NAME

Net::DNS::Domain – DNS domains

SYNOPSIS

```
use Net::DNS::Domain;

$domain = new Net::DNS::Domain('example.com');
$name   = $domain->name;
```

DESCRIPTION

The Net::DNS::Domain module implements a class of abstract DNS domain objects with associated class and instance methods.

Each domain object instance represents a single DNS domain which has a fixed identity throughout its lifetime.

Internally, the primary representation is a (possibly empty) list of ASCII domain name labels, and optional link to an arbitrary origin domain object topologically closer to the DNS root.

The computational expense of Unicode character-set conversion is partially mitigated by use of caches.

METHODS

new

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

Creates a domain object which represents the DNS domain specified by the character string argument. The argument consists of a sequence of labels delimited by dots.

A character preceded by \ represents itself, without any special interpretation.

Arbitrary 8−bit codes can be represented by \ followed by exactly three decimal digits. Character code points are ASCII, irrespective of the character coding scheme employed by the underlying platform.

Argument string literals should be delimited by single quotes to avoid escape sequences being interpreted as octal character codes by the Perl compiler.

The character string presentation format follows the conventions for zone files described in RFC1035.

Users should be aware that non-ASCII domain names will be transcoded to NFC before encoding, which is an irreversible process.

name

```
$name = $domain->name;
```

Returns the domain name as a character string corresponding to the ‘‘common interpretation’’ to which RFC1034, 3.1, paragraph 9 alludes.

Character escape sequences are used to represent a dot inside a domain name label and the escape character itself.

Any non-printable code point is represented using the appropriate numerical escape sequence.

fqdn

```
@fqdn = $domain->fqdn;
```

Returns a character string containing the fully qualified domain name, including the trailing dot.

xname

```
$xname = $domain->xname;
```

Interprets an extended name containing Unicode domain name labels encoded as Punycode A−labels.

If decoding is not possible, the ACE encoded name is returned.

label

```
@label = $domain->label;
```

Identifies the domain by means of a list of domain labels.

string

```
$string = $object->string;
```

Returns a character string containing the fully qualified domain name as it appears in a zone file.

Characters which are recognised by RFC1035 zone file syntax are represented by the appropriate escape sequence.

origin

```
$create = origin Net::DNS::Domain( $ORIGIN );
$result = &$create( sub{ new Net::DNS::RR( 'mx MX 10 a' ); } );
$expect = new Net::DNS::RR( "mx.$ORIGIN. MX 10 a.$ORIGIN." );
```

Class method which returns a reference to a subroutine wrapper which executes a given constructor in a dynamically scoped context where relative names become descendents of the specified `$ORIGIN`.

BUGS

Coding strategy is intended to avoid creating unnecessary argument lists and stack frames. This improves efficiency at the expense of code readability.

Platform specific character coding features are conditionally compiled into the code.

COPYRIGHT

LICENSE

SEE ALSO

perl, Net::DNS, Net::LibIDN2, RFC1034, RFC1035, RFC5891, Unicode TR#16