

**NAME**

ldns\_rdf, ldns\_rdf\_type - rdata field type

**SYNOPSIS**

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

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

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

**DESCRIPTION**

*ldns\_rdf*

Resource record data field.

The data is a network ordered array of bytes, which size is specified by the (16-bit) size field. To correctly parse it, use the type specified in the (16-bit) type field with a value from \ref ldns\_rdf\_type.

```
struct ldns_struct_rdf
```

```
{
```

**The size of the data (in octets):**

```
size_t _size;
```

**The type of the data:**

```
ldns_rdf_type _type;
```

**Pointer to the data (raw octets):**

```
void *_data;
```

```
};
```

```
typedef struct ldns_struct_rdf ldns_rdf;
```

*ldns\_rdf\_type*

The different types of RDATA fields.

```
enum ldns_enum_rdf_type
```

```
{
```

**none:**

```
LDNS_RDF_TYPE_NONE,
```

**domain name:**

```
LDNS_RDF_TYPE_DNAME,
```

**8 bits:**

```
LDNS_RDF_TYPE_INT8,
```

**16 bits:**

```
LDNS_RDF_TYPE_INT16,
```

**32 bits:**  
LDNS\_RDF\_TYPE\_INT32,

**A record:**  
LDNS\_RDF\_TYPE\_A,

**AAAA record:**  
LDNS\_RDF\_TYPE\_AAAA,

**txt string:**  
LDNS\_RDF\_TYPE\_STR,

**apl data:**  
LDNS\_RDF\_TYPE\_APL,

**b32 string:**  
LDNS\_RDF\_TYPE\_B32\_EXT,

**b64 string:**  
LDNS\_RDF\_TYPE\_B64,

**hex string:**  
LDNS\_RDF\_TYPE\_HEX,

**nsec type codes:**  
LDNS\_RDF\_TYPE\_NSEC,

**a RR type:**  
LDNS\_RDF\_TYPE\_TYPE,

**a class:**  
LDNS\_RDF\_TYPE\_CLASS,

**certificate algorithm:**  
LDNS\_RDF\_TYPE\_CERT\_ALG,

**a key algorithm:**  
LDNS\_RDF\_TYPE\_ALG,

**unknown types:**  
LDNS\_RDF\_TYPE\_UNKNOWN,

**time (32 bits):**  
LDNS\_RDF\_TYPE\_TIME,

**period:**  
LDNS\_RDF\_TYPE\_PERIOD,

**tsig time 48 bits:**  
LDNS\_RDF\_TYPE\_TSIGTIME,

/\*\* Represents the Public Key Algorithm, HIT and Public Key fields  
for the HIP RR types. A HIP specific rdf type is used because of  
the unusual layout in wireformat (see RFC 5205 Section 5) \*/  
LDNS\_RDF\_TYPE\_HIP,

/\*\* variable length any type rdata where the length  
is specified by the first 2 bytes \*/

LDNS\_RDF\_TYPE\_INT16\_DATA,

**protocol and port bitmaps:**

LDNS\_RDF\_TYPE\_SERVICE,

**location data:**

LDNS\_RDF\_TYPE\_LOC,

**well known services:**

LDNS\_RDF\_TYPE\_WKS,

**NSAP:**

LDNS\_RDF\_TYPE\_NSAP,

**ATMA:**

LDNS\_RDF\_TYPE\_ATMA,

**IPSECKEY:**

LDNS\_RDF\_TYPE\_IPSECKEY,

**nsec3 hash salt:**

LDNS\_RDF\_TYPE\_NSEC3\_SALT,

**nsec3 base32 string (with length byte on wire):**

LDNS\_RDF\_TYPE\_NSEC3\_NEXT\_OWNER,

/\*\* 4 shorts represented as 4 \* 16 bit hex numbers

\* separated by colons. For NID and L64.

\*/

LDNS\_RDF\_TYPE\_ILNP64,

**6 \* 8 bit hex numbers separated by dashes. For EUI48.:**

LDNS\_RDF\_TYPE\_EUI48,

**8 \* 8 bit hex numbers separated by dashes. For EUI64.:**

LDNS\_RDF\_TYPE\_EUI64,

/\*\* A non-zero sequence of US-ASCII letters and numbers in lower case.

\* For CAA.

\*/

LDNS\_RDF\_TYPE\_TAG,

/\*\* A <character-string> encoding of the value field as specified

\* [RFC1035], Section 5.1., encoded as remaining rdata.

\* For CAA.

\*/

LDNS\_RDF\_TYPE\_LONG\_STR,

/\*\* Since RFC7218 TLSA records can be given with mnemonics,

```

* hence these rdata field types. But as with DNSKEYs, the output
* is always numeric.
*/
LDNS_RDF_TYPE_CERTIFICATE_USAGE,
LDNS_RDF_TYPE_SELECTOR,
LDNS_RDF_TYPE_MATCHING_TYPE,

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

draft-ietf-dnsop-svcb-https *:
LDNS_RDF_TYPE_SVCPARAMS,

/* Aliases */
LDNS_RDF_TYPE_BITMAP = LDNS_RDF_TYPE_NSEC
};
typedef enum ldns_enum_rdf_type ldns_rdf_type;

```

**AUTHOR**

The ldns team at NLnet Labs.

**REPORTING BUGS**

Please report bugs to [ldns-team@nlnetlabs.nl](mailto: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\_rdf\_set\_size, ldns\_rdf\_set\_type, ldns\_rdf\_set\_data, ldns\_rdf\_size, ldns\_rdf\_get\_type, ldns\_rdf\_data, ldns\_rdf\_compare, ldns\_rdf\_new, ldns\_rdf\_clone, ldns\_rdf\_new\_frm\_data, ldns\_rdf\_new\_frm\_str, ldns\_rdf\_new\_frm\_fp, ldns\_rdf\_free, ldns\_rdf\_deep\_free, ldns\_rdf\_print, ldns\_native2rdf\_int8, ldns\_native2rdf\_int16, ldns\_native2rdf\_int32, ldns\_native2rdf\_int16\_data, ldns\_rdf2native\_int8, ldns\_rdf2native\_int16, ldns\_rdf2native\_int32, ldns\_rdf2native\_sockaddr\_storage,*

*ldns\_rdf2native\_time\_t, ldns\_native2rdf\_int8, ldns\_native2rdf\_int16, ldns\_native2rdf\_int32, ldns\_native2rdf\_int16\_data, ldns\_rdf2native\_int8, ldns\_rdf2native\_int16, ldns\_rdf2native\_int32, ldns\_rdf2native\_sockaddr\_storage, ldns\_rdf2native\_time\_t, ldns\_native2rdf\_int8, ldns\_native2rdf\_int16, ldns\_native2rdf\_int32, ldns\_native2rdf\_int16\_data, ldns\_rdf2native\_int8, ldns\_rdf2native\_int16, ldns\_rdf2native\_int32, ldns\_rdf2native\_sockaddr\_storage, ldns\_rdf2native\_time\_t.* And **perldoc Net::DNS, RFC1034, RFC1035, RFC4033, RFC4034** and **RFC4035**.

## REMARKS

This manpage was automatically generated from the ldns source code.