

NAME

ares_dns_record_rr_add, ares_dns_record_rr_cnt, ares_dns_record_rr_del, ares_dns_record_rr_get, ares_dns_record_rr_get_const, ares_dns_rr_get_addr, ares_dns_rr_get_addr6, ares_dns_rr_get_bin, ares_dns_rr_get_abin_cnt, ares_dns_rr_get_abin, ares_dns_rr_get_class, ares_dns_rr_get_name, ares_dns_rr_get_opt, ares_dns_rr_get_opt_byid, ares_dns_rr_get_opt_cnt, ares_dns_rr_get_str, ares_dns_rr_get_ttl, ares_dns_rr_get_type, ares_dns_rr_get_u16, ares_dns_rr_get_u32, ares_dns_rr_get_u8, ares_dns_rr_key_t, ares_dns_rr_set_addr, ares_dns_rr_set_addr6, ares_dns_rr_set_bin, ares_dns_rr_set_abin, ares_dns_rr_del_abin, ares_dns_rr_set_opt, ares_dns_rr_set_str, ares_dns_rr_set_u16, ares_dns_rr_set_u32, ares_dns_rr_set_u8, ares_dns_section_t, ares_tlsa_match_t, ares_tlsa_selector_t, ares_tlsa_usage_t – DNS Resource Record creating, reading, and writing functions.

SYNOPSIS

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

```
size_t ares_dns_record_rr_cnt(const ares_dns_record_t *dnsrec,
                             ares_dns_section_t sect);

ares_status_t ares_dns_record_rr_add(ares_dns_rr_t **rr_out,
                                     ares_dns_record_t *dnsrec,
                                     ares_dns_section_t sect,
                                     const char *name,
                                     ares_dns_rec_type_t type,
                                     ares_dns_class_t rclass,
                                     unsigned int ttl);

ares_dns_rr_t *ares_dns_record_rr_get(ares_dns_record_t *dnsrec,
                                       ares_dns_section_t sect,
                                       size_t idx);

const ares_dns_rr_t *ares_dns_record_rr_get_const(const ares_dns_record_t *dnsrec,
                                                  ares_dns_section_t sect,
                                                  size_t idx);

ares_status_t ares_dns_record_rr_del(ares_dns_record_t *dnsrec,
                                     ares_dns_section_t sect,
                                     size_t idx);

const char *ares_dns_rr_get_name(const ares_dns_rr_t *rr);

ares_dns_rec_type_t ares_dns_rr_get_type(const ares_dns_rr_t *rr);

ares_dns_class_t ares_dns_rr_get_class(const ares_dns_rr_t *rr);

unsigned int ares_dns_rr_get_ttl(const ares_dns_rr_t *rr);

ares_status_t ares_dns_rr_set_addr(ares_dns_rr_t *dns_rr,
                                   ares_dns_rr_key_t key,
                                   const struct in_addr *addr);

ares_status_t ares_dns_rr_set_addr6(ares_dns_rr_t *dns_rr,
                                    ares_dns_rr_key_t key,
                                    const struct in6_addr *addr);

ares_status_t ares_dns_rr_set_str(ares_dns_rr_t *dns_rr,
                                  ares_dns_rr_key_t key,
```

```

        const char    *val);

ares_status_t ares_dns_rr_set_u8(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        unsigned char    val);

ares_status_t ares_dns_rr_set_u16(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        unsigned short    val);

ares_status_t ares_dns_rr_set_u32(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        unsigned int    val);

ares_status_t ares_dns_rr_set_bin(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        const unsigned char *val,
        size_t    len);

ares_status_t ares_dns_rr_add_abin(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        const unsigned char *val,
        size_t len);

ares_status_t ares_dns_rr_del_abin(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        size_t idx);

ares_status_t ares_dns_rr_set_opt(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        unsigned short    opt,
        const unsigned char *val,
        size_t    val_len);

ares_status_t ares_dns_rr_del_opt_byid(ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key,
        unsigned short    opt);

const struct in_addr *ares_dns_rr_get_addr(const ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key);

const struct ares_in6_addr *ares_dns_rr_get_addr6(const ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key);

const char *ares_dns_rr_get_str(const ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key);

unsigned char ares_dns_rr_get_u8(const ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key);

unsigned short ares_dns_rr_get_u16(const ares_dns_rr_t *dns_rr,
        ares_dns_rr_key_t key);

unsigned int ares_dns_rr_get_u32(const ares_dns_rr_t *dns_rr,

