NAME

ldns_pkt_id, ldns_pkt_qr, ldns_pkt_aa, ldns_pkt_tc, ldns_pkt_rd, ldns_pkt_cd, ldns_pkt_ra, ldns_pkt_ad, ldns_pkt_get_opcode, ldns_pkt_get_rcode, ldns_pkt_qdcount, ldns_pkt_ancount, ldns_pkt_nscount, ldns_pkt_arcount, ldns_pkt_answerfrom, ldns_pkt_querytime, ldns_pkt_size, ldns_pkt_tsig, ldns_pkt_question, ldns_pkt_answer, ldns_pkt_authority, ldns_pkt_additional, ldns_pkt_get_section_clone, ldns_pkt_rr_list_by_name, ldns_pkt_rr_list_by_type, ldns_pkt_rr_list_by_name_and_type **-** get ldns_pkt attributes

SYNOPSIS

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

#include <ldns/ldns.h>

 ldns_pkt_id();

bool ldns_pkt_qr(const ldns_pkt *p);

bool ldns_pkt_aa(const ldns_pkt *p);

bool ldns_pkt_tc(const ldns_pkt *p);

bool ldns_pkt_rd(const ldns_pkt *p);

bool ldns_pkt_cd(const ldns_pkt *p);

bool ldns_pkt_ra(const ldns_pkt *p);

bool ldns_pkt_ad(const ldns_pkt *p);

ldns_pkt_opcode ldns_pkt_get_opcode(const ldns_pkt *p);

ldns_pkt_rcode ldns_pkt_get_rcode(const ldns_pkt *p);

uint16_t ldns_pkt_qdcount(const ldns_pkt *p);

uint16_t ldns_pkt_ancount(const ldns_pkt *p);

uint16_t ldns_pkt_nscount(const ldns_pkt *p);
```

uint16_t ldns_pkt_arcount(const ldns_pkt *p);

ldns_rdf* ldns_pkt_answerfrom(const ldns_pkt *p);

uint32_t ldns_pkt_querytime(const ldns_pkt *p);

size_t ldns_pkt_size(const ldns_pkt *p);

ldns_rr* ldns_pkt_tsig(const ldns_pkt *p);

ldns_rr_list* ldns_pkt_question(const ldns_pkt *p);

ldns_rr_list* ldns_pkt_answer(const ldns_pkt *p);

ldns_rr_list* ldns_pkt_authority(const ldns_pkt *p);

ldns_rr_list* ldns_pkt_additional(const ldns_pkt *p);

ldns_rr_list* ldns_pkt_get_section_clone(const ldns_pkt *p, ldns_pkt_section s);

ldns_rr_list* ldns_pkt_rr_list_by_name(const ldns_pkt *p, const ldns_rdf *r, ldns_pkt_section s);

ldns_rr_list* ldns_pkt_rr_list_by_type(const ldns_pkt *p, ldns_rr_type t, ldns_pkt_section s);

ldns_rr_list* ldns_pkt_rr_list_by_name_and_type(const ldns_pkt *packet, const ldns_rdf *ownername, ldns_rr_type type, ldns_pkt_section sec);

DESCRIPTION

ldns_pkt_id()

ldns_pkt_qr() Read the packet's qr bit
p: the packet
Returns value of the bit

ldns_pkt_aa() Read the packet's aa bit
p: the packet
Returns value of the bit

ldns_pkt_tc() Read the packet's tc bit
p: the packet

Returns value of the bit

ldns_pkt_rd() Read the packet's rd bit
p: the packet
Returns value of the bit

ldns_pkt_cd() Read the packet's cd bit
p: the packet
Returns value of the bit

ldns_pkt_ra() Read the packet's ra bit
p: the packet
Returns value of the bit

ldns_pkt_ad() Read the packet's ad bit
p: the packet
Returns value of the bit

ldns_pkt_get_opcode() Read the packet's code
p: the packet
Returns the opcode

ldns_pkt_get_rcode() Return the packet's response code
p: the packet
Returns the response code

ldns_pkt_qdcount() Return the packet's qd count
p: the packet
Returns the qd count

ldns_pkt_ancount() Return the packet's an count
p: the packet
Returns the an count

ldns_pkt_nscount() Return the packet's ns count
p: the packet
Returns the ns count

ldns_pkt_arcount() Return the packet's ar count
p: the packet

Returns the ar count

ldns_pkt_answerfrom() Return the packet's answerfrom
p: packet
Returns the name of the server

ldns_pkt_querytime() Return the packet's querytime
p: the packet
Returns the querytime

ldns_pkt_size() Return the packet's size in bytes
p: the packet
Returns the size

ldns_pkt_tsig() Return the packet's tsig pseudo rr's
p: the packet
Returns the tsig rr

ldns_pkt_question() Return the packet's question section
p: the packet
Returns the section

ldns_pkt_answer() Return the packet's answer section
p: the packet
Returns the section

ldns_pkt_authority() Return the packet's authority section
p: the packet
Returns the section

ldns_pkt_additional() Return the packet's additional section
p: the packet
Returns the section

ldns_pkt_get_section_clone() return all the rr_list's in the packet. Clone the lists, instead of returning pointers.
p: the packet to look in
s: what section(s) to return
Returns ldns_rr_list with the rr's or NULL if none were found

ldns_pkt_rr_list_by_name() return all the rr with a specific name from a packet. Optionally specify from which section in the packet
p: the packet
r: the name
s: the packet's section
Returns a list with the rr's or NULL if none were found

ldns_pkt_rr_list_by_type() return all the rr with a specific type from a packet. Optionally specify from which section in the packet
p: the packet
t: the type
s: the packet's section
Returns a list with the rr's or NULL if none were found

ldns_pkt_rr_list_by_name_and_type() return all the rr with a specific type and type from a packet. Optionally specify from which section in the packet
packet: the packet
ownername: the name
type: the type
sec: the packet's section
Returns a list with the rr's or NULL if none were found

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

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.