

NAME

atoi, atol, atoll – convert a string to an integer

LIBRARY

Standard C library (*libc*, *-lc*)

SYNOPSIS

```
#include <stdlib.h>
```

```
int atoi(const char *nptr);
```

```
long atol(const char *nptr);
```

```
long long atoll(const char *nptr);
```

Feature Test Macro Requirements for glibc (see **feature_test_macros(7)**):

```
atoll():
```

```
_ISOC99_SOURCE
```

```
|| /* glibc <= 2.19: */ _BSD_SOURCE || _SVID_SOURCE
```

DESCRIPTION

The **atoi()** function converts the initial portion of the string pointed to by *nptr* to *int*. The behavior is the same as

```
strtol(nptr, NULL, 10);
```

except that **atoi()** does not detect errors.

The **atol()** and **atoll()** functions behave the same as **atoi()**, except that they convert the initial portion of the string to their return type of *long* or *long long*.

RETURN VALUE

The converted value or 0 on error.

ATTRIBUTES

For an explanation of the terms used in this section, see **attributes(7)**.

Interface	Attribute	Value
atoi() , atol() , atoll()	Thread safety	MT-Safe locale

VERSIONS

POSIX.1 leaves the return value of **atoi()** on error unspecified. On glibc, musl libc, and uClibc, 0 is returned on error.

STANDARDS

C11, POSIX.1-2008.

HISTORY

C99, POSIX.1-2001, SVr4, 4.3BSD.

C89 and POSIX.1-1996 include the functions **atoi()** and **atol()** only.

BUGS

errno is not set on error so there is no way to distinguish between 0 as an error and as the converted value. No checks for overflow or underflow are done. Only base-10 input can be converted. It is recommended to instead use the **strtol()** and **strtoul()** family of functions in new programs.

SEE ALSO

atof(3), **strtod(3)**, **strtol(3)**, **strtoul(3)**