

**NAME**

`ares_mkquery` – Compose a single-question DNS query buffer

**SYNOPSIS**

```
#include <ares.h>
```

```
int ares_mkquery(const char *name, int dnsclass, int type,  
unsigned short id, int rd, unsigned char **buf,  
int *buflen)
```

**DESCRIPTION**

The `ares_mkquery` function composes a DNS query with a single question. The parameter *name* gives the query name as a NUL-terminated C string of period-separated labels optionally ending with a period; periods and backslashes within a label must be escaped with a backslash. The parameters *dnsclass* and *type* give the class and type of the query using the values defined in `<arpa/nameser.h>`. The parameter *id* gives a 16-bit identifier for the query. The parameter *rd* should be nonzero if recursion is desired, zero if not. The query will be placed in an allocated buffer, a pointer to which will be stored in the variable pointed to by *buf*, and the length of which will be stored in the variable pointed to by *buflen*. It is the caller's responsibility to free this buffer using `ares_free_string(3)` when it is no longer needed.

**RETURN VALUES**

`ares_mkquery` can return any of the following values:

**ARES\_SUCCESS**

Construction of the DNS query succeeded.

**ARES\_EBADNAME**

The query name *name* could not be encoded as a domain name, either because it contained a zero-length label or because it contained a label of more than 63 characters.

**ARES\_ENOMEM**

Memory was exhausted.

**SEE ALSO**

`ares_expand_name(3)`, `ares_free_string(3)`

**AUTHOR**

Greg Hudson, MIT Information Systems  
Copyright 1998, 2000 by the Massachusetts Institute of Technology.