NAME

Net::DNS::ZoneFile – DNS zone file

SYNOPSIS

```
use Net::DNS::ZoneFile;

$zonefile = new Net::DNS::ZoneFile( 'named.example' );

while ( $rr = $zonefile->read ) {
    $rr->print;
}

@zone = $zonefile->read;
```

DESCRIPTION

Each Net::DNS::ZoneFile object instance represents a zone file together with any subordinate files introduced by the `$INCLUDE` directive. Zone file syntax is defined by RFC1035.

A program may have multiple zone file objects, each maintaining its own independent parser state information.

The parser supports both the `$TTL` directive defined by RFC2308 and the BIND `$GENERATE` syntax extension.

All RRs in a zone file must have the same class, which may be specified for the first RR encountered and is then propagated automatically to all subsequent records.

METHODS

new

```
$zonefile = new Net::DNS::ZoneFile( 'filename', ['example.com'] );

$handle   = new IO::File( 'filename', '<:encoding(ISO8859-7)' );
$zonefile = new Net::DNS::ZoneFile( $handle, ['example.com'] );
```

The **new()** constructor returns a Net::DNS::ZoneFile object which represents the zone file specified in the argument list.

The specified file or file handle is open for reading and closed when exhausted or all references to the ZoneFile object cease to exist.

The optional second argument specifies `$ORIGIN` for the zone file.

Character encoding is specified indirectly by creating a file handle with the desired encoding layer, which is then passed as an argument to **new()**. The specified encoding is propagated to files introduced by `$include` directives.

read

```
$rr = $zonefile->read;
@rr = $zonefile->read;
```

When invoked in scalar context, **read()** returns a Net::DNS::RR object representing the next resource record encountered in the zone file, or undefined if end of data has been reached.

When invoked in list context, **read()** returns the list of Net::DNS::RR objects in the order that they appear in the zone file.

Comments and blank lines are silently disregarded.

`$INCLUDE`, `$ORIGIN`, `$TTL` and `$GENERATE` directives are processed transparently.

name

```
$filename = $zonefile->name;
```

Returns the name of the current zone file. Embedded `$INCLUDE` directives will cause this to differ from the filename argument supplied when the object was created.

line

```
$line = $zonefile->line;
```

Returns the number of the last line read from the current zone file.

origin

```
$origin = $zonefile->origin;
```

Returns the fully qualified name of the current origin within the zone file.

ttl

```
$ttl = $zonefile->ttl;
```

Returns the default TTL as specified by the `$TTL` directive.

COMPATIBILITY WITH Net::DNS::ZoneFile 1.04

Applications which depended on the defunct Net::DNS::ZoneFile 1.04 CPAN distribution will continue to operate with minimal change using the compatibility interface described below. New application code should use the object-oriented interface.

```
use Net::DNS::ZoneFile;

$listref = Net::DNS::ZoneFile->read( $filename );
$listref = Net::DNS::ZoneFile->read( $filename, $include_dir );

$listref = Net::DNS::ZoneFile->readfh( $filehandle );
$listref = Net::DNS::ZoneFile->readfh( $filehandle, $include_dir );

$listref = Net::DNS::ZoneFile->parse(  $string );
$listref = Net::DNS::ZoneFile->parse( \$string );
$listref = Net::DNS::ZoneFile->parse(  $string, $include_dir );
$listref = Net::DNS::ZoneFile->parse( \$string, $include_dir );

$_->print for @$listref;
```

The optional second argument specifies the default path for filenames. The current working directory is used by default.

Although not available in the original implementation, the RR list can be obtained directly by calling any of these methods in list context.

```
@rr = Net::DNS::ZoneFile->read( $filename, $include_dir );
```

The partial result is returned if an error is encountered by the parser.

read

```
$listref = Net::DNS::ZoneFile->read( $filename );
$listref = Net::DNS::ZoneFile->read( $filename, $include_dir );
```

read() parses the contents of the specified file and returns a reference to the list of Net::DNS::RR objects. The return value is undefined if an error is encountered by the parser.

readfh

```
$listref = Net::DNS::ZoneFile->readfh( $filehandle );
$listref = Net::DNS::ZoneFile->readfh( $filehandle, $include_dir );
```

readfh() parses data from the specified file handle and returns a reference to the list of Net::DNS::RR objects. The return value is undefined if an error is encountered by the parser.

parse

```
$listref = Net::DNS::ZoneFile->parse(  $string );
$listref = Net::DNS::ZoneFile->parse( \$string );
$listref = Net::DNS::ZoneFile->parse(  $string, $include_dir );
$listref = Net::DNS::ZoneFile->parse( \$string, $include_dir );
```

parse() interprets the text in the argument string and returns a reference to the list of Net::DNS::RR objects. The return value is undefined if an error is encountered by the parser.

ACKNOWLEDGEMENTS

This package is designed as an improved and compatible replacement for Net::DNS::ZoneFile 1.04 which was created by Luis Munoz in 2002 as a separate CPAN module.

The present implementation is the result of an agreement to merge our two different approaches into one package integrated into Net::DNS. The contribution of Luis Munoz is gratefully acknowledged.

Thanks are also due to Willem Toorop for his constructive criticism of the initial version and invaluable assistance during testing.

COPYRIGHT

LICENSE

SEE ALSO

perl, Net::DNS, Net::DNS::RR, RFC1035 Section 5.1, RFC2308, BIND 9 Administrator Reference Manual