```

```

ares_dns_rr_key_t key);

const unsigned char *ares_dns_rr_get_bin(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key,
size_t *len);

size_t ares_dns_rr_get_abin_cnt(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key);

const unsigned char *ares_dns_rr_get_abin(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key, size_t idx,
size_t *len);

size_t ares_dns_rr_get_opt_cnt(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key);

unsigned short ares_dns_rr_get_opt(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key,
size_t idx,
const unsigned char **val,
size_t *val_len);

ares_bool_t ares_dns_rr_get_opt_byid(const ares_dns_rr_t *dns_rr,
ares_dns_rr_key_t key,
unsigned short opt,
const unsigned char **val,
size_t *val_len);

```

ENUMERATIONS

ares_dns_section_t - DNS RR section types:

ARES_SECTION_ANSWER - Answer section

ARES_SECTION_AUTHORITY - Authority section

ARES_SECTION_ADDITIONAL - Additional Information section

ares_dns_rr_key_t - Keys used for handling RR record parameters:

ARES_RR_A_ADDR - A Record. Address. Datatype: *ARES_DATATYPE_INADDR*

ARES_RR_NS_NSDDNAME - NS Record. Name. Datatype: *ARES_DATATYPE_NAME*

ARES_RR_CNAME_CNAME - CNAME Record. CName. Datatype: *ARES_DATATYPE_NAME*

ARES_RR_SOA_MNAME - SOA Record. MNAME, Primary Source of Data. Datatype: *ARES_DATATYPE_NAME*

ARES_RR_SOA_RNAME - SOA Record. RNAME, Mailbox of person responsible. Datatype: *ARES_DATATYPE_NAME*

ARES_RR_SOA_SERIAL - SOA Record. Serial, version. Datatype: *ARES_DATATYPE_U32*

ARES_RR_SOA_REFRESH - SOA Record. Refresh, zone refresh interval. Datatype: *ARES_DATATYPE_U32*

ARES_RR_SOA_RETRY - SOA Record. Retry, failed refresh retry interval. Datatype: *ARES_DATATYPE_U32*

ARES_RR_SOA_EXPIRE - SOA Record. Expire, upper limit on authority. Datatype: *ARES_DATATYPE_U32*

ARES_RR_SOA_MINIMUM - SOA Record. Minimum, RR TTL. Datatype: *ARES_DATATYPE_U32*

ARES_RR_PTR_DNAME - PTR Record. DNAME, pointer domain. Datatype: *ARES_DATATYPE_NAME*

