

**NAME**

ldns\_pkt\_edns, ldns\_pkt\_edns\_udp\_size, ldns\_pkt\_edns\_extended\_rcode, ldns\_pkt\_edns\_version, ldns\_pkt\_edns\_z, ldns\_pkt\_edns\_data, ldns\_pkt\_set\_edns\_udp\_size, ldns\_pkt\_set\_edns\_extended\_rcode, ldns\_pkt\_set\_edns\_version, ldns\_pkt\_set\_edns\_z, ldns\_pkt\_set\_edns\_data - ldns\_pkt ends0 related functions

**SYNOPSIS**

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

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

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

```
bool ldns_pkt_edns(const ldns_pkt *packet);
```

```
uint16_t ldns_pkt_edns_udp_size(const ldns_pkt *packet);
```

```
uint8_t ldns_pkt_edns_extended_rcode(const ldns_pkt *packet);
```

```
uint8_t ldns_pkt_edns_version(const ldns_pkt *packet);
```

```
uint16_t ldns_pkt_edns_z(const ldns_pkt *packet);
```

```
ldns_rdf* ldns_pkt_edns_data(const ldns_pkt *packet);
```

```
void ldns_pkt_set_edns_udp_size(ldns_pkt *packet, uint16_t s);
```

```
void ldns_pkt_set_edns_extended_rcode(ldns_pkt *packet, uint8_t c);
```

```
void ldns_pkt_set_edns_version(ldns_pkt *packet, uint8_t v);
```

```
void ldns_pkt_set_edns_z(ldns_pkt *packet, uint16_t z);
```

```
void ldns_pkt_set_edns_data(ldns_pkt *packet, ldns_rdf *data);
```

**DESCRIPTION**

*ldns\_pkt\_edns()* returns true if this packet needs and EDNS rr to be sent. At the moment the only reason is an expected packet size larger than 512 bytes, but for instance dnssec would be a good reason too.

**packet:** the packet to check  
Returns true if packet needs edns rr

*ldns\_pkt\_edns\_udp\_size()* return the packet's edns udp size

**packet:** the packet  
Returns the size

*ldns\_pkt\_edns\_extended\_rcode()* return the packet's edns extended rcode

**packet:** the packet  
Returns the rcode

*ldns\_pkt\_edns\_version()* return the packet's edns version

**packet:** the packet  
Returns the version

*ldns\_pkt\_edns\_z()* return the packet's edns z value

**packet:** the packet  
Returns the z value

*ldns\_pkt\_edns\_data()* return the packet's EDNS data

**packet:** the packet  
Returns the data

*ldns\_pkt\_set\_edns\_udp\_size()* Set the packet's edns udp size

**packet:** the packet  
**s:** the size

*ldns\_pkt\_set\_edns\_extended\_rcode()* Set the packet's edns extended rcode

**packet:** the packet  
**c:** the code

*ldns\_pkt\_set\_edns\_version()* Set the packet's edns version

**packet:** the packet  
**v:** the version

*ldns\_pkt\_set\_edns\_z()* Set the packet's edns z value

**packet:** the packet  
**z:** the value

*ldns\_pkt\_set\_edns\_data()* Set the packet's EDNS data

**packet:** the packet

**data:** the data

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

## REMARKS

This manpage was automatically generated from the ldns source code.