

**NAME**

ne\_token, ne\_qtoken - string tokenizers

**SYNOPSIS**

```
#include <ne_string.h>
```

```
char *ne_token(char **str, char sep);
```

```
char *ne_qtoken(char **str, char sep, const char *quotes);
```

**DESCRIPTION**

**ne\_token** and **ne\_qtoken** tokenize the string at the location stored in the pointer *str*. Each time the function is called, it returns the next token, and modifies the *str* pointer to point to the remainder of the string, or NULL if there are no more tokens in the string. A token is delimited by the separator character *sep*; if **ne\_qtoken** is used any quoted segments of the string are skipped when searching for a separator. A quoted segment is enclosed in a pair of one of the characters given in the *quotes* string.

The string being tokenized is modified each time the tokenizing function is called; replacing the next separator character with a NUL terminator.

**EXAMPLES**

The following function prints out each token in a comma-separated string *list*, which is modified in-place:

```
static void splitter(char *list)
{
    do {
        printf("Token: %s\n", ne_token(&list, ','));
        while (list);
    }
}
```

**AUTHOR**

**Joe Orton**

Author.

**COPYRIGHT**