DNS MIB Extensions

12–November–1992

DNS Working Group

Rob Austein
Epilogue Technology Corporation
sra@epilogue.com

Jon Saperia
Digital Equipment Corporation
saperia@tcpjon.ogo.dec.com

Status of this Memo

This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF), its Areas, and its Working Groups. Note that other groups may also distribute working documents as Internet Drafts. Internet Drafts are draft documents valid for a maximum of six months. Internet Drafts may be updated, replaced, or obsoleted by other documents at any time. It is not appropriate to use Internet Drafts as reference material or to cite them other than as a "working draft" or "work in progress." Please check the I-D abstract listing contained in each Internet Draft directory to learn the current status of this or any other Internet Draft.

This draft document will be submitted to the RFC editor as a proposed extension to the SNMP MIB. Distribution of this document is unlimited. Please send comments or corrections to the authors.

Abstract

This memo defines a set of DNS (Domain Name System) extensions that have been created for the Internet MIB. When used in conjunction with the Structure of Management Information (RFC 1155), the Management Information Base for Network Management of TCP/IP-based internets (RFC 1213) and the Simple Network Management Protocol (RFC 1157), it will be possible to provide integrated network management of DNS resolver and server software in standard TCP /IP based environments. This document was produced by the DNS working group.

Digital Equipment Corporation
Maynard, Massachusetts

CONTENTS

1 Introduction 1

2 The DNS Model 1

3 Selected Objects 2

4 Objects 3

4.1 Format of Definitions 3

4.2 Textual Conventions 3

5 Object Definitions 3

6 Acknowledgements 37

7 References 37

1 Introduction

With the adoption of The Simple Network Management Protocol (RFC 1157), the Management Information Base for network management of TCP/IP-based internets (RFC 1213), and the Structure of Management Information (RFC 1155) by the Internet, and a large number of vendor implementations of these standards in commercially available products, it became possible to provide a higher level of effective network management in TCP/IP-based internets than previously available. With the growth in the use of these standards, it has become possible to consider the management of other elements of the infrastructure beyond the basic TCP/IP protocols. A key element of the TCP/IP infrastructure is the DNS.

Up to this point there has been no mechanism to integrate the management of the DNS with SNMP-based managers. This memo provides the mechanisms by which IP-based management stations can effectively manage DNS client and server software in an integrated fashion through the use of the standard Internet SMI, MIB and Simple Network Management Protocol.

New DNS MIB objects have been defined to be used in conjunction with the Internet MIB to allow access and control of the DNS via SNMP by the Internet community.

2 The DNS Model

In theory, the DNS world is pretty simple. There are two kinds of entities: resolvers and name servers. Resolvers ask questions. Name servers answer them. The real world, however, is not so simple. Implementors have made widely differing choices about how to divide DNS functions between resolvers and servers. They have also constructed various sorts of exotic hybrids. The most difficult task in defining this MIB was to accommodate this wide range of entities without having to come up with a separate MIB for each.

We divided up the various DNS functions into two, non-overlapping classes, called 'resolver functions' and 'name server functions'. A DNS entity that performs what we define as resolver functions contains a resolver, and therefore must implement the MIB groups required of all resolvers. Some resolvers also implement 'optional' functions such as a cache. In this example, they will also implement the cache group contained in this MIB. A DNS entity which implements name server functions is considered to be a name server, and must implement the MIB groups required for name servers. If the same piece of software performs both resolver and server functions, we imagine that it contains both a resolver and a server.

In our model, a resolver is a program (or piece thereof) which obtains resource records from servers. Normally it does so at the behest of an application, but may also do so as part of its own operation. A resolver sends DNS protocol queries and receives DNS protocol replies. A resolver neither receives queries nor sends replies. A full service resolver is one that knows how to resolve queries: it obtains the needed resource records by contacting a server authoritative for the records desired. A stub resolver does not know how to resolve queries: it sends all queries to a local name server, setting the recursion desired flag to indicate that it hopes that the name server will be willing resolve the query. A resolver may (optionally) have a cache for remembering previously acquired resource records. It may also have a negative cache for remembering names or data that have been determined not to exist.

A name server is a program (or piece thereof) that provides resource records to resolvers. All references in this document to 'a name server' imply 'the name server's role'. (In some cases the name server's role and the resolver's role might be combined into a single program.) A name server receives DNS protocol queries and sends DNS protocol replies. A name server neither sends queries nor receives replies. As a con- sequence, name servers do not have caches. Normally, a name server would expect to receive only those queries for which it could respond to with authoritative information. However, if a name server receives a query that it cannot respond to with purely authoritative information, it may choose to try to obtain the necessary additional information from a resolver which may or may not be a separate process.

3 Selected Objects

Many of the objects included in this memo have been created from information contained in the DNS specification. The DNS specification is found in Domain Names - Concepts and Facilities (RFC 1034) and Domain Names - Implementation and Specification (RFC 1035), as amended and clarified by Requirements for Internet Hosts - Application and Support (RFC1123). Additional usage information is found in the Domain Administrators Guide (RFC 1032), and the Domain Administrators Operations Guide (RFC 1033). Other objects have been created based on experience with existing DNS management tools, expected operational need, and the statistics generated by existing DNS implementations. These objects have been ordered into groups as follows:

- General Configuration Group
- Resolver Configuration Group
- Server Configuration Group
- Resolver Counter Group
- Server Counter Group
- Records Group
- Resolver Cache Group
- Resolver Negative Cache Group
- Resolver Statistics Group
- Server Management Group

The ordering of objects into these groups reflects the DNS model in which the resolver and server functions can be separate pieces of code which may or may not reside on the same host. This approach accommodates common implementations such as BIND, but it is not constrained by that or any other implementation paradigm.

Some of the objects defined in this memo have been created from information contained in existing configuration files used by many DNS implementations. This information has been converted into a standard form using the Internet Standard SMI defined in RFC 1155.

4 Objects

The objects in this memo are described using the standard Internet SMI and BER of RFC 1155. Each object description includes the objects name, its syntax and encoding. Just as with objects supported in the MIB (RFC 1156), an object name is identified with an object identifier which has been administratively assigned. This identifies an Object Type. When an object type is combined with a specific instance - the particular object is uniquely identified. Use of Object Descriptors in this memo is consistent with that of RFC 1156 - meaning that they are text strings meant to be read by humans. The descriptors have been created from a variety of sources. For the most part, the descriptions are influenced by by the DNS related RFCs noted above. For example, the descriptors for counters used for the various types of queries of DNS records are influenced by the definitions used for the various record types found in Domain Names - Implementation and Specification RFC 1035.

4.1 Format of Definitions

An object in this memo is specified by five fields of information: Object, Syntax, Description, Access, and Status.

The OBJECT is a textual name (OBJECT DESCRIPTOR) for that object type combined with an administratively obtained OBJECT IDENTIFIER.

SYNTAX : For each object type, its abstract syntax is presented using the ASN.1 specified in RFC 1155.

DESCRIPTION: A general description of the object type.

ACCESS : The standard access keywords supported in RFC 1156 are used. The keywords used in this MIB are: read-only, read-write, and not-accessible.

