

NAME

ldns_rr, ldns_rr_class, ldns_rr_type, ldns_rr_compress, ldns_rr_list - types representing dns resource records

SYNOPSIS

```
#include <stdint.h>
#include <stdbool.h>

#include <ldns/ldns.h>
```

DESCRIPTION

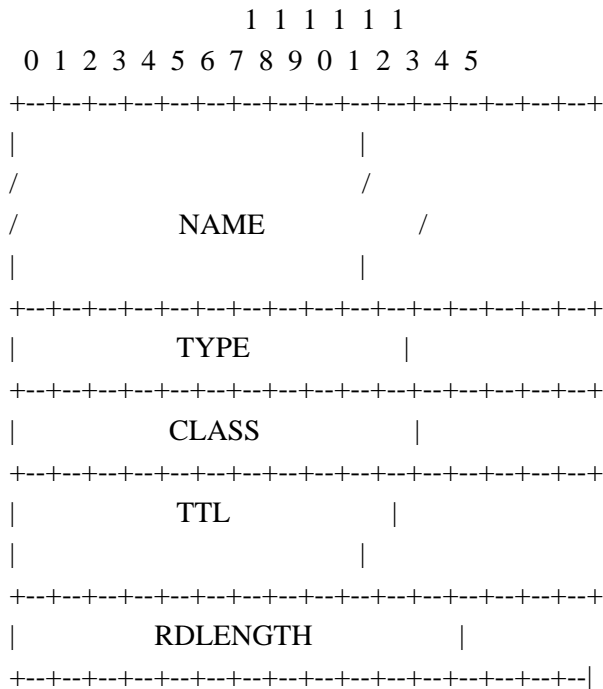
ldns_rr
Resource Record

This is the basic DNS element that contains actual data

From RFC1035:

<pre>
3.2.1. Format

All RRs have the same top level format shown below:



```

/          RDATA          /
/
+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

where:

NAME an owner name, i.e., the name of the node to which this resource record pertains.

TYPE two octets containing one of the RR TYPE codes.

CLASS two octets containing one of the RR CLASS codes.

TTL a 32 bit signed integer that specifies the time interval that the resource record may be cached before the source of the information should again be consulted. Zero values are interpreted to mean that the RR can only be used for the transaction in progress, and should not be cached. For example, SOA records are always distributed with a zero TTL to prohibit caching. Zero values can also be used for extremely volatile data.

RDLENGTH an unsigned 16 bit integer that specifies the length in octets of the RDATA field.

RDATA a variable length string of octets that describes the resource. The format of this information varies according to the TYPE and CLASS of the resource record.

</pre>

The actual amount and type of rdata fields depend on the RR type of the RR, and can be found by using `\ref ldns_rr_descriptor` functions.

```
struct ldns_struct_rr
```

```
{
```

Owner name, uncompressed:

```
ldns_rdf        *_owner;
```

Time to live:

```
uint32_t        _ttl;
```

Number of data fields:

```
size_t         _rd_count;
```

the type of the RR. A, MX etc.:

```
ldns_rr_type  _rr_type;
```

Class of the resource record.:

```
ldns_rr_class  _rr_class;
```

```
/* everything in the rdata is in network order */
```

The array of rdata's:

```
ldns_rdf      **_rdata_fields;
```

```
/** question rr [it would be nicer if thous is after _rd_count]
```

```
    ABI change: Fix this in next major release
```

```
*/
```

```
bool          _rr_question;
```

```
};
```

```
typedef struct ldns_struct_rr ldns_rr;
```

ldns_rr_class

The different RR classes.

```
enum ldns_enum_rr_class
```

```
{
```

the Internet:

```
LDNS_RR_CLASS_IN      = 1,
```

Chaos class:

```
LDNS_RR_CLASS_CH      = 3,
```

Hesiod (Dyer 87):

```
LDNS_RR_CLASS_HS      = 4,
```

None class, dynamic update:

```
LDNS_RR_CLASS_NONE    = 254,
```

Any class:

```
LDNS_RR_CLASS_ANY     = 255,
```

```
LDNS_RR_CLASS_FIRST   = 0,
```

```
LDNS_RR_CLASS_LAST    = 65535,
```

```
LDNS_RR_CLASS_COUNT   = LDNS_RR_CLASS_LAST -
```

```
LDNS_RR_CLASS_FIRST + 1
```

```
};
```

```
typedef enum ldns_enum_rr_class ldns_rr_class;
```

ldns_rr_type

The different RR types.

