

Search:

Go

Not logged in

register

log in

Reference <stdio> **snprintf**

C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
<cassert> (assert.h)
<cctype> (ctype.h)
<cerrno> (errno.h)
<cfenv> (fenv.h)
<cmath> (math.h)
<cintrypes> (intrypes.h)
<ciso646> (iso646.h)
<climits> (limits.h)
<locale> (locale.h)
<cmath> (math.h)
<setjmp> (setjmp.h)
<csignal> (signal.h)
<csdarg> (stdarg.h)
<csdbool> (stdbool.h)
<csddef> (stddef.h)
<csdint> (stdint.h)
<csdio> (stdio.h)
<csdlib> (stdlib.h)
<csstring> (string.h)
<ctgmth> (tgmath.h)
<ctime> (time.h)
<cuchar> (uchar.h)
<cwchar> (wchar.h)
<cwctype> (wctype.h)
Containers:
Input/Output:
Multi-threading:
Other:

<stdio> (stdio.h)
functions:
clearerr
fclose
feof
ferror
fflush
fgetc
fgetpos
fgets
fopen
fprintf
fputc
fputs
fread
freopen
fscanf
fseek
fsetpos
ftell
fwrite
getc
getchar
gets
perror
printf
putc
putchar
puts
remove
rename
rewind
scanf
setbuf
setvbuf
snprintf
sprintf
sscanf
tmpfile
tmpnam
ungetc
vfprintf
vfprintf
vprintf
vscanf
vsprintf
vsprintf

function

snprintf

<stdio>

```
int snprintf ( char * s, size_t n, const char * format, ... );
```

Write formatted output to sized buffer

Composes a string with the same text that would be printed if *format* was used on [printf](#), but instead of being printed, the content is stored as a *C string* in the buffer pointed by *s* (taking *n* as the maximum buffer capacity to fill).

If the resulting string would be longer than *n-1* characters, the remaining characters are discarded and not stored, but counted for the value returned by the function.

A terminating null character is automatically appended after the content written.

After the *format* parameter, the function expects at least as many additional arguments as needed for *format*.

Parameters

- s
- Pointer to a buffer where the resulting C-string is stored.
The buffer should have a size of at least *n* characters.
- n
- Maximum number of bytes to be used in the buffer.
The generated string has a length of at most *n-1*, leaving space for the additional terminating null character.
size_t is an unsigned integral type.
- format
- C string that contains a format string that follows the same specifications as *format* in [printf](#) (see [printf](#) for details).
- ... (additional arguments)
- Depending on the *format* string, the function may expect a sequence of additional arguments, each containing a value to be used to replace a *format specifier* in the *format* string (or a pointer to a storage location, for *n*).
There should be at least as many of these arguments as the number of values specified in the *format specifiers*.
Additional arguments are ignored by the function.

Return Value

The number of characters that would have been written if *n* had been sufficiently large, not counting the terminating null character.

If an encoding error occurs, a negative number is returned.

Notice that only when this returned value is non-negative and less than *n*, the string has been completely written.

Example

```
1 /* snprintf example */
2 #include <stdio.h>
3
4 int main ()
5 {
6     char buffer [100];
7     int cx;
8
9     cx = snprintf ( buffer, 100, "The half of %d is %d", 60, 60/2 );
10
11     if (cx>=0 && cx<100)        // check returned value
12         snprintf ( buffer+cx, 100-cx, ", and the half of that is %d.", 60/2/2 );
13
14     puts (buffer);
15
16     return 0;
17 }
18
```

Edit & Run

Output:

The half of 60 is 30, and the half of that is 15.

For more examples on formatting see [printf](#).

See also

printf	Print formatted data to stdout (function)
sprintf	Write formatted data to string (function)
strcat	Concatenate strings (function)
sscanf	Read formatted data from string (function)

[vscanf](#)**objects:**[stderr](#)[stdin](#)[stdout](#)**types:**[FILE](#)[fpos_t](#)[size_t](#)**macro constants:**[BUFSIZ](#)[EOF](#)[FILENAME_MAX](#)[FOPEN_MAX](#)[L_tmpnam](#)[NULL](#)[TMP_MAX](#)[Home page](#) | [Privacy policy](#)

© cplusplus.com, 2000-2020 - All rights reserved - v3.2

[Spotted an error? contact us](#)