STATUS : The status field is used to describe with a single keyword whether the object type is mandatory or optional. Status keywords of obsolete and deprecated are not used in this memo since this is the first version of the DNS MIB.

4.2 Textual Conventions

Several datatypes have been introduced as a textual conventions in this DNS MIB document. These additions will facilitate the common understanding of information used by the DNS. No changes to the SMI or the SNMP are necessary to support these conventions which are described in 5.0 (Object Definitions).

5 Object Definitions

```
RFCxxxx-dnsMIB

DEFINITIONS ::= BEGIN
```

```
     IMPORTS
         mgmt, NetworkAddress, IpAddress,
         Counter, Gauge, TimeTicks
           FROM RFC1155-SMI
         DisplayString, mib-2
           FROM RFC1213-MIB
         OBJECT-TYPE
           FROM RFC-1212;

-- DNS MIB

     dns            OBJECT IDENTIFIER ::= { experimental 9999 }

-- textual conventions

DnsDate ::= OCTET STRING (SIZE (8 | 11))
-- This data type is intended to provide a consistent
-- method of reporting date information.  The information
-- is organized as follows:  the first two octets represent
-- the year, the next two are for the month and day of the
-- year.  The next three octets are for hours, minutes and
-- seconds.  The next octet is for deci-seconds.  Direction
-- from UT is in the next octet.  The next two octets are for
-- hours and minutes from UT.  Note that in systems which do not
-- track UT, they will return only the first 8 octets.  The table
-- below is intended to help to make clear this convention.
--
--     field  octets  contents                  range
--       1      1-2   year                      0..65536
--       2       3    month                     1..12
--       3       4    day                       1..31
--       4       5    hour                      0..23
--       5       6    minutes                   0..59
--       6       7    seconds                   0..60
--                    (use 60 for leap-second)
--       7       8    deci-seconds              0..9
--       8       9    direction from UT         "+" / "-"
--       9      10    hours from UT             0..11
--      10      11    minutes from UT           0..59
-- For example, Tuesday May 26, 1992 at 1:30:15 PM EDT could
-- be displayed as on a management station:
--                 1992-5-26,13:30:15.0,-4:0

DnsName ::= OCTET STRING
-- A DNS name is a sequence of labels.  When DNS names are displayed,
-- the boundaries between labels are typically indicated by dots (e.g.
-- "Acme" and "COM" are labels in the name "Acme.COM" ).  In the DNS
-- protocol, however, no such separators are needed because each label
-- is encoded as a length octet followed by the indicated number of
-- octets of label.  For  example, "Acme.COM" is encoded as the octet
-- sequence { 4, 'A', 'c', 'm', 'e', 3, 'C', 'O', 'M', 0 } (the final
-- 0 is the length of the name of the root domain, which appears
-- implicitly at the end of any DNS name).  This MIB uses the same
-- encoding as the DNS protocol.

-- A DnsName must always be a fully qualified name.  It is an error to
-- encode a relative domain name as a DnsName without first making it a
-- fully qualified name.

DnsClass ::= INTEGER (0..65535)
-- This data type is used to represent the class values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new classes of records to be defined.
-- Existing standard classes are listed in the DNS specification.
```

```
DnsType ::= INTEGER (0..65535)
-- This data type is used to represent the type values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new record types to be defined.
-- Existing standard types are listed in the DNS specification.

DnsQClass ::= INTEGER (0..65535)
-- This data type is used to represent the QClass values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new QClass records to be defined.  Existing
-- standard Qclasses are listed in the DNS specification.

DnsQType ::= INTEGER (0..65535)
-- This data type is used to represent the QType values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new QType records to be defined.
-- Existing standard QTypes are listed in DNS specification.

DnsTime ::= INTEGER
-- DnsTime values are 32-bit unsigned integers which measure time in
-- seconds.

DnsValid ::= INTEGER { valid (1), clear (2) }
-- Many of the tables in this MIB have as one of their columns, an
-- object which can be set to a value of 2 to delete that
-- row of the table.  If a read operation is performed on this
-- object, a value of 1 is returned to indicate a valid row in the
-- table.

DnsOpCode ::= INTEGER (0..15)
-- This data type is used to represent the DNS OPCODE used in the
-- header section of DNS messages.  Existing standard OPCODE values
-- are listed in the DNS specification.

DnsRespCode ::= INTEGER (0..15)
-- This data type is used to represent the DNS RCODE value in response
-- messages.  Existing standard RCODE values are listed in the DNS
-- specification.

-- groups in the dns mib

      dnsGenConfig              OBJECT IDENTIFIER ::= { dns 1 }
      dnsResConfig              OBJECT IDENTIFIER ::= { dns 2 }
      dnsServConfig             OBJECT IDENTIFIER ::= { dns 3 }
      dnsResCounter             OBJECT IDENTIFIER ::= { dns 4 }
      dnsServCounter            OBJECT IDENTIFIER ::= { dns 5 }
      dnsRec                    OBJECT IDENTIFIER ::= { dns 6 }
      dnsResCache               OBJECT IDENTIFIER ::= { dns 7 }
      dnsResNCache              OBJECT IDENTIFIER ::= { dns 8 }
      dnsResStats               OBJECT IDENTIFIER ::= { dns 9 }
      dnsServMgmt               OBJECT IDENTIFIER ::= { dns 10 }

-- General Configuration Group

-- The implementation of the General Configuration group is
-- mandatory for all systems.
```

```
dnsGenConfigDnsUse OBJECT-TYPE
     SYNTAX  INTEGER {
          primary (1),
          not-primary (2),
          not-used (3)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "A value of 1 indicates that the DNS is used as the
             primary mechanism for name resolution on this system.
             A 2 indicates that some other mechanism is used as the
             primary mechanism for name resolution and that the DNS
             is used as a back-up.  A value of 3 indicates that DNS
             is not used for name resolution."
     ::= { dnsGenConfig 1 }

-- Resolver Configuration Group

-- The implementation of the Resolver Configuration group is
-- mandatory for all systems which implement any resolver software
-- functions.

dnsResConfigImplementIdent OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The implementation identification string for the resolver
             software in use on the system, for example; RES2.1"
     ::= { dnsResConfig 1 }

dnsResConfigService OBJECT-TYPE
     SYNTAX  INTEGER {
          recursive-only (1),
          iterative-only (2),
          recursive-and-iterative (3)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
          "Kind of DNS resolution service provided.
          RECURSIVE-ONLY indicates a stub resolver.
          ITERATIVE-ONLY indicates a normal full service resolver.
          RECURSIVE-AND-ITERATIVE indicates a full service resolver
          which performs a mix of recursive and iterative queries."
     ::= { dnsResConfig 2 }

dnsResConfigMaxCnames OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
          "Limit on how many CNAMEs the resolver should allow before
          deciding that there's a CNAME loop.  Zero means that
          resolver has no explicit CNAME limit."
     ::= { dnsResConfig 3 }

-- DNS Resolver Seat Belt Table
```

```
dnsResConfigSeatBeltTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResConfigSeatBeltEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
          "Table of safety belt information used by the resolver
          when it hasn't got any better idea of where to send a
          query, such as when the resolver is booting or is a stub
          resolver."
     ::= { dnsResConfig 4 }

dnsResConfigSeatBeltEntry  OBJECT-TYPE
     SYNTAX  DnsResConfigSeatBeltEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "An entry in the resolver's seatbelt table."
     INDEX { dnsResConfigSeatBeltAddr,
             dnsResConfigSeatBeltSubTree,
             dnsResConfigSeatBeltClass }
     ::= { dnsResConfigSeatBeltTable 1 }

DnsResConfigSeatBeltEntry ::=
     SEQUENCE {
        dnsResConfigSeatBeltAddr
            IpAddress,
        dnsResConfigSeatBeltName
            DnsName,
        dnsResConfigSeatBeltSvc
            INTEGER,
        dnsResConfigSeatBeltPref
            INTEGER,
        dnsResConfigSeatBeltSubTree
            DnsName,
        dnsResConfigSeatBeltClass
            DnsClass,
        dnsResConfigSeatBeltFile
            OCTET STRING,
        dnsResConfigSeatBeltDate
            DnsDate,
        dnsResConfigSeatBeltValid
            DnsValid
     }
dnsResConfigSeatBeltAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The IP address of the SeatBelt name server identified
             by this row of the table."
     ::= { dnsResConfigSeatBeltEntry 1 }

dnsResConfigSeatBeltName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The DNS name of a SeatBelt nameserver identified by this
             row of the table.  A value of NULL indicates that the name
             is not known by the resolver."
     ::= { dnsResConfigSeatBeltEntry 2 }
```

```
dnsResConfigSeatBeltSvc OBJECT-TYPE
     SYNTAX  INTEGER {
          iterative (1),
          recursive (2),
