URL: http://cxc.harvard.edu/ciao3.4/sscanf.tm.html
Last modified: December 2006

AHELP for CIAO 3.4

sscanf

Context: slangrtl

Jump to: Description Example See Also

Synopsis

Parse a formatted string

Syntax

```
Int_Type sscanf (s, fmt, r1, ... rN)
```

Description

```
   String_Type s, fmt;
   Ref_Type r1, ..., rN
```

The sscanf function parses the string s according to the format fmt and sets the variables whose references are given by r1, ..., rN. The function returns the number of references assigned, or −1 upon error.

The format string fmt consists of ordinary characters and conversion specifiers. A conversion specifier begins with the special character % and is described more fully below. A white space character in the format string matches any amount of whitespace in the input string. Parsing of the format string stops whenever a match fails.

The % is used to denote a conversion specifier whose general form is given by %[*][width][type]format where the brackets indicate optional items. If * is present, then the conversion will be performed by no assignment to a reference will be made. The width specifier specifies the maximum field width to use for the conversion. The type modifier is used to indicate size of the object, e.g., a short integer, as follows.

If type is given as the character h, then if the format conversion is for an integer (dioux), the object assigned will be a short integer. If type is l, then the conversion will be to a long integer for integer conversions, or to a double precession floating point number for floating point conversions.

The format specifier is a character that specifies the conversion:

```
      %     Matches a literal percent character.  No assigment is
            performed.
      d     Matches a signed decimal integer.
      D     Matches a long decimal integer (equiv to `ld')
      u     Matches an unsigned decimal integer
```

```
      U     Matches an unsigned long decimal integer (equiv to `lu')
      i     Matches either a hexidecimal integer, decimal integer, or
            octal integer.
      I     Equivalent to `li'.
      x     Matches a hexidecimal integer.
      X     Matches a long hexidecimal integer (same as `lx').
      e,f,g Matches a decimal floating point number (Float_Type).
      E,F,G Matches a double precision floating point number, same as `lf'.
      s     Matches a string of non−whitespace characters (String_Type).
      c     Matches one character.  If width is given, width
            characters are matched.
      n     Assigns the number of characters scanned so far.
      [...] Matches zero or more characters from the set of characters
            enclosed by the square brackets.  If '^' is given as the
            first character, then the complement set is matched.
```

Example

Suppose that s is "Coffee: (3,4,12.4)". Then

```
   n = sscanf (s, "%[a−zA−Z]: (%d,%d,%lf)", &item, &x, &y, &z);
```

will set n to 4, item to "Coffee", x to 3 , y to 4, and z to the double precision number 12.4. However,

```
   n = sscanf (s, "%s: (%d,%d,%lf)", &item, &x, &y, &z);
```

will set n to 1, item to "Coffee:" and the remaining variables will not be assigned.

See Also

slangrtl

_apropos, _print_stack, _slang_guess_type, atof, char, double, fread, fwrite, int, integer, is_substr, isdigit, message, pack, pad_pack_format, putenv, set_float_format, sizeof_pack, sprintf, strcat, string, string_match, string_match_nth, tolower, toupper, typecast, uname, unpack, verror, vmessage

---

URL: http://cxc.harvard.edu/ciao3.4/sscanf.tm.html
Last modified: December 2006