```
enum ldns_enum_rr_type
```

```
{
```

a host address:

LDNS_RR_TYPE_A = 1,

an authoritative name server:

LDNS_RR_TYPE_NS = 2,

a mail destination (Obsolete - use MX):

LDNS_RR_TYPE_MD = 3,

a mail forwarder (Obsolete - use MX):

LDNS_RR_TYPE_MF = 4,

the canonical name for an alias:

LDNS_RR_TYPE_CNAME = 5,

marks the start of a zone of authority:

LDNS_RR_TYPE_SOA = 6,

a mailbox domain name (EXPERIMENTAL):

LDNS_RR_TYPE_MB = 7,

a mail group member (EXPERIMENTAL):

LDNS_RR_TYPE_MG = 8,

a mail rename domain name (EXPERIMENTAL):

LDNS_RR_TYPE_MR = 9,

a null RR (EXPERIMENTAL):

LDNS_RR_TYPE_NULL = 10,

a well known service description:

LDNS_RR_TYPE_WKS = 11,

a domain name pointer:

LDNS_RR_TYPE_PTR = 12,

host information:

LDNS_RR_TYPE_HINFO = 13,

mailbox or mail list information:

LDNS_RR_TYPE_MINFO = 14,

mail exchange:

LDNS_RR_TYPE_MX = 15,

text strings:

LDNS_RR_TYPE_TXT = 16,

RFC1183:

LDNS_RR_TYPE_RP = 17,

RFC1183:

LDNS_RR_TYPE_AFSDB = 18,

RFC1183:

LDNS_RR_TYPE_X25 = 19,

RFC1183:

LDNS_RR_TYPE_ISDN = 20,

RFC1183:

LDNS_RR_TYPE_RT = 21,

RFC1706:

LDNS_RR_TYPE_NSAP = 22,

RFC1348:

LDNS_RR_TYPE_NSAP_PTR = 23,

2535typecode:

LDNS_RR_TYPE_SIG = 24,

2535typecode:

LDNS_RR_TYPE_KEY = 25,

RFC2163:

LDNS_RR_TYPE_PX = 26,

RFC1712:

LDNS_RR_TYPE_GPOS = 27,

ipv6 address:

LDNS_RR_TYPE_AAAA = 28,

LOC record RFC1876:

LDNS_RR_TYPE_LOC = 29,

2535typecode:

LDNS_RR_TYPE_NXT = 30,

draft-ietf-nimrod-dns-01.txt:

LDNS_RR_TYPE_EID = 31,

draft-ietf-nimrod-dns-01.txt:

LDNS_RR_TYPE_NIMLOC = 32,

SRV record RFC2782:

LDNS_RR_TYPE_SRV = 33,

<http://www.jhsoft.com/rfc/af-saa-0069.000.rtf>:

LDNS_RR_TYPE_ATMA = 34,

RFC2915:

LDNS_RR_TYPE_NAPTR = 35,

RFC2230:

LDNS_RR_TYPE_KX = 36,

RFC2538:

LDNS_RR_TYPE_CERT = 37,

RFC2874:

LDNS_RR_TYPE_A6 = 38,

RFC2672:

LDNS_RR_TYPE_DNAME = 39,

dnsind-kitchen-sink-02.txt:

LDNS_RR_TYPE_SINK = 40,

OPT record RFC 6891:

LDNS_RR_TYPE_OPT = 41,

RFC3123:

LDNS_RR_TYPE_APL = 42,

RFC4034, RFC3658:

LDNS_RR_TYPE_DS = 43,

SSH Key Fingerprint:

LDNS_RR_TYPE_SSHFP = 44, /* RFC 4255 */

IPsec Key:

LDNS_RR_TYPE_IPSECKEY = 45, /* RFC 4025 */

DNSSEC:

LDNS_RR_TYPE_RRSIG = 46, /* RFC 4034 */

LDNS_RR_TYPE_NSEC = 47, /* RFC 4034 */

LDNS_RR_TYPE_DNSKEY = 48, /* RFC 4034 */

LDNS_RR_TYPE_DHCID = 49, /* RFC 4701 */

/* NSEC3 */

LDNS_RR_TYPE_NSEC3 = 50, /* RFC 5155 */

LDNS_RR_TYPE_NSEC3PARAM = 51, /* RFC 5155 */

LDNS_RR_TYPE_NSEC3PARAMS = 51,

LDNS_RR_TYPE_TLSA = 52, /* RFC 6698 */

LDNS_RR_TYPE_SMIMEA = 53, /* RFC 8162 */

LDNS_RR_TYPE_HIP = 55, /* RFC 5205 */

draft-reid-dnsex-t-zs:

LDNS_RR_TYPE_NINFO = 56,

draft-reid-dnsex-t-rkey:

LDNS_RR_TYPE_RKEY = 57,

draft-ietf-dnsop-trust-history:

LDNS_RR_TYPE_TALINK = 58,

LDNS_RR_TYPE_CDS = 59, /* RFC 7344 */

LDNS_RR_TYPE_CDNSKEY = 60, /* RFC 7344 */

LDNS_RR_TYPE_OPENPGPKEY = 61, /* RFC 7929 */

LDNS_RR_TYPE_CSYSN = 62, /* RFC 7477 */

LDNS_RR_TYPE_ZONEMD = 63, /* draft-ietf-dnsop-dns-zone-digest */

LDNS_RR_TYPE_SVCB = 64, /* draft-ietf-dnsop-svcb-https */

LDNS_RR_TYPE_HTTPS = 65, /* draft-ietf-dnsop-svcb-https */

LDNS_RR_TYPE_SPF = 99, /* RFC 4408 */

LDNS_RR_TYPE_UINFO = 100,
LDNS_RR_TYPE_UID = 101,
LDNS_RR_TYPE_GID = 102,
LDNS_RR_TYPE_UNSPEC = 103,

LDNS_RR_TYPE_NID = 104, /* RFC 6742 */
LDNS_RR_TYPE_L32 = 105, /* RFC 6742 */
LDNS_RR_TYPE_L64 = 106, /* RFC 6742 */
LDNS_RR_TYPE_LP = 107, /* RFC 6742 */

LDNS_RR_TYPE_EUI48 = 108, /* RFC 7043 */
LDNS_RR_TYPE_EUI64 = 109, /* RFC 7043 */

LDNS_RR_TYPE_TKEY = 249, /* RFC 2930 */
LDNS_RR_TYPE_TSIG = 250,
LDNS_RR_TYPE_IXFR = 251,
LDNS_RR_TYPE_AXFR = 252,

A request for mailbox-related records (MB, MG or MR):

LDNS_RR_TYPE_MAILB = 253,

A request for mail agent RRs (Obsolete - see MX):

LDNS_RR_TYPE_MAILA = 254,

any type (wildcard):

LDNS_RR_TYPE_ANY = 255,
LDNS_RR_TYPE_URI = 256, /* RFC 7553 */
LDNS_RR_TYPE_CAA = 257, /* RFC 6844 */
LDNS_RR_TYPE_AVC = 258, /* Cisco's DNS-AS RR, see www.dns-as.org */
LDNS_RR_TYPE_DOA = 259, /* draft-durand-doa-over-dns */

draft-ietf-mboned-driad-amt-discovery *:

LDNS_RR_TYPE_AMTRELAY = 260,

DNSSEC Trust Authorities:

LDNS_RR_TYPE_TA = 32768,
/* RFC 4431, 5074, DNSSEC Lookaside Validation */
LDNS_RR_TYPE_DLV = 32769,

/* type codes from nsec3 experimental phase

LDNS_RR_TYPE_NSEC3 = 65324,
LDNS_RR_TYPE_NSEC3PARAMS = 65325, */
LDNS_RR_TYPE_FIRST = 0,