          recursive-and-iterative (3)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Type-Of-Service resolver expects from seatbelt
             nameserver.
             ITERATIVE indicates that resolver will be directing
             iterative queries to this name server (RD bit turned
             off).
             RECURSIVE indicates that resolver will be directing
             recursive queries to this name server (RD bit turned
             on).
             RECURSIVE-AND-ITERATIVE indicates that the resolver will
             be directing both recursive and iterative queries to the
             server identified in this row of the table."
     ::= { dnsResConfigSeatBeltEntry 3 }

dnsResConfigSeatBeltPref OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This value identifies the preference for the server
             identified in this row of the table.  The lower the
             value, the more desirable the particular server is
             considered."
     ::= { dnsResConfigSeatBeltEntry 4 }

dnsResConfigSeatBeltSubTree OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Queries sent to the Seatbelt name server identified by
              this row of the table are limited to those for names
              in the name subtree identified by this variable.  If no
              such limitation applies, the value of this variable is
              the name of the root domain."
     ::= { dnsResConfigSeatBeltEntry 5 }

dnsResConfigSeatBeltClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The class of DNS queries that will be sent to the
             server identified by this row of the table."
     ::= { dnsResConfigSeatBeltEntry 6 }

dnsResConfigSeatBeltFile OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The name of the file from which the information in
             this row of the table was last initialized or updated.
             The value is NULL if information came from a source other
             than a configuration file."
     ::= { dnsResConfigSeatBeltEntry 7 }
```

```
dnsResConfigSeatBeltDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The date and time that the file named by the
             dnsResConfigSeatBeltFile variable for this row had last
             been updated at the time that this row was last initialized
             or updated.  The value is NULL if unknown or not applicable
             because the dnsResConfigSeatBeltFile variable is NULL."
     ::= { dnsResConfigSeatBeltEntry 8 }

dnsResConfigSeatBeltValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this variable to CLEAR deletes this SeatBelt
             server."
     ::= { dnsResConfigSeatBeltEntry 9 }


-- Server Configuration Group

-- The implementation of the Server Configuration Group is
-- mandatory for all systems which implement DNS server software
-- functions.

dnsServConfigImplementIdent OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The implementation identification string for the DNS
             server software in use on the system, for example;
             FNS2.1"
    ::= { dnsServConfig 1 }

dnsServConfigRecurs  OBJECT-TYPE
     SYNTAX  INTEGER {
          available (1),
          restricted (2),
          unavailable (3)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "This represents the recursion status of requests made
             to this server.  The possible values are: available -
             performs recursion on requests from clients.
             Restricted - recursion is performed on requests only
             from certain clients, for example; clients on an access
             control list.  Unavailable - recursion is not available."
     ::= { dnsServConfig 2 }


-- Resolver Counters Group

-- The implementation of the Resolver Counters Group is mandatory for
-- all systems which implement resolver functions
```

```
dnsResCounterUpTime OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "If the resolver has a persistent state, e.g., a
             process; this value will be the time elapsed since it
             started.  For software that does not have persistence,
             this value will be 0."
    ::= { dnsResCounter 1 }

dnsResCounterResetTime OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed time since cache was reloaded."
    ::= { dnsResCounter 2 }

dnsResCounterInUpkts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of UDP packets received by the resolver
              process(s)."
    ::= { dnsResCounter 3 }

dnsResCounterOutUPkts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of UDP packets sent by the resolver
              process(s)."
    ::= { dnsResCounter 4 }

dnsResCounterTCPInitiatns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of attempts to initiate TCP connections with
             servers."
     ::= { dnsResCounter 5 }

dnsResCounterInTCPMesgs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of received DNS messages over TCP by the
             resolver process."
    ::= { dnsResCounter 6 }

dnsResCounterOutTCPMesgs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of out bound DNS messages sent over TCP by the
             resolver."
    ::= { dnsResCounter 7 }

-- Resolver Counter Table
```

```
dnsResCounterTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResCounterTableEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Table of the current count of resolver queries and
             answers."
    ::= { dnsResCounter 8 }

dnsResCounterTableEntry OBJECT-TYPE
     SYNTAX  DnsResCounterTableEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Entry in the resolver counter table.  Entries are
             indexed by dns OpCode."
     INDEX { dnsResCounterOpCode }
     ::= { dnsResCounterTable 1 }

DnsResCounterTableEntry ::=
     SEQUENCE {
         dnsResCounterOpCode
              DnsOpCode,
         dnsResCounterQueries
              Counter,
         dnsResCounterResponses
              Counter
     }
dnsResCounterOpCode OBJECT-TYPE
     SYNTAX  DnsOpCode -- INTEGER (0..15)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The index to this table.  The OpCodes that have
             already been defined are found in RFC1035."
     ::= { dnsResCounterTableEntry 1 }

dnsResCounterQueries OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries [total] that have sent out by the
             resolver since initialization for the OpCode which is
             the index to this row of the table."
     ::= { dnsResCounterTableEntry 2 }

dnsResCounterResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses [total] that have been received
             by the resolver since initialization for the OpCode
             which is the index to this row of the table."
     ::= { dnsResCounterTableEntry 3 }

-- Resolver Response Code Counter Table

dnsResResponseTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResResponseTableEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Table of the current count of responses to resolver
             queries."
    ::= { dnsResCounter 9 }
```

```
dnsResResponseTableEntry OBJECT-TYPE
     SYNTAX  DnsResResponseTableEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Entry in the resolver response table.  Entries are
             indexed by DNS response code."
     INDEX { dnsResResponseCode }
     ::= { dnsResResponseTable 1 }

DnsResResponseTableEntry ::=
     SEQUENCE {
         dnsResResponseCode
              DnsRespCode,
         dnsResResponses
              Counter
     }
dnsResResponseCode OBJECT-TYPE
     SYNTAX  DnsRespCode -- INTEGER (0..15)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The index to this table.  The Response Codes that have
             already been defined are found in RFC1035."
     ::= { dnsResResponseTableEntry 1 }

dnsResResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses the resolver has received for the
             response code value which identifies this row of the
             table."
     ::= { dnsResResponseTableEntry 2 }

-- Additional DNS Resolver Counter Objects

dnsResCounterNonAuthDataResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests made by the resolver for which a
             non-authoritative answer (cached data) was received."
     ::= { dnsResCounter 10 }

dnsResCounterNonAuthNoDataResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests made by the resolver for which a
             non-authoritative answer - no such data response
             (empty answer) was received."
     ::= { dnsResCounter 11 }

dnsResCounterMartians OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were received from
             servers that the resolver does not think it asked."
     ::= { dnsResCounter 12 }
```

```
dnsResCounterRecdResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received to all queries."
     ::= { dnsResCounter 13 }

dnsResCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were duplicates and
             ignored by the resolver."
     ::= { dnsResCounter 14 }

dnsResCounterUnparseResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were unparseable."
     ::= { dnsResCounter 15 }

dnsResCounterRootAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of attempts made by the resolver to connect
             with root servers."
     ::= { dnsResCounter 16 }

dnsResCounterReferals OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses which were received from
             servers redirecting query to another server."
     ::= { dnsResCounter 17 }

dnsResCounterRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number requests retransmitted for all reasons."
