NAME

Net::DNS::DomainName – DNS name representation

SYNOPSIS

```
use Net::DNS::DomainName;

$object = new Net::DNS::DomainName('example.com');
$name = $object->name;
$data = $object->encode;

( $object, $next ) = decode Net::DNS::DomainName( \$data, $offset );
```

DESCRIPTION

The Net::DNS::DomainName module implements the concrete representation of DNS domain names used within DNS packets.

Net::DNS::DomainName defines methods for encoding and decoding wire format octet strings as defined in RFC1035. All other behaviour, including the **new()** constructor, is inherited from Net::DNS::Domain.

The Net::DNS::DomainName1035 and Net::DNS::DomainName2535 packages implement disjoint domain name subtypes which provide the name compression and canonicalisation specified by RFC1035 and RFC2535. These are necessary to meet the backward compatibility requirements introduced by RFC3597.

METHODS

new

```
$object = new Net::DNS::DomainName('example.com');
```

Creates a domain name object which identifies the domain specified by the character string argument.

canonical

```
$data = $object->canonical;
```

Returns the canonical wire-format representation of the domain name as defined in RFC2535(8.1).

decode

```
$object = decode Net::DNS::DomainName( \$buffer, $offset, $hash );

( $object, $next ) = decode Net::DNS::DomainName( \$buffer, $offset, $hash );
```

Creates a domain name object which represents the DNS domain name identified by the wire-format data at the indicated offset within the data buffer.

The argument list consists of a reference to a scalar containing the wire-format data and specified offset. The optional reference to a hash table provides improved efficiency of decoding compressed names by exploiting already cached compression pointers.

The returned offset value indicates the start of the next item in the data buffer.

encode

```
$data = $object->encode;
```

Returns the wire-format representation of the domain name suitable for inclusion in a DNS packet buffer.

Net::DNS::DomainName1035

Net::DNS::DomainName1035 implements a subclass of domain name objects which are to be encoded using the compressed wire format defined in RFC1035.

```
use Net::DNS::DomainName;

$object = new Net::DNS::DomainName1035('compressible.example.com');
$data   = $object->encode( $offset, $hash );

( $object, $next ) = decode Net::DNS::DomainName1035( \$data, $offset );
```

Note that RFC3597 implies that the RR types defined in RFC1035 section 3.3 are the only types eligible for

compression.

encode

```
$data = $object->encode( $offset, $hash );
```

Returns the wire-format representation of the domain name suitable for inclusion in a DNS packet buffer.

The optional arguments are the offset within the packet data where the domain name is to be stored and a reference to a hash table used to index compressed names within the packet.

If the hash reference is undefined, **encode()** returns the lowercase uncompressed canonical representation defined in RFC2535(8.1).

Net::DNS::DomainName2535

Net::DNS::DomainName2535 implements a subclass of domain name objects which are to be encoded using uncompressed wire format.

Note that RFC3597, and latterly RFC4034, specifies that the lower case canonical encoding defined in RFC2535 is to be used for RR types defined prior to RFC3597.

```
use Net::DNS::DomainName;

$object = new Net::DNS::DomainName2535('incompressible.example.com');
$data   = $object->encode( $offset, $hash );

( $object, $next ) = decode Net::DNS::DomainName2535( \$data, $offset );
```

encode

```
$data = $object->encode( $offset, $hash );
```

Returns the uncompressed wire-format representation of the domain name suitable for inclusion in a DNS packet buffer.

If the hash reference is undefined, **encode()** returns the lowercase canonical form defined in RFC2535(8.1).

COPYRIGHT

LICENSE

SEE ALSO

perl, Net::DNS, Net::DNS::Domain, RFC1035, RFC2535, RFC3597, RFC4034