NAME

ares_parse_txt_reply − Parse a reply to a DNS query of type TXT

SYNOPSIS

#include <ares.h>

int ares_parse_txt_reply(const unsigned char* *abuf*, **int** *alen*,
struct ares_txt_reply *****txt_out*);**

int ares_parse_txt_reply_ext(const unsigned char* *abuf*, **int** *alen*,
struct ares_txt_ext *****txt_out*);**

DESCRIPTION

The **ares_parse_txt_reply** (**ares_parse_txt_reply_ext**) function parses the response to a query of type TXT into a linked list (one element per sub-string) of *struct ares_txt_reply* (*struct ares_txt_ext*) The parameters *abuf* and *alen* give the contents of the response. The result is stored in allocated memory and a pointer to it stored into the variable pointed to by *txt_out*. It is the caller's responsibility to free the resulting *txt_out* structure when it is no longer needed using the function **ares_free_data**

The structure *ares_txt_reply* contains the following fields:

```
struct ares_txt_reply {
  struct ares_txt_reply  *next;
  unsigned int  length;
  unsigned char *txt;
};
```

The structure *ares_txt_ext* contains the following fields:

```
struct ares_txt_ext {
  struct ares_txt_ext  *next;
  unsigned int  length;
  unsigned char *txt;
  unsigned char record_start;
};
```

The *record_start* field in *struct ares_txt_ext* is 1 if this structure is a start of a TXT record, and 0 if the structure is a continuation of a previous record. The linked list of the *struct ares_txt_ext* will have at least one item with *record_start* equal to 1, and may have some items with *record_start* equal to 0 between them.

These sequences of *struct ares_txt_ext* (starting from the item with *record_start* equal to 1, and ending right before the record start item) may be treated as either components of a single TXT record or as a multiparted TXT record, depending on particular use case.

RETURN VALUES

ares_parse_txt_reply (**ares_parse_txt_reply_ext**) can return any of the following values:

ARES_SUCCESS
The response was successfully parsed.

ARES_EBADRESP
The response was malformatted.

ARES_ENODATA
The response did not contain an answer to the query.

ARES_ENOMEM
Memory was exhausted.

AVAILABILITY

This function was first introduced in c-ares version 1.7.0.

SEE ALSO

ares_query(3) **ares_free_data**(3)

AUTHOR

Written by Jakub Hrozek <jhrozek@redhat.com>, on behalf of Red Hat, Inc http://www.redhat.com

Amended by Fedor Indutny <fedor@indutny.com>, on behalf of PayPal, Inc https://www.paypal.com