     ::= { dnsResCounter 18 }

dnsResCounterNoResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that were retransmitted because of
             no response."
     ::= { dnsResCounter 19 }

dnsResCounterRootRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that were retransmitted that were to
             root servers."
     ::= { dnsResCounter 20 }
```

```
dnsResCounterInternals OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests internally generated by the
             resolver."
     ::= { dnsResCounter 21 }

dnsResCounterInternalTimeOuts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests internally generated which timed
             out."
     ::= { dnsResCounter 22 }


-- Server Counter Group

-- The implementation of the Server Counter Group is mandatory for
-- all systems which offer either recursive or non recursive server
-- software functions.

-- Several of the objects in this group make distinctions between
-- queries which originate with the local host, within the same
-- zone as the local host or come from outside the zone of the
-- server.  The purpose of these distinctions is to allow for
-- implementations to group queries and responses on this basis.
-- One way in which servers may make these distinctions is by
-- looking at the source IP address of the DNS query.  If the
-- source of the query is 'your own' then the query should be
-- counted as 'yourself' - local host.  If the source of the query
-- matches an 'access list' - the the query came from a friend,
-- that is; within the local zone.  What constitutes an 'access
-- list' is implementation dependent and could be as simple as all
-- hosts on the same IP network as the DNS server are classed as in
-- the local zone and are 'friends'.  In order to avoid double
-- counting, the following rules apply:

-- 1.  No host is in more than one of the three groups defined above.
-- 2.  All queries from the local host are always counted in the
--     'yourself' group regardless of what the access list, if any,
--     says.
-- 3.  The access list should not define 'your friends' in such a way
--     that it includes all hosts, that is 'not everybody is your
--     friend'.


dnsServCounterUTime OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "If the server has a persistent state,  e.g., a
             process; this value will be the time elapsed since it
             started.  For software that does not have persistence,
             this value will be 0."
    ::= { dnsServCounter 1 }

dnsServCounterInUPkts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number [total] of UDP packets received by the server."
    ::= { dnsServCounter 2 }
```

```
dnsServCounterOutUPkts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number [total] of UDP packets sent by the server."
    ::= { dnsServCounter 3 }

dnsServCounterTCPConns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of TCP connections received by the server."
    ::= { dnsServCounter 4 }

dnsServCounterTCPInitiatns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of attempts to initiate TCP connections with
             other servers."
    ::= { dnsServCounter 5 }

dnsServCounterSelfAuthAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been an authoritative answer."
     ::= { dnsServCounter 6 }

dnsServCounterSelfAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been an authoritative no such name answer
             given."
     ::= { dnsServCounter 7 }

dnsServCounterSelfAuthNoDataResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been an authoritative no such data answer
             (empty answer) made."
     ::= { dnsServCounter 8 }

dnsServCounterSelfNonAuthDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             a non-authoritative answer (cached data) was made."
     ::= { dnsServCounter 9 }
```

```
dnsServCounterSelfNonAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been a non-authoritative no such name answer
             given."
     ::= { dnsServCounter 10 }

dnsServCounterSelfNonAuthNoDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             a non-authoritative answer - no such data response was
             made (empty answer)."
     ::= { dnsServCounter 11 }

dnsServCounterSelfCAnys OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host which
             were answers to class=* queries."
     ::= { dnsServCounter 12 }

dnsServCounterSelfInversess OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of inverse queries the server has processed which
             originated from a resolver on the same host."
     ::= { dnsServCounter 13 }

dnsServCounterSelfRefs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries the server has processed
             which originated from a resolver on the same
             host and were referred to other servers."
     ::= { dnsServCounter 14 }

dnsServCounterSelfErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host which
             have been answered with errors (RCODE values other
             than 0 and 3).  RCODE values are defined in RFC1035."
     ::= { dnsServCounter 15 }
```

```
dnsServCounterSelfRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received for names that are only
             1 label long (text form - no internal dots) the
             server has processed which originated from a resolver
             on the same host."
     ::= { dnsServCounter 16 }

dnsServCounterFriendsAuthAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends which
             were authoritatively answered.  The definition of
             friends is a locally defined matter."
     ::= { dnsServCounter 17 }

dnsServCounterFriendsAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends, for
             which authoritative no such name (NXDOMAIN) responses
             were made.  The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 18 }

dnsServCounterFriendsAuthNoDataResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends for
             which authoritative no such data (empty answer)
             responses were made.  The definition of friends is
             a locally defined matter."
     ::= { dnsServCounter 19 }

dnsServCounterFriendsNonAuthDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends which
             were non-authoritatively answered (cached data).
             The definition of friends is a locally defined matter."
     ::= { dnsServCounter 20 }

dnsServCounterFriendsNonAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from friends for which there has been a
             non-authoritative no such name answer given."
     ::= { dnsServCounter 21 }
```

```
dnsServCounterFriendsNonAuthNoDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends which
             were non-authoritatively answered with no such data
             (empty answer)."
     ::= { dnsServCounter 22 }

dnsServCounterFriendsCAnys OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to class=* queries which originated
             from friends.  The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 23 }

dnsServCounterFriendsInverses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses to inverse queries which originated
             from friends. The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 24 }

dnsServCounterFriendsRefs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests which originated from friends
             that were referred to other servers.  The definition
             of friends is a locally defined matter."
     ::= { dnsServCounter 25 }

dnsServCounterFriendsErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from friends and were answered with errors
             (RCODE values other than 0 and 3).  RCODE values are
             defined in RFC1035.  The definition of friends is a
             locally defined matter."
     ::= { dnsServCounter 26 }

dnsServCounterFriendsRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received for names from friends
             that are only 1 label long (text form - no internal
             dots) the server has processed."
     ::= { dnsServCounter 27 }
```

```
dnsServCounterRemoteAuthAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which have
             not been defined as friends, which were
             authoritatively answered."
     ::= { dnsServCounter 28 }

dnsServCounterRemoteAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, for which
             authoritative no such name responses were made."
     ::= { dnsServCounter 29 }

dnsServCounterRemoteAuthNoDataResps OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, for which
             authoritative no such data (empty answer)
             responses were made."
     ::= { dnsServCounter 30 }

dnsServCounterRemoteNonAuthDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, which were
             non-authoritatively answered (cached data)."
     ::= { dnsServCounter 31 }

dnsServCounterRemoteNonAuthNoNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, for which there
             has been a  non-authoritative no such name answer
             given."
     ::= { dnsServCounter 32 }

dnsServCounterRemoteNonAuthNoDatas OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, which were
             non-authoritatively answered with no data
             (empty answer)."
     ::= { dnsServCounter 33 }
```

```
dnsServCounterRemoteCAnys OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to class=* queries which originated
             from systems which have not been defined as friends."
     ::= { dnsServCounter 34 }

