

NAME

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 creation, destruction and printing

SYNOPSIS

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

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

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

```
ldns_rr* ldns_rr_new(void);
```

```
ldns_rr* ldns_rr_new_frm_type(ldns_rr_type t);
```

```
ldns_status ldns_rr_new_frm_str(ldns_rr **n, const char *str, uint32_t default_ttl, const ldns_rdf *origin, ldns_rdf **prev);
```

```
ldns_status ldns_rr_new_frm_fp(ldns_rr **rr, FILE *fp, uint32_t *default_ttl, ldns_rdf **origin, ldns_rdf **prev);
```

```
void ldns_rr_free(ldns_rr *rr);
```

```
void ldns_rr_print(FILE *output, const ldns_rr *rr);
```

DESCRIPTION

ldns_rr_new() creates a new rr structure.

Returns ldns_rr *

ldns_rr_new_frm_type() creates a new rr structure, based on the given type. alloc enough space to hold all the rdf's

ldns_rr_new_frm_str() creates an rr from a string. The string should be a fully filled-in rr, like
 ownername <space> TTL <space> CLASS <space> TYPE <space>
 RDATA.

n: the rr to return

str: the string to convert

default_ttl: default ttl value for the rr. If 0 DEF_TTL will be used

origin: when the owner is relative add this. The caller must ldns_rdf_deep_free it.

prev: the previous ownername. if this value is not NULL, the function overwrites this with the

ownername found in this string. The caller must then `ldns_rdf_deep_free` it.
Returns a status msg describing an error or `LDNS_STATUS_OK`

ldns_rr_new_frm_fp() creates a new rr from a file containing a string.

rr: the new rr

fp: the file pointer to use

default_ttl: pointer to a default ttl for the rr. If `NULL` `DEF_TTL` will be used the pointer will be updated if the file contains a `$TTL` directive

origin: when the owner is relative add this the pointer will be updated if the file contains a `$ORIGIN` directive The caller must `ldns_rdf_deep_free` it.

prev: when the owner is whitespaces use this as the * ownername the pointer will be updated after the call The caller must `ldns_rdf_deep_free` it.

Returns a `ldns_status` with an error or `LDNS_STATUS_OK`

ldns_rr_free() frees an RR structure

***rr**: the RR to be freed

Returns void

ldns_rr_print() Prints the data in the resource record to the given file stream (in presentation format)

output: the file stream to print to

rr: the resource record to print

Returns void

AUTHOR

The ldns team at NLnet Labs.

REPORTING BUGS

Please report bugs to `dns-team@nlnetlabs.nl` or on GitHub at <https://github.com/NLnetLabs/ldns/issues>

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, *ldns_rr_list*. And **perldoc Net::DNS, RFC1034, RFC1035, RFC4033, RFC4034** and **RFC4035**.

REMARKS

This manpage was automatically generated from the ldns source code.