```

        LDNS_RR_TYPE_LAST = 65535,
        LDNS_RR_TYPE_COUNT = LDNS_RR_TYPE_LAST - LDNS_RR_TYPE_FIRST + 1
    };
    typedef enum ldns_enum_rr_type ldns_rr_type;

```

ldns_rr_compress

Used to specify whether compression is allowed.

```

enum ldns_enum_rr_compress
{
    compression is allowed:
    LDNS_RR_COMPRESS,
    LDNS_RR_NO_COMPRESS
};
typedef enum ldns_enum_rr_compress ldns_rr_compress;

```

ldns_rr_list

List or Set of Resource Records

Contains a list of rr's

No official RFC-like checks are made

```

struct ldns_struct_rr_list
{
    size_t _rr_count;
    size_t _rr_capacity;
    ldns_rr **_rrs;
};
typedef struct ldns_struct_rr_list ldns_rr_list;

```

AUTHOR

The ldns team at NLnet Labs.

REPORTING BUGS

Please report bugs to ldns-team@nlnetlabs.nl or in our bugzilla at <http://www.nlnetlabs.nl/bugs/index.html>

COPYRIGHT

Copyright (c) 2004 - 2006 NLnet Labs.

Licensed under the BSD License. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

SEE ALSO

ldns_rr_new, *ldns_rr_new_frm_type*, *ldns_rr_new_frm_str*, *ldns_rr_new_frm_fp*, *ldns_rr_free*, *ldns_rr_print*, *ldns_rr_set_owner*, *ldns_rr_set_ttl*, *ldns_rr_set_type*, *ldns_rr_set_rd_count*, *ldns_rr_set_class*, *ldns_rr_set_rdf*, *ldns_rr_push_rdf*, *ldns_rr_pop_rdf*, *ldns_rr_rdf*, *ldns_rr_owner*, *ldns_rr_rd_count*, *ldns_rr_ttl*, *ldns_rr_get_class*, *ldns_rr_list_rr_count*, *ldns_rr_list_set_rr_count*, *ldns_rr_list_new*, *ldns_rr_list_free*, *ldns_rr_list_cat*, *ldns_rr_list_push_rr*, *ldns_rr_list_pop_rr*, *ldns_is_rrset*, *ldns_rr_set_push_rr*, *ldns_rr_set_pop_rr*, *ldns_get_rr_class_by_name*, *ldns_get_rr_type_by_name*, *ldns_rr_list_clone*, *ldns_rr_list_sort*, *ldns_rr_compare*, *ldns_rr_compare_ds*, *ldns_rr_uncompressed_size*, *ldns_rr2canonical*, *ldns_rr_label_count*, *ldns_is_rrset*, *ldns_rr_descriptor*, *ldns_rr_descript*. And **perldoc Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

REMARKS

This manpage was automatically generated from the ldns source code.