dnsServCounterRemoteInverses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to inverse queries which originated
             from systems which have not been defined as friends."
     ::= { dnsServCounter 35 }

dnsServCounterRemoteRefs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests which originated from systems which
             have not been defined as friends that were referred to
             other servers."
     ::= { dnsServCounter 36 }

dnsServCounterRemoteErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from systems which have not been defined as
             friends that were answered with errors (RCODE values other
             than 0 and 3).  RCODE values are defined in RFC1035."
     ::= { dnsServCounter 37 }

dnsServCounterRemoteRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received by the server which originated
             from systems which have not been defined as friends for
             names that are only 1 label long (text form - no
             internal dots)."
     ::= { dnsServCounter 38 }

dnsServCounterReqRefs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of DNS requests refused by the server process."
     ::= { dnsServCounter 39 }

dnsServCounterReqUnparses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received which were unparseable."
     ::= { dnsServCounter 40 }
```

```
dnsServCounterLocalErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests which were aborted for other
             (local) server errors."
     ::= { dnsServCounter 41 }

dnsServCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of duplicate requests which were received and
             not forwarded."
     ::= { dnsServCounter 42 }

-- DNS Server Counter Table

dnsServCounterTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsServCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Counter information based on DNS class and
             record type."
    ::= { dnsServCounter 43 }

dnsServCounterEntry OBJECT-TYPE
     SYNTAX  DnsServCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "This table contains count information for each DNS
             record and class type known to the server.  The index
             allows management software to to create indices to the
             table to get the specific information desired, e.g.,
             number or A record queries over UDP which came to this
             server from sources which have been defined as friends."
    INDEX  { dnsServCounterOpCode, dnsServCounterQClass,
             dnsServCounterQType, dnsServCounterSource,
             dnsServCounterTransport }
    ::= { dnsServCounterTable 1 }

DnsServCounterEntry ::=
    SEQUENCE {
        dnsServCounterOpCode
            DnsOpCode,
        dnsServCounterQClass
            DnsClass,
        dnsServCounterQType
            DnsType,
        dnsServCounterSource
            INTEGER,
        dnsServCounterTransport
            INTEGER,
        dnsServCounterRequests
            Counter,
        dnsServCounterResponses
            Counter
        }
dnsServCounterOpCode OBJECT-TYPE
     SYNTAX  DnsOpCode -- INTEGER (0..15)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
```

```
                "The DNS OpCode being counted in this row of the
                table."
        ::= { dnsServCounterEntry 1 }

    dnsServCounterQClass OBJECT-TYPE
        SYNTAX  DnsClass -- INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The class of record being counted in this row
                of the table."
        ::= { dnsServCounterEntry 2 }

    dnsServCounterQType OBJECT-TYPE
        SYNTAX DnsType -- INTEGER (0..65535)
        ACCESS read-only
        STATUS mandatory
        DESCRIPTION
                "The type of record which is being counted in this row
                in the table."
        ::= { dnsServCounterEntry 3 }

    dnsServCounterSource OBJECT-TYPE
        SYNTAX  INTEGER {
            self (1),
            friend (2),
            remote (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The source of the queries being counted in this row of
                the table.  A value of 1 indicates that the queries reported
                on this row came from a resolver which resides on the same
                system as the server.  A value of 2 indicates the
                queries originated from friends of the server - the
                definition of friends is a locally defined matter.  A
                value of 3 indicates that the queries counted on this
                row of the table came from systems which have not been
                defined as friends."
        ::= { dnsServCounterEntry 4 }

    dnsServCounterTransport OBJECT-TYPE
        SYNTAX  INTEGER {
           udp (1),
           tcp (2),
           other (3)
        }
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A value of 1 indicates that the queries reported on
                this row were sent using UDP.  A value of 2 indicates
                that TCP was used.  3 is for any transport other than
                TCP or UDP."
        ::= { dnsServCounterEntry 5 }

    dnsServCounterRequests OBJECT-TYPE
        SYNTAX  Counter
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Number of requests (queries) that have been recorded
                in this row of the table."
        ::= { dnsServCounterEntry 6 }
```

```
dnsServCounterResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses made by the server since
             initialization for the type of query identified
             on this row of the table."
     ::= { dnsServCounterEntry 7 }

-- Records Group

-- The implementation of the Records Group is mandatory for all
-- systems which implement DNS server software functions.

-- Records Table

-- The records table contains information about the resource
-- records for which the server is authoritative.

dnsRecTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsRecEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Configuration information about all RR records for the
             DNS software."
    ::= { dnsRec 1 }

dnsRecEntry  OBJECT-TYPE
     SYNTAX  DnsRecEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "This table contains information about the resource
             records for which the server is authoritative."
     INDEX { dnsRecZoneName, dnsRecZoneClass, dnsRecName,
             dnsRecType, dnsRecIndex }
     ::= { dnsRecTable 1 }

DnsRecEntry ::=
    SEQUENCE {
        dnsRecZoneName
            DnsName,
        dnsRecZoneClass
            DnsClass,
        dnsRecName
            DnsName,
        dnsRecType
            DnsType,
        dnsRecTTL
            DnsTime,
        dnsRecData
            OCTET STRING,
        dnsRecIndex
            INTEGER,
        dnsRecHits
            Counter
        }
dnsRecZoneName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Zone name to which the Resource Record which
             is identified in this row of the table belongs.
             This is the owner name of the zone's SOA RR, as
```

```
                described in RFC1034."
        ::= { dnsRecEntry 1 }

    dnsRecZoneClass OBJECT-TYPE
        SYNTAX  DnsClass -- INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "DNS class of the zone contained in this row.  For classes
                listed in the DNS specification, the values are the same."
        ::= { dnsRecEntry 2 }

    dnsRecName OBJECT-TYPE
        SYNTAX  DnsName  --  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Name of the Resource Record which is identified in
                this row of the table.  As described in RFC1034, the
                owner of the record is the domain name where the RR is
                found."
        ::= { dnsRecEntry 4 }

    dnsRecType OBJECT-TYPE
        SYNTAX  DnsType  --  INTEGER (0..65535)
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The type of the record identified in this row of the
                table.  For types defined in the DNS specification,
                the values are the same."
        ::= { dnsRecEntry 5 }

    dnsRecTTL OBJECT-TYPE
        SYNTAX  DnsTime
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "The larger of the Time-To-Live value for this record
                and the Zone Minimum for the zone containing it."
        ::= { dnsRecEntry 6 }

    dnsRecData OBJECT-TYPE
        SYNTAX  OCTET STRING
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "Variable length of octets.  Type and Class information
                provided in this row of the table along with this data
                tell the management station how to interpret the Record.
                For information on the details of DNS Resource Records
                and their formats, see RFC1035."
        ::= { dnsRecEntry 7 }

    dnsRecIndex OBJECT-TYPE
        SYNTAX  INTEGER
        ACCESS  read-only
        STATUS  mandatory
        DESCRIPTION
                "A unique index value which identifies each entry in
                this table."
        ::= { dnsRecEntry 8 }
```

```
dnsRecHits OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of hits the server has had on the resource record
             defined in this row of the table.  A hit on a RR is counted
             whenever it is included in a response."
     ::= { dnsRecEntry 9 }

-- Resolver Cache Group

-- The implementation of the Resolver Cache Group is mandatory for
-- all systems which implement a cache.

dnsResCacheStatus OBJECT-TYPE
     SYNTAX  INTEGER {
          enabled (1),
          disabled (2)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Status of the resolver's cache.
