```
int str2sl( long *restrict n, const char *s);
```

```
int str2sll( long long *restrict n, const char *s);
```

## Description

These functions and macro are convenience wrappers around [a2s\(3\)](#), providing common defaults for the remaining arguments. The base is 0; the end pointer is NULL; and the limits are those of the type.

## Examples

The following calls are all equivalent:

```
str2sl(&n, s);
str2s(long, &n, s);
str2i(long, &n, s);
a2sl(&n, s, NULL, 0, LONG_MIN, LONG_MAX);
a2s(long, &n, s, NULL, 0, LONG_MIN, LONG_MAX);
a2i(long, &n, s, NULL, 0, LONG_MIN, LONG_MAX);
```

## See also

[a2i\(3\)](#), [a2s\(3\)](#), [a2u\(3\)](#), [str2i\(3\)](#), [str2u\(3\)](#)

## Name

str2u, str2uhh, str2uh, str2ui, str2ul, str2ull – convert a string to an unsigned integer

## Library

String-to-integer library (*liba2i*, *-la2i*)

## Synopsis

```
#include <a2i/str2i/str2u.h>
```

```
int str2u( TYPE, TYPE *restrict n, const char *s);
```

```
int str2uhh( unsigned char *restrict n, const char *s);
```

```
int str2uh( unsigned short *restrict n, const char *s);
```

```
int str2ui( unsigned int *restrict n, const char *s);
```

```
int str2ul( unsigned long *restrict n, const char *s);
```

```
int str2ull( unsigned long long *restrict n, const char *s);
```

## Description

These functions and macros are convenience wrappers around [a2u\(3\)](#), providing common defaults for the remaining arguments. The base is 0; the end pointer is NULL; and the limits are those of the type.

## Examples

The following calls are all equivalent:

```
str2ul(&n, s);
str2u(unsigned long, &n, s);
str2i(unsigned long, &n, s);
a2ul(&n, s, NULL, 0, 0, ULONG_MAX);
a2u(unsigned long, &n, s, NULL, 0, 0, ULONG_MAX);
a2i(unsigned long, &n, s, NULL, 0, 0, ULONG_MAX);
```

## See also

[a2i\(3\)](#), [a2s\(3\)](#), [a2u\(3\)](#), [str2i\(3\)](#), [str2s\(3\)](#)