

NAME

ldns_dnssec_name_new, ldns_dnssec_name_new_frm_rr, ldns_dnssec_name_free,
ldns_dnssec_name_name, ldns_dnssec_name_set_name, ldns_dnssec_name_set_nsec,
ldns_dnssec_name_cmp, ldns_dnssec_name_add_rr, ldns_dnssec_name_find_rrset,
ldns_dnssec_name_print - functions for ldns_dnssec_name

SYNOPSIS

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

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

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

```
ldns_dnssec_name* ldns_dnssec_name_new(void);
```

```
ldns_dnssec_name* ldns_dnssec_name_new_frm_rr(ldns_rr *rr);
```

```
void ldns_dnssec_name_free(ldns_dnssec_name *name);
```

```
ldns_rdf* ldns_dnssec_name_name(const ldns_dnssec_name *name);
```

```
void ldns_dnssec_name_set_name(ldns_dnssec_name *name, ldns_rdf *dname);
```

```
void ldns_dnssec_name_set_nsec(ldns_dnssec_name *name, ldns_rr *nsec);
```

```
int ldns_dnssec_name_cmp(const void *a, const void *b);
```

```
ldns_status ldns_dnssec_name_add_rr(ldns_dnssec_name *name, ldns_rr *rr);
```

```
ldns_dnssec_rrsets* ldns_dnssec_name_find_rrset(const ldns_dnssec_name *name, ldns_rr_type type);
```

```
void ldns_dnssec_name_print(FILE *out, const ldns_dnssec_name *name);
```

DESCRIPTION

ldns_dnssec_name_new() Create a new data structure for a dnssec name
Returns the allocated structure

ldns_dnssec_name_new_frm_rr() Create a new data structure for a dnssec name for the given RR

rr: the RR to derive properties from, and to add to the name

ldns_dnssec_name_free() Frees the name structure and its rrs and rrsets. Individual *ldns_rr* records therein are not freed

name: the structure to free

ldns_dnssec_name_name() Returns the domain name of the given *dnssec_name* structure

name: the dnssec name to get the domain name from
Returns the domain name

ldns_dnssec_name_set_name() Sets the domain name of the given *dnssec_name* structure

name: the dnssec name to set the domain name of
dname: the domain name to set it to. This data is **not** copied.

ldns_dnssec_name_set_nsec() Sets the NSEC(3) RR of the given *dnssec_name* structure

name: the dnssec name to set the domain name of
nsec: the nsec rr to set it to. This data is **not** copied.

ldns_dnssec_name_cmp() Compares the domain names of the two arguments in their canonical ordering.

a: The first *dnssec_name* to compare
b: The second *dnssec_name* to compare
Returns -1 if the domain name of a comes before that of b in canonical ordering, 1 if it is the other way around, and 0 if they are equal

ldns_dnssec_name_add_rr() Inserts the given rr at the right place in the current *dnssec_name* No checking is done whether the name matches

name: The *ldns_dnssec_name* to add the RR to
rr: The RR to add
Returns *LDNS_STATUS_OK* on success, error code otherwise

ldns_dnssec_name_find_rrset() Find the RRset with the given type in within this name structure

name: the name to find the RRset in
type: the type of the RRset to find
Returns the RRset, or NULL if not present

ldns_dnssec_name_print() Prints the RRs in the `dnssec` name structure to the given file descriptor

out: the file descriptor to print to

name: the name structure to print the contents of

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

REMARKS

This manpage was automatically generated from the ldns source code.