             ENABLED means that the use of the cache is allowed.
             DISABLED means that the cache is not being used."
     ::= { dnsResCache 1 }

dnsResCacheValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this to CLEAR deletes the resolver's entire
             cache."
     ::= { dnsResCache 2 }

dnsResCacheMaxTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Maximum Time-To-Live for RRs in this cache.  If the
             resolver does not implement a TTL ceiling, the value of
             this field should be hexadecimal FFFFFFFF."
     ::= { dnsResCache 3 }

-- Resolver Cache Table

-- The Resolver Cache Table contains information about Resource
-- Records currently in the resolver's cache.

dnsResCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "The entire contents of the resolver's cache."
    ::= { dnsResCache 4 }
```

```
dnsResCacheEntry OBJECT-TYPE
     SYNTAX  DnsResCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "A single entry in the resolvers's cache."
     INDEX { dnsResCacheName, dnsResCacheClass,
             dnsResCacheType, dnsResCacheIndex  }
     ::= { dnsResCacheTable 1 }

DnsResCacheEntry ::=
     SEQUENCE {
         dnsResCacheName
             DnsName,
         dnsResCacheClass
             DnsClass,
         dnsResCacheType
             DnsType,
         dnsResCacheTTL
             DnsTime,
         dnsResCacheElapsedTTL
             DnsTime,
         dnsResCacheSource
             IpAddress,
         dnsResCacheRData
             OCTET STRING,
         dnsResCacheHits
             Counter,
         dnsResCacheEntryValid
             DnsValid,
         dnsResCacheIndex
             INTEGER
     }
dnsResCacheName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Owner name of the Resource Record in the cache which
             is identified in this row of the table.  As described
             in RFC1034, the owner of the record is the domain name
             were the RR is found."
     ::= { dnsResCacheEntry 1 }

dnsResCacheClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS class of the Resource Record in the cache which
             is identified in this row of the table."
     ::= { dnsResCacheEntry 2 }

dnsResCacheType OBJECT-TYPE
     SYNTAX  DnsType -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS type of the Resource Record in the cache which
             is identified in this row of the table."
     ::= { dnsResCacheEntry 3 }
```

```
dnsResCacheTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Time-To-Live of RR in DNS cache.  This is the initial
             TTL value which was received with the RR when it was
             originally received."
     ::= { dnsResCacheEntry 4 }

dnsResCacheElapsedTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since RR was received."
     ::= { dnsResCacheEntry 5 }

dnsResCacheSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Host from which RR was received, 0.0.0.0 if unknown."
     ::= { dnsResCacheEntry 6 }

dnsResCacheRData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "RDATA portion of a cached RR."
     ::= { dnsResCacheEntry 7 }

dnsResCacheHits OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The number of hits on the cached RR identified in
             this row of the table."
     ::= { dnsResCacheEntry 8 }

dnsResCacheEntryValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Set to CLEAR to delete this RR entry from the
             cache table."
     ::= { dnsResCacheEntry 9 }

dnsResCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "A unique index value which identifies each entry in
             this table."
     ::= { dnsResCacheEntry 10 }
```

```
-- Resolver Negative Cache Group

-- The implementation of the Resolver Negative Cache group is
-- mandatory for all systems which implement negative a negative
-- cache function.

dnsResNCacheStatus OBJECT-TYPE
     SYNTAX  INTEGER {
          enabled (1),
          disabled (2)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Status of the resolver's negative cache.
             ENABLED means that the use of the negative cache is
             allowed.
             DISABLED means that the negative caching is not being
             used."
     ::= { dnsResNCache 1 }

dnsResNCacheValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this to CLEAR deletes the resolver's entire
             negative response cache."
     ::= { dnsResNCache 2 }

dnsResNCacheMaxTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Maximum Time-To-Live for cached authoritative errors.
             If the resolver does not implement a TTL ceiling, the
             value of this field should be hexadecimal FFFFFFFF."
     ::= { dnsResNCache 3 }

-- Resolver Negative Cache Table

-- The Resolver Negative Cache Table contains information about
-- Resource Records currently in the resolver's cache.

dnsResNCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResNCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "The resolver's negative response cache."
     ::= { dnsResNCache 4 }

dnsResNCacheEntry OBJECT-TYPE
     SYNTAX  DnsResNCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "An entry in the resolver's negative response cache."
     INDEX { dnsResNCacheErrQName, dnsResNCacheErrQClass,
             dnsResNCacheErrQType, dnsResNCacheIndex  }
     ::= { dnsResNCacheTable 1 }
```

```
DnsResNCacheEntry ::=
     SEQUENCE {
         dnsResNCacheErrQName
             DnsName,
         dnsResNCacheErrQClass
             DnsQClass,
         dnsResNCacheErrQType
             DnsQType,
         dnsResNCacheErrTTL
             DnsTime,
         dnsResNCacheErrElapsedTTL
             DnsTime,
         dnsResNCacheErrSource
             IpAddress,
         dnsResNCacheErrCode
             INTEGER,
         dnsResNCacheHits
             Counter,
         dnsResNCacheErrValid
             DnsValid,
         dnsResNCacheIndex
             INTEGER
     }
dnsResNCacheErrQName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "QNAME associated with a cached authoritative error."
     ::= { dnsResNCacheEntry 1 }

dnsResNCacheErrQClass OBJECT-TYPE
     SYNTAX  DnsQClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
            "DNS QCLASS associated with a cached authoritative error."
     ::= { dnsResNCacheEntry 2 }

dnsResNCacheErrQType OBJECT-TYPE
     SYNTAX  DnsQType -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS QTYPE associated with a cached authoritative error."
     ::= { dnsResNCacheEntry 3 }

dnsResNCacheErrTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Time-To-Live of a cached authoritative error at the
             time of the error, it should not be decremented
             by the number of seconds since it was recieved. This
             should be the TTL as copied from the MINIMUM field of
             the SOA that accompanied the authoritative error."
     ::= { dnsResNCacheEntry 4 }

dnsResNCacheErrElapsedTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since authoritative error was received."
     ::= { dnsResNCacheEntry 5 }
```

```
dnsResNCacheErrSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Host which sent the authoritative error, 0.0.0.0 if
             unknown."
     ::= { dnsResNCacheEntry 6 }

dnsResNCacheErrCode OBJECT-TYPE
     SYNTAX  INTEGER {
         nonexistantName (1),
         noData (2),
         other (3)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
            "The authoritative error that has been cached.
             NonexistantName indicates a cached authoritative name
             error.  NoData indicates a cached authoritative
             response with no error and no relevant data."
     ::= { dnsResNCacheEntry 7 }

dnsResNCacheHits OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The number of hits on the RR identified in
             this row of the table."
     ::= { dnsResNCacheEntry 8 }

dnsResNCacheErrValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
            "Set to CLEAR to delete this cached error."
     ::= { dnsResNCacheEntry 9 }

dnsResNCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "A unique index value which identifies each entry in
             the Negative Cache table."
     ::= { dnsResNCacheEntry 10 }

-- Resolver Statistics Group

-- The implementation of the Resolver Statistics Group is
-- mandatory for all resolver software implementations.

dnsResStatGoodCaches OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of RRs the resolver has cached successfully.
             RRs which have already been successfully cached are
             not counted again."
     ::= { dnsResStats 1 }
```

```
dnsResStatBadCaches OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of RRs the resolver has refused to cache
              because they appear to be dangerous or irrelevant.
              E.g., RRs with suspiciously high TTLs, unsolicited root
              information, or that just don't appear to be relevant
              to the question the resolver asked.  RRs which the
              resolver has refused to cache previously are not
              counted again."
     ::= { dnsResStats 2 }

dnsResStatGoodNCaches OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of authoritative errors the resolver has cached
              successfully."
     ::= { dnsResStats 3 }

dnsResStatBadNCaches OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of authoritative errors the resolver would
              have liked to cache but was unable to because the
              appropriate SOA RR was not supplied or looked
