

**NAME**

ldns\_dnssec\_zone, ldns\_dnssec\_name, ldns\_dnssec\_rrs, ldns\_dnssec\_rrsets - data structures

**SYNOPSIS**

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

#include <ldns/ldns.h>

ldns_dnssec_zone();
```

**DESCRIPTION**

*ldns\_dnssec\_zone()*

*ldns\_dnssec\_name*

Structure containing all resource records for a domain name

Including the derived NSEC3, if present

```
struct ldns_struct_dnssec_name
```

```
{
```

```
    /**
```

```
    * pointer to a dname containing the name.
```

```
    * Usually points to the owner name of the first RR of the first RRset
```

```
    */
```

```
    ldns_rdf *name;
```

```
    /**
```

```
    * Usually, the name is a pointer to the owner name of the first rr for
```

```
    * this name, but sometimes there is no actual data to point to,
```

```
    * for instance in
```

```
    * names representing empty nonterminals. If so, set name_allocated to true to
```

```
    * indicate that this data must also be freed when the name is freed
```

```
    */
```

```
    bool name_allocated;
```

```
    /**
```

```
    * The rrsets for this name
```

```
    */
```

```
    ldns_dnssec_rrsets *rrsets;
```

```
    /**
```

```
    * NSEC pointing to the next name (or NSEC3 pointing to the next NSEC3)
```

```
    */
```

```

ldns_rr *nsec;
/**
 * signatures for the NSEC record
 */
ldns_dnssec_rrs *nsec_signatures;
/**
 * Unlike what the name is_glue suggests, this field is set to true by
 * ldns_dnssec_zone_mark_glue() or ldns_dnssec_zone_mark_and_get_glue()
 * when the name, this dnssec_name struct represents, is occluded.
 * Names that contain other occluded rrsets and records with glue on
 * the delegation point will NOT have this bool set to true.
 * This field should NOT be read directly, but only via the
 * ldns_dnssec_name_is_glue() function!
 */
bool is_glue;
/**
 * pointer to store the hashed name (only used when in an NSEC3 zone
 */
ldns_rdf *hashed_name;
};

typedef struct ldns_struct_dnssec_name ldns_dnssec_name;

```

*ldns\_dnssec\_rrs*

Singly linked list of rrs

```

struct ldns_struct_dnssec_rrs
{
    ldns_rr *rr;
    ldns_dnssec_rrs *next;
};

```

```

typedef struct ldns_struct_dnssec_rrs ldns_dnssec_rrs;

```

*ldns\_dnssec\_rrsets*

Singly linked list of RRsets

```

struct ldns_struct_dnssec_rrsets
{
    ldns_dnssec_rrs *rrs;
    ldns_rr_type type;
    ldns_dnssec_rrs *signatures;
};

```

```
        ldns_dnssec_rrsets *next;
    };

typedef struct ldns_struct_dnssec_rrsets ldns_dnssec_rrsets;
```

## AUTHOR

The ldns team at NLnet Labs.

## REPORTING BUGS

Please report bugs to [dns-team@nlnetlabs.nl](mailto: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\_dnssec\_zone\_new*, *ldns\_dnssec\_name\_new*, *ldns\_dnssec\_rrs\_new*, *ldns\_dnssec\_rrsets\_new*. And **perldoc Net::DNS**, **RFC1034**, **RFC1035**, **RFC4033**, **RFC4034** and **RFC4035**.

## REMARKS

This manpage was automatically generated from the ldns source code.