

NAME**libirccd-subst** - create templated string**SYNOPSIS**

#include <irccd/subst.h>

```
ssize_t  
irc_subst(char *dst, size_t dstsz, const char *template, const struct irc_subst *subst);
```

DESCRIPTION

This header provides functions to generate templated strings using a custom format as defined in irccd-templates(7).

The *struct irc_subst_keyword* and *struct irc_subst* are declared as:

```
struct irc_subst_keyword {  
    const char *key;  
    const char *value;  
};  
  
struct irc_subst {  
    time_t time;  
    enum irc_subst_flags flags;  
    const struct irc_subst_keyword *keywords;  
    size_t keywordsz;  
};
```

The fields of *struct irc_subst_keyword* are:

key The keyword name.

value The keyword value.

The fields of *struct irc_subst* are:

time The timestamp if the template contains time formatting and IRC_SUBST_DATE is present in *flags*.

flags The allowed formats in the source string as an OR'ed combination, see below.

keywords An array of user keywords to replace in the source string.

keywordsz The number of elements present in *keywords* field.

The *enum irc_subst_flags* is declared as:

```
enum irc_subst_flags {
    IRC_SUBST_DATE      = (1 << 0),
    IRC_SUBST_KEYWORDS  = (1 << 1),
    IRC_SUBST_ENV        = (1 << 2),
    IRC_SUBST_SHELL      = (1 << 3),
    IRC_SUBST_IRC_ATTRS  = (1 << 4),
    IRC_SUBST_SHELL_ATTRS = (1 << 5)
};
```

The following enumerators are available:

IRC_SUBST_DATE Allows date formatting. In that case the *time* field should not be 0.

IRC_SUBST_KEYWORDS Allows keywords substitutions, the fields *keywords* and *keywordsz* should be set but are still optional.

IRC_SUBST_ENV Allows environment variables substitution.

IRC_SUBST_IRC_ATTRS Allows IRC colors and attributes. This is mutually exclusive with **IRC_SUBST_SHELL_ATTRS**.

IRC_SUBST_SHELL_ATTRS Allows shell escape sequences for color and attributes. This is mutually exclusive with **IRC_SUBST_IRC_ATTRS**.

The **irc_subst()** functions convert the NUL terminated string *template* and store the result into the non-NUL *dst* buffer of *dtsz* max size. The function never writes more than *dtsz* bytes including the NUL terminator. The non-NUL *subst* argument controls the substitution parameters and allowed flags.

RETURN VALUES

The **irc_subst()** function return the number of bytes written in *dst* not including the NUL terminator. If an error occurs, -1 is returned and *errno* is set to indicate the error.

ERRORS

The function **irc_subst()** may set one of the following error:

[ENOMEM] Not enough space was available in *dst*.

[EINVAL] The source string contain invalid escape sequences or syntax.

EXAMPLES

This call will replace the keywords "fruit" and "dessert" along with the time formatting.

```
struct irc_subst subst = {
    .time = time(NULL),
    .flags = IRC_SUBST_KEYWORDS | IRC_SUBST_DATE,
    .keywordsz = 2,
    .keywords = (const struct irc_subst_keyword[]) {
        { "fruit",     "orange"      },
        { "dessert",   "tiramisu"   }
    }
};
char output[512];
const char *input = "You've eaten #{fruit} and #{dessert} on %H:%M";

irc_subst(output, sizeof (output), input, &subst);
/* output == You've eaten orange and tiramisu on 23:23 */
```

SEE ALSO

irccd-templates(7), libirccd(3)

AUTHORS

The **irccd** daemon was written by David Demelier <*markand@malikania.fr*>.