              suspicious."
     ::= { dnsResStats 4 }

-- Lame Delegation Table

dnsResLameDelegationTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResLameDelegationEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
            "Table of name servers returning lame delegations."
     ::= { dnsResStats 5 }

dnsResLameDelegationEntry OBJECT-TYPE
     SYNTAX  DnsResLameDelegationEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
         "Entry in lame delegation table."
     INDEX { dnsResLameDelegationSource,
             dnsResLameDelegationName,
             dnsResLameDelegationClass }
     ::= { dnsResLameDelegationTable 1 }
```

```
DnsResLameDelegationEntry ::=
     SEQUENCE {
         dnsResLameDelegationSource
             IpAddress,
         dnsResLameDelegationName
             DnsName,
         dnsResLameDelegationClass
             DnsClass,
         dnsResLameDelegationCounts
             Counter,
         dnsResLameDelegationValid
             DnsValid
     }
dnsResLameDelegationSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Source of lame delegation."
     ::= { dnsResLameDelegationEntry 1 }

dnsResLameDelegationName OBJECT-TYPE
     SYNTAX  DnsName
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS name for which lame delegation was received."
     ::= { dnsResLameDelegationEntry 2 }

dnsResLameDelegationClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS class of received lame delegation."
     ::= { dnsResLameDelegationEntry 3 }

dnsResLameDelegationCounts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "How many times this lame delegation has been received."
     ::= { dnsResLameDelegationEntry 4 }

dnsResLameDelegationValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
            "Set to CLEAR to delete this lame delegation entry."
     ::= { dnsResLameDelegationEntry 5 }

-- Server Management Group

-- The implementation of the Server Management Group is mandatory
-- for all systems which implement DNS server software functions.
```

```
dnsServMgmtReload OBJECT-TYPE
     SYNTAX  INTEGER {
          reload (1),
          other (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "This read-write variable is used to force a database
             reload with a restart of the name server process
             When set to the values above, the following actions
             are taken: reload - Reload and restart name server
             process(s).  Other is returned when a read operation
            is performed on this object."
     ::= { dnsServMgmt 1 }

-- DNS Managment Zone Configuration Table

dnsServMgmtZoneTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsServMgmtZoneEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Table of zones for which this name server is
             configured.  If name server doesn't load any zones,
             this table is empty."
     ::= { dnsServMgmt 2 }

dnsServMgmtZoneEntry OBJECT-TYPE
     SYNTAX  DnsServMgmtZoneEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "An entry in the name server zone table."
     INDEX { dnsServMgmtZoneMName,
             dnsServMgmtZoneClass }
     ::= { dnsServMgmtZoneTable 1 }

DnsServMgmtZoneEntry ::=
     SEQUENCE {
         dnsServMgmtZoneMName
             DnsName,
         dnsServMgmtZoneRName
             DnsName,
         dnsServMgmtZoneSerial
             INTEGER,
         dnsServMgmtZoneRefresh
             DnsTime,
         dnsServMgmtZoneRetry
             DnsTime,
         dnsServMgmtZoneExpire
             DnsTime,
         dnsServMgmtZoneMin
             DnsTime,
         dnsServMgmtZoneLoad
             OCTET STRING,
         dnsServMgmtZoneDate
             DnsDate,
         dnsServMgmtZoneClass
             DnsClass,
         dnsServMgmtZoneLastReload
             DnsTime,
         dnsServMgmtZoneLastReloadAttempt
             DnsTime,
         dnsServMgmtZoneLastSource
             IpAddress,
```

```
            dnsServMgmtZoneValid
                DnsValid
        }
    dnsServMgmtZoneMName OBJECT-TYPE
         SYNTAX  DnsName
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "DNS name of the server that was the original or
                 primary source of the data for this zone - see RFC
                 1035 for details of the SOA RDATA format."
         ::= { dnsServMgmtZoneEntry 1 }

    dnsServMgmtZoneRName OBJECT-TYPE
         SYNTAX  DnsName
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "The mailbox address of the person who is responsible
                 for the administration of the zone specified in this
                 row of the table.  Note that the format of the address
                 follows the domain-name format as in
                 postman.server.acme."
         ::= { dnsServMgmtZoneEntry 2 }

    dnsServMgmtZoneSerial OBJECT-TYPE
         SYNTAX  INTEGER
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "The current serial number of the SOA record in the
                 file specified by dnsServMgmtZoneLoad."
         ::= { dnsServMgmtZoneEntry 3 }

    dnsServMgmtZoneRefresh OBJECT-TYPE
         SYNTAX  DnsTime
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "REFRESH interval for zone.  This is the REFRESH field
                  of the zone's SOA RR, if known, otherwise a value
                  of zero must be returned."
         ::= { dnsServMgmtZoneEntry 4 }

    dnsServMgmtZoneRetry OBJECT-TYPE
         SYNTAX  DnsTime
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "RETRY interval for zone.  This is the RETRY field of
                 the zone's SOA RR, if known, otherwise a value of 0
                 must be returned."
         ::= { dnsServMgmtZoneEntry 5 }

    dnsServMgmtZoneExpire OBJECT-TYPE
         SYNTAX  DnsTime
         ACCESS  read-only
         STATUS  mandatory
         DESCRIPTION
                 "EXPIRE interval for zone.  This is the EXPIRE field of
                 the zone's SOA RR, if known, otherwise a value of 0
                 must be returned."
         ::= { dnsServMgmtZoneEntry 6 }
```

```
dnsServMgmtZoneMin OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This value is the minimum TTL for the zone, as specified
             in the zone minimum parameter of the zone's SOA record."
    ::= { dnsServMgmtZoneEntry 7 }

dnsServMgmtZoneLoad OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The name of the zone file read to load the data
             for this zone when the zone was last loaded or updated.
             A value of NULL indicates that the data was not loaded
             from a named file (e.g., because it was downloaded from
             another server using the DNS protocol's zone transfer
             facility)."
    ::= { dnsServMgmtZoneEntry 8 }

dnsServMgmtZoneDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This is the date that the information found in
             dnsServMgmtZoneLoad was last changed at the time
             the zone was last loaded. A value of NULL indicates
             that there is no dnsServMgmtZoneLoad file."
    ::= { dnsServMgmtZoneEntry 9 }

dnsServMgmtZoneClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS class of the RRs in this zone."
     ::= { dnsServMgmtZoneEntry 10 }

dnsServMgmtZoneLastReload OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since last successful reload of
              this zone."
     ::= { dnsServMgmtZoneEntry 11 }

dnsServMgmtZoneLastReloadAttempt OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since last attempted reload of
             this zone."
     ::= { dnsServMgmtZoneEntry 12 }
```

```
dnsServMgmtZoneLastSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "IP address of host from which most recent reload of
              this zone was received.  If unknown or irrelevant,
              value should be 0.0.0.0."
     ::= { dnsServMgmtZoneEntry 13 }

dnsServMgmtZoneValid OBJECT-TYPE
     SYNTAX  DnsValid
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Set to CLEAR to delete this zone entry."
     ::= { dnsServMgmtZoneEntry 14 }

-- DNS Management Server Table

dnsServMgmtServTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsServMgmtServEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
          "This table is a list of zones which the server tries to
          acquire using DNS zone transfer operations.  It lists
          for each zone, the set of IP addresses the server is
          configured to query when trying to obtain the data for
          that zone."
     ::= { dnsServMgmt 3 }

dnsServMgmtServEntry OBJECT-TYPE
     SYNTAX  DnsServMgmtServEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "An entry in the name server server (sic) table."
     INDEX { dnsServMgmtServName,
             dnsServMgmtServClass,
             dnsServMgmtServAddr }
     ::= { dnsServMgmtServTable 1 }

DnsServMgmtServEntry ::=
     SEQUENCE {
         dnsServMgmtServName
             DnsName,
         dnsServMgmtServClass
             DnsClass,
         dnsServMgmtServAddr
             IpAddress
     }

dnsServMgmtServName OBJECT-TYPE
     SYNTAX  DnsName
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS name of the zone to which this entry applies."
     ::= { dnsServMgmtServEntry 1 }

dnsServMgmtServClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS class of zone to which this entry applies."
     ::= { dnsServMgmtServEntry 2 }
```

```
dnsServMgmtServAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "IP address of name server host from which this zone
             might be obtainable."
     ::= { dnsServMgmtServEntry 3 }
END
```

