

**NAME**

Net::DNS::Header – DNS packet header class

**SYNOPSIS**

```
use Net::DNS::Header;
```

**DESCRIPTION**

A `Net::DNS::Header` object represents the header portion of a DNS packet.

**METHODS****new**

```
$header = Net::DNS::Header->new;
```

`new` creates a header object appropriate for making a DNS query.

**parse**

```
($header, $offset) = Net::DNS::Header->parse(\$data);
```

Parses the header record at the start of a DNS packet. The argument is a reference to the packet data.

Returns a `Net::DNS::Header` object and the offset of the next location in the packet.

Parsing is aborted if the header object cannot be created (e.g., corrupt or insufficient data).

**print**

```
$header->print;
```

Prints the header record on the standard output.

**string**

```
print $header->string;
```

Returns a string representation of the header object.

**id**

```
print "query id = ", $header->id, "\n";
$header->id(1234);
```

Gets or sets the query identification number.

**qr**

```
print "query response flag = ", $header->qr, "\n";
$header->qr(0);
```

Gets or sets the query response flag.

**opcode**

```
print "query opcode = ", $header->opcode, "\n";
$header->opcode("UPDATE");
```

Gets or sets the query opcode (the purpose of the query).

**aa**

```
print "answer is ", $header->aa ? "" : "non-", "authoritative\n";
$header->aa(0);
```

Gets or sets the authoritative answer flag.

**tc**

```
print "packet is ", $header->tc ? "" : "not ", "truncated\n";
$header->tc(0);
```

Gets or sets the truncated packet flag.

**rd**

```
print "recursion was ", $header->rd ? "" : "not ", "desired\n";
$header->rd(0);
```

Gets or sets the recursion desired flag.

**cd**

```
print "checking was ", $header->cd ? "not" : "", "desired\n";
$header->cd(0);
```



Gets or sets the checking disabled flag.

#### **ra**

```
print "recursion is ", $header->ra ? " " : "not ", "available\n";
$header->ra(0);
```

Gets or sets the recursion available flag.

#### **ad**

```
print "The result has ", $header->ad ? " " : "not", "been verified\n"
```

Relevant in DNSSEC context.

(The AD bit is only set on answers where signatures have been cryptographically verified or the server is authoritative for the data and is allowed to set the bit by policy.)

#### **rcode**

```
print "query response code = ", $header->rcode, "\n";
$header->rcode("SERVFAIL");
```

Gets or sets the query response code (the status of the query).

#### **qdcount, zocount**

```
print "# of question records: ", $header->qdcount, "\n";
$header->qdcount(2);
```

Gets or sets the number of records in the question section of the packet. In dynamic update packets, this field is known as zocount and refers to the number of RRs in the zone section.

#### **ancount, prcount**

```
print "# of answer records: ", $header->ancount, "\n";
$header->ancount(5);
```

Gets or sets the number of records in the answer section of the packet. In dynamic update packets, this field is known as prcount and refers to the number of RRs in the prerequisite section.

#### **nscount, upcount**

```
print "# of authority records: ", $header->nscount, "\n";
$header->nscount(2);
```

Gets or sets the number of records in the authority section of the packet. In dynamic update packets, this field is known as upcount and refers to the number of RRs in the update section.

#### **arcount, adcount**

```
print "# of additional records: ", $header->arcount, "\n";
$header->arcount(3);
```

Gets or sets the number of records in the additional section of the packet. In dynamic update packets, this field is known as adcount.

#### **data**

```
$hdata = $header->data;
```

Returns the header data in binary format, appropriate for use in a DNS query packet.

## **COPYRIGHT**

Copyright (c) 1997–2002 Michael Fuhr.

Portions Copyright (c) 2002–2004 Chris Reinhardt.

Portions Copyright (c) 2007 Dick Franks.

All rights reserved. This program is free software; you may redistribute it and/or modify it under the same terms as Perl itself.

## **SEE ALSO**

*perl*(1), Net::DNS, Net::DNS::Resolver, Net::DNS::Packet, Net::DNS::Update, Net::DNS::Question, Net::DNS::RR, RFC 1035 Section 4.1.1