ARES_RR_HINFO_CPU - HINFO Record. CPU. Datatype: *ARES_DATATYPE_STR*

ARES_RR_HINFO_OS - HINFO Record. OS. Datatype: *ARES_DATATYPE_STR*
ARES_RR_MX_PREFERENCE - MX Record. Preference. Datatype: *ARES_DATATYPE_U16*
ARES_RR_MX_EXCHANGE - MX Record. Exchange, domain. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_TXT_DATA - TXT Record. Data. Datatype: *ARES_DATATYPE_ABINP*
ARES_RR_SIG_TYPE_COVERED - SIG Record. Type Covered. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SIG_ALGORITHM - SIG Record. Algorithm. Datatype: *ARES_DATATYPE_U8*
ARES_RR_SIG_LABELS - SIG Record. Labels. Datatype: *ARES_DATATYPE_U8*
ARES_RR_SIG_ORIGINAL_TTL - SIG Record. Original TTL. Datatype: *ARES_DATATYPE_U32*
ARES_RR_SIG_EXPIRATION - SIG Record. Signature Expiration. Datatype: *ARES_DATATYPE_U32*
ARES_RR_SIG_INCEPTION - SIG Record. Signature Inception. Datatype: *ARES_DATATYPE_U32*
ARES_RR_SIG_KEY_TAG - SIG Record. Key Tag. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SIG_SIGNERS_NAME - SIG Record. Signer's Name. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_SIG_SIGNATURE - SIG Record. Signature. Datatype: *ARES_DATATYPE_BIN*
ARES_RR_AAAA_ADDR - AAAA Record. Address. Datatype: *ARES_DATATYPE_INADDR6*
ARES_RR_SRV_PRIORITY - SRV Record. Priority. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SRV_WEIGHT - SRV Record. Weight. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SRV_PORT - SRV Record. Port. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SRV_TARGET - SRV Record. Target domain. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_NAPTR_ORDER - NAPTR Record. Order. Datatype: *ARES_DATATYPE_U16*
ARES_RR_NAPTR_PREFERENCE - NAPTR Record. Preference. Datatype: *ARES_DATATYPE_U16*
ARES_RR_NAPTR_FLAGS - NAPTR Record. Flags. Datatype: *ARES_DATATYPE_STR*
ARES_RR_NAPTR_SERVICES - NAPTR Record. Services. Datatype: *ARES_DATATYPE_STR*
ARES_RR_NAPTR_REGEXP - NAPTR Record. Regexp. Datatype: *ARES_DATATYPE_STR*
ARES_RR_NAPTR_REPLACEMENT - NAPTR Record. Replacement. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_OPT_UDP_SIZE - OPT Record. UDP Size. Datatype: *ARES_DATATYPE_U16*
ARES_RR_OPT_VERSION - OPT Record. Version. Datatype: *ARES_DATATYPE_U8*
ARES_RR_OPT_FLAGS - OPT Record. Flags. Datatype: *ARES_DATATYPE_U16*
ARES_RR_OPT_OPTIONS - OPT Record. Options. See *ares_opt_param_t*. Datatype: *ARES_DATATYPE_OPT*
ARES_RR_TLSA_CERT_USAGE - TLSA Record. Certificate Usage. See *ares_tlsa_usage_t*. Datatype: *ARES_DATATYPE_U8*
ARES_RR_TLSA_SELECTOR - TLSA Record. Selector. See *ares_tlsa_selector_t*. Datatype: *ARES_DATATYPE_U8*
ARES_RR_TLSA_MATCH - TLSA Record. Matching Type. See *ares_tlsa_match_t*. Datatype: *ARES_DATATYPE_U8*
ARES_RR_TLSA_DATA - TLSA Record. Certificate Association Data. Datatype: *ARES_DATATYPE_BIN*
ARES_RR_SVCB_PRIORITY - SVCB Record. SvcPriority. Datatype: *ARES_DATATYPE_U16*
ARES_RR_SVCB_TARGET - SVCB Record. TargetName. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_SVCB_PARAMS - SVCB Record. SvcParams. See *ares_svcb_param_t*. Datatype: *ARES_DATATYPE_OPT*
ARES_RR_HTTPS_PRIORITY - HTTPS Record. SvcPriority. Datatype: *ARES_DATATYPE_U16*
ARES_RR_HTTPS_TARGET - HTTPS Record. TargetName. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_HTTPS_PARAMS - HTTPS Record. SvcParams. See *ares_svcb_param_t*. Datatype: *ARES_DATATYPE_OPT*
ARES_RR_URI_PRIORITY - URI Record. Priority. Datatype: *ARES_DATATYPE_U16*
ARES_RR_URI_WEIGHT - URI Record. Weight. Datatype: *ARES_DATATYPE_U16*

ARES_RR_URI_TARGET - URI Record. Target domain. Datatype: *ARES_DATATYPE_NAME*
ARES_RR_CAA_CRITICAL - CAA Record. Critical flag. Datatype: *ARES_DATATYPE_U8*
ARES_RR_CAA_TAG - CAA Record. Tag/Property. Datatype: *ARES_DATATYPE_STR*
ARES_RR_CAA_VALUE - CAA Record. Value. Datatype: *ARES_DATATYPE_BINP*
ARES_RR_RAW_RR_TYPE - RAW Record. RR Type. Datatype: *ARES_DATATYPE_U16*
ARES_RR_RAW_RR_DATA - RAW Record. RR Data. Datatype: *ARES_DATATYPE_BIN*

ares_tlsa_usage_t - TLSA Record *ARES_RR_TLSA_CERT_USAGE* known values

ARES_TLSA_USAGE_CA - Certificate Usage 0. CA Constraint
ARES_TLSA_USAGE_SERVICE - Certificate Usage 1. Service Certificate Constraint
ARES_TLSA_USAGE_TRUSTANCHOR - Certificate Usage 2. Trust Anchor Assertion
ARES_TLSA_USAGE_DOMAIN - Certificate Usage 3. Domain-issued certificate

ares_tlsa_selector_t - TLSA Record *ARES_RR_TLSA_SELECTOR* known values:

ARES_TLSA_SELECTOR_FULL - Full Certificate
ARES_TLSA_SELECTOR_SUBJPUBKEYINFO - DER-encoded SubjectPublicKeyInfo

ares_tlsa_match_t - TLSA Record *ARES_RR_TLSA_MATCH* known values:

ARES_TLSA_MATCH_EXACT - Exact match
ARES_TLSA_MATCH_SHA256 - Sha256 match
ARES_TLSA_MATCH_SHA512 - Sha512 match

DESCRIPTION

The *ares_dns_record_rr_cnt(3)* function returns the number of resource records in the DNS record provided by the *dnsrec* parameter for the section provided in the *sect* parameter.