6 Acknowledgements

This document is the result of work undertaken the by DNS working group. In addition, the contributions and comments of the following members are also specially acknowledged:

Win Treese, Digital Equipment Corporation
Philip Almquist, Computer Communication consultant
Joe Peck, Digital Equipment Corporation
Mimi Zohar, IBM

7 References

J. Reynolds and J. Postel, Assigned Numbers. Internet Working Group Request for Comments 1010. Network Information Center, SRI International, Menlo Park, California, (May, 1987).

M. Stahl, Domain Administrators Guide. Internet Working Group Request for Comments 1032. Network Information Center, SRI International, Menlo Park, California, (November, 1987).

M. Lottor, Domain Administrators Operations Guide, Internet Working Group Request for Comments 1033. Network Information Center, SRI International, Menlo Park, California, (November, 1987).

P. Mockapetris, Domain Names - Concepts and Facilities, Internet Working Group Request for Comments 1034. Network Information Center, SRI International, Menlo Park, California, (November, 1987).

P. Mockapetris, Domain Names - Implementation and Specification, Internet Working Group Request for Comments 1035. Network Information Center, SRI International, Menlo Park, California, (November, 1987).

V. Cerf, IAB Recommendations for the Development of Internet Network Management Standards. Internet Working Group Request for Comments 1052. Network Information Center, SRI International, Menlo Park, California, (April, 1988).

R. Braden (editor) Requirements for Internet Hosts—Application and Support, Internet Working Group Request for Comments 1123. Network Information Center, SRI International,Menlo Park, California, (October, 1989).

M.T. Rose and K. McCloghrie, Structure and Identification of Management Information for TCP/IP-based internets, Internet Working Group Request for Comments 1155. Network Information Center, SRI International, Menlo Park, California, (May, 1990).

K. McCloghrie and M.T. Rose, Management Information Base for Network Management of TCP/IP-based internets, Internet Working Group Request for Comments 1156, Network Information Center, SRI International, Menlo Park, California, (May, 1990).

J.D. Case, M.S. Fedor, M.L. Schoffstall, and J.R. Davin, Simple Network Management Protocol, Internet Working Group Request for Comments 1157. Network Information Center, SRI International, Menlo Park, California, (May, 1990).

M.T. Rose, The Open Book, A Practical Perspective on OSI. Prentice Hall, Englewood Cliffs, New Jersey, (1990).

M.T. Rose and K. McCloghrie (editors) Concise MIB Definitions, Internet Working Group Request for Comments 1212, Network Information Center, SRI International, Menlo Park, California, (March, 1991).

K. McCloghrie and M.T. Rose (editors), Management Information Base for Network Management of TCP/IP-based internets: MIB-II, Internet Working Group Request for Comments 1213. Network Information Center, SRI International, Menlo Park, California, (March, 1991).