

NAME

ne_buffer_append, ne_buffer_zappend, ne_buffer_concat – append data to a string buffer

SYNOPSIS

```
#include <ne_string.h>
void ne_buffer_append(ne_buffer *buf, const char *string, size_t len);
void ne_buffer_zappend(ne_buffer *buf, const char *string);
void ne_buffer_concat(ne_buffer *buf, const char *str, ...);
```

DESCRIPTION

The **ne_buffer_append** and **ne_buffer_zappend** functions append a string to the end of a buffer; extending the buffer as necessary. The *len* passed to **ne_buffer_append** specifies the length of the string to append; there must be no NUL terminator in the first *len* bytes of the string.

ne_buffer_zappend must be passed a NUL-terminated string.

The **ne_buffer_concat** function takes a variable-length argument list following *str*; each argument must be a **char *** pointer to a NUL-terminated string. A NULL pointer must be given as the last argument to mark the end of the list. The strings (including *str*) are appended to the buffer in the order given. None of the strings passed to **ne_buffer_concat** are modified.

EXAMPLES

The following code will output "Hello, world. And goodbye.".

```
ne_buffer *buf = ne_buffer_create();
ne_buffer_zappend(buf, "Hello");
ne_buffer_concat(buf, ", world.", "And ", "goodbye.", NULL);
puts(buf->data);
ne_buffer_destroy(buf);
```

SEE ALSO

[ne_buffer](#), [ne_buffer_create](#), [ne_buffer_destroy](#)

AUTHOR

Joe Orton <neon AT lists DOT manyfish DOT co DOT uk>

Author.

COPYRIGHT

