

**NAME**

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\_rdf creation, destruction and printing

**SYNOPSIS**

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

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

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

```
ldns_rdf* ldns_rdf_new(ldns_rdf_type type, size_t size, void *data);
```

```
ldns_rdf* ldns_rdf_clone(const ldns_rdf *rd);
```

```
ldns_rdf* ldns_rdf_new_frm_data(ldns_rdf_type type, size_t size, const void *data);
```

```
ldns_rdf* ldns_rdf_new_frm_str(ldns_rdf_type type, const char *str);
```

```
ldns_status ldns_rdf_new_frm_fp(ldns_rdf **r, ldns_rdf_type type, FILE *fp);
```

```
void ldns_rdf_free(ldns_rdf *rd);
```

```
void ldns_rdf_deep_free(ldns_rdf *rd);
```

```
void ldns_rdf_print(FILE *output, const ldns_rdf *rdf);
```

**DESCRIPTION**

*ldns\_rdf\_new()* allocates a new rdf structure and fills it. This function DOES NOT copy the contents from the buffer, unlike *ldns\_rdf\_new\_frm\_data()*

**type:** type of the rdf

**size:** size of the buffer

**data:** pointer to the buffer to be copied

Returns the new rdf structure or NULL on failure

*ldns\_rdf\_clone()* clones a rdf structure. The data is copied.

**rd:** rdf to be copied

Returns a new rdf structure

*ldns\_rdf\_new\_frm\_data()* allocates a new rdf structure and fills it. This function does copy the

contents from the buffer, unlike `ldns_rdf_new()`

**type:** type of the rdf

**size:** size of the buffer

**data:** pointer to the buffer to be copied

Returns the new rdf structure or NULL on failure

*ldns\_rdf\_new\_frm\_str()* creates a new rdf from a string.

**type:** type to use

**str:** string to use

Returns `ldns_rdf*` or NULL in case of an error

*ldns\_rdf\_new\_frm\_fp()* creates a new rdf from a file containing a string.

**r:** the new rdf

**type:** type to use

**fp:** the file pointer to use

Returns `LDNS_STATUS_OK` or the error

*ldns\_rdf\_free()* frees a rdf structure, leaving the data pointer intact.

**rd:** the pointer to be freed

Returns void

*ldns\_rdf\_deep\_free()* frees a rdf structure *and* frees the data. rdf should be created with

`_new_frm_data`

**rd:** the rdf structure to be freed

Returns void

*ldns\_rdf\_print()* Prints the data in the rdata field to the given file stream (in presentation format)

**output:** the file stream to print to

**rd:** the rdata field to print

Returns void

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

**REMARKS**

This manpage was automatically generated from the ldns source code.