The *ares_dns_record_rr_add(3)* function adds a new resource record entry the the DNS record provided by the *dnsrec* parameter. The resulting resource record is stored into the variable pointed to by *rr_out*. The DNS section the resource record belongs to is specified by the *sect* parameter. The domain name associated with the resource record is specified by the *name* parameter, which can not be NULL but may be an empty string, or ".". The resource record type is specified in the *type* parameter, along with the DNS record class in the *rclass* parameter, and the Time To Live (TTL) in the *tll* parameter.

The *ares_dns_record_rr_get(3)* and *ares_dns_record_rr_get_const(3)* functions are used to retrieve the resource record pointer from the DNS record provided in the *dnsrec* parameter, for the resource record section provided in the *sect* parameter, for the specified index in the *idx* parameter. The index must be less than *ares_dns_record_rr_cnt(3)*. The former returns a writable pointer to the resource record, while the latter returns a read-only pointer to the resource record.

The *ares_dns_record_rr_del(3)* is used to delete a resource record from the DNS record specified in the *dnsrec* parameter. Its primary use is to remove a *ARES_REC_TYPE_OPT* record when needing to retry a query without EDNS support. The DNS RR section is specified via the *sect* parameter, and the index to remove is specified in the *idx* parameter. The index must be less than *ares_dns_record_rr_cnt(3)*.

The *ares_dns_rr_get_name(3)* function is used to retrieve the resource record domain name from the Resource Record pointer provided in the *rr* parameter.

The *ares_dns_rr_get_type(3)* function is used to retrieve the resource record type from the Resource Record pointer provided in the *rr* parameter.

The *ares_dns_rr_get_class(3)* function is used to retrieve the resource record class from the Resource Record pointer provided in the *rr* parameter.

The *ares_dns_rr_get_ttl(3)* function is used to retrieve the resource record class Time to Live (TTL) from the Resource Record pointer provided in the *rr* parameter.

The *ares_dns_rr_set_addr(3)* function is used to set an IPv4 address for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_INADDR*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *addr* parameter.

The *ares_dns_rr_set_addr6(3)* function is used to set an IPv6 address for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_INADDR6*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *addr* parameter.

The *ares_dns_rr_set_str(3)* function is used to set a string for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_STR* or *ARES_DATATYPE_NAME*. Most strings are limited to 255 bytes, however some records, such as a TXT record may allow longer as they are output as multiple strings. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter.

The *ares_dns_rr_set_u8(3)* function is used to set an 8bit unsigned value for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_U8*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter.

The *ares_dns_rr_set_u16(3)* function is used to set an 16bit unsigned value for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_U16*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter.

The *ares_dns_rr_set_u32(3)* function is used to set an 32bit unsigned value for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_U32*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter.

The *ares_dns_rr_set_bin(3)* function is used to set a binary value for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_BIN* or *ARES_DATATYPE_BINP*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter. And the associated value length is provided in the *len* parameter.

The *ares_dns_rr_add_abin(3)* function is used to append a binary value in the array for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_ABINP*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the value is provided in the *val* parameter. And the associated value length is provided in the *len* parameter.

The *ares_dns_rr_del_abin(3)* function is used to delete a binary value in the array for the associated resource record key/parameter when the datatype is *ARES_DATATYPE_ABINP*. The resource record to be modified is provided in the *dns_rr* parameter, the key/parameter is provided in the *key* parameter, and the index to remove is provided in the *idx* parameter.

The *ares_dns_rr_set_opt(3)* function is used to set option/parameter keys and values for the resource record when the datatype is *ARES_DATATYPE_OPT*. The resource record to be modified is provided in the

dns_rr parameter. The key/parameter is provided in the *key* parameter. The option/parameter value specific to the resource record is provided in the *opt* parameter, and this is left to the user to determine the appropriate value to use. Some known values may be provided by *ares_svcb_param_t* and *ares_opt_param_t* enumerations. The value for the option is always provided in binary form in *val* with length provided in *val_len*.

The *ares_dns_rr_del_opt_byid(3)* function is used to delete option/parameter keys and values for the resource record when the datatype is *ARES_DATATYPE_OPT*. The resource record to be modified is provided in the *dns_rr* parameter. The key/parameter is provided in the *key* parameter. The option/parameter value specific to the resource record is provided in the *opt* parameter. This function returns *ARES_SUCCESS* if the record is successfully removed, or *ARES_ENOTFOUND* if the record could not be found.

The *ares_dns_rr_get_addr(3)* function is used to retrieve the IPv4 address from the resource record when the datatype is *ARES_DATATYPE_INADDR*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_addr6(3)* function is used to retrieve the IPv6 address from the resource record when the datatype is *ARES_DATATYPE_INADDR6*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_str(3)* function is used to retrieve a string from the resource record when the datatype is *ARES_DATATYPE_STR* or *ARES_DATATYPE_NAME*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_u8(3)* function is used to retrieve an 8bit integer from the resource record when the datatype is *ARES_DATATYPE_U8*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_u16(3)* function is used to retrieve a 16bit integer from the resource record when the datatype is *ARES_DATATYPE_U16*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_u32(3)* function is used to retrieve a 32bit integer from the resource record when the datatype is *ARES_DATATYPE_U32*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_bin(3)* function is used to retrieve binary data from the resource record when the datatype is *ARES_DATATYPE_BIN* or *ARES_DATATYPE_BINP*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter, and length is stored into the variable pointed to by *len*.

The *ares_dns_rr_get_abin_cnt(3)* function is used to retrieve the count of the array of stored binary values from the resource record when the datatype is *ARES_DATATYPE_ABINP*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter.

The *ares_dns_rr_get_abin(3)* function is used to retrieve binary data from the resource record array when the datatype is *ARES_DATATYPE_ABINP*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter, and the index to retrieve from the array is provided by the *idx* parameter, and length is stored into the variable pointed to by *len*.

The *ares_dns_rr_get_opt_cnt(3)* function is used to retrieve the count of options/parameters associated with the resource record when the datatype is *ARES_DATATYPE_OPT*. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key*.

The *ares_dns_rr_get_opt(3)* function is used to retrieve binary option data from the resource record when the datatype is *ARES_DATATYPE_OPT* for the specified index. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter, the index to retrieve the option data from is provided in the *idx* parameter. The value is stored into the variable pointed to by *val* and length is stored into the variable pointed to by *val_len*.

The *ares_dns_rr_get_opt_byid(3)* function is used to retrieve binary option data from the resource record when the datatype is *ARES_DATATYPE_OPT* for the specified option identifier, if it exists. The resource record is provided in the *dns_rr* parameter and the key/parameter to retrieve is provided in the *key* parameter, the identifier to retrieve the option data from is provided in the *opt* parameter. The value is stored into the variable pointed to by *val* and length is stored into the variable pointed to by *val_len*.

RETURN VALUES

ares_dns_record_rr_cnt(3) and *ares_dns_rr_get_opt_cnt(3)* return the respective counts.

ares_dns_record_rr_add(3), *ares_dns_record_rr_del(3)*, *ares_dns_rr_set_addr(3)*, *ares_dns_rr_set_addr6(3)*, *ares_dns_rr_set_str(3)*, *ares_dns_rr_set_u8(3)*, *ares_dns_rr_set_u16(3)*, *ares_dns_rr_set_u32(3)*, *ares_dns_rr_set_bin(3)*, *ares_dns_rr_add_abin(3)*, *ares_dns_rr_del_abin(3)*, and *ares_dns_rr_set_opt(3)* all return an *ares_status_t* error code. **ARES_SUCCESS** is returned on success, **ARES_ENOMEM** is returned on out of memory, **ARES_EFORMERR** is returned on misuse.

ares_dns_rr_get_name(3), *ares_dns_rr_get_type(3)*, *ares_dns_rr_get_class(3)*, *ares_dns_rr_get_ttl(3)*, *ares_dns_rr_get_addr(3)*, *ares_dns_rr_get_addr6(3)*, *ares_dns_rr_get_str(3)*, *ares_dns_rr_get_u8(3)*, *ares_dns_rr_get_u16(3)*, *ares_dns_rr_get_u32(3)*, *ares_dns_rr_get_bin(3)*, *ares_dns_rr_get_abin_cnt(3)*, *ares_dns_rr_get_abin(3)*, *ares_dns_rr_get_opt(3)* all return their prescribed datatype values and in general can't fail except for misuse cases, in which a 0 (or NULL) may be returned, however 0 can also be a valid return value for most of these functions.

ares_dns_record_rr_get(3) and *ares_dns_record_rr_get_const(3)* will return the requested resource record pointer or NULL on failure (misuse).

ares_dns_rr_get_opt_byid(3) will return **ARES_TRUE** if the option was found, otherwise **ARES_FALSE** if not found (or misuse).

AVAILABILITY

These functions were first introduced in c-ares version 1.22.0.

SEE ALSO

ares_dns_mapping(3), *ares_dns_record(3)*, *ares_free_string(3)*