DNS MIB Extensions

5–March–1992

DNS Working Group

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

Status of this Memo

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 editor.

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 client 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

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.

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

2 Selected Objects

The objects included in this memo have been created from information contained in the relevant DNS related RFCs which include: Domain Administrators Guide (RFC 1032), Domain Administrators Operations Guide (RFC 1033) Domain Names - Concepts and Facilities (RFC 1034), Domain Names - Implementation and Specification (RFC 1035), and Requirements for Internet Hosts - Application and Support (RFC 1123). In addition, objects have been created based on expected operational need and from the example of existing DNS specific tools. These objects have be ordered into groups as follows:

- General Configuration Group
- Client Configuration Group
- Server Configuration Group
- Resolver Counter Group
- Server Counter Group
- Full Service Resolver Counter Group
- Recursive Server Counter Group
- Non Recursive Server Counter Group
- A Records Group
- NS Records Group
- CNAME Records Group
- WKS Records Group
- PTR Records Group
- HINFO Records Group

- MX Records Group
- Undefined Records Group
- Server Management Group

Many 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.

3 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 Concepts and Facilities - RFC 1034.

3.1 Format of Definitions

An object in this memo is specified by five fields of information: Object, Syntax, Definition, 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.

DEFINITION: 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.

3.2 Textual Conventions

A new datatype has been introduced as a textual convention in this DNS MIB document. This addition will facilitate the common understanding of information used by the DNS. No changes to the SMI the the SNMP are necessary to support these conventions which are described in 4.0 (Object Definitions).

4 Object Definitions

```
     RFCxxxx-dnsMIB

     DEFINITIONS ::= BEGIN

     IMPORTS
         mgmt, OBJECT-TYPE, NetworkAddress, IpAddress,
         Counter, Gauge, TimeTicks
           FROM RFC1155-SMI
         DisplayString
           FROM RFC1213;

-- DNS MIB

     dns             OBJECT IDENTIFIER ::= { experimental xx }

-- textual conventions

DnsDate ::= DisplayString
-- This data type is intended to provide a consistent
-- method of reporting date information.  The first four
-- octets represent the year, the next two month of the year
-- expressed numerically, the next four octets represent the day
-- followed by hour of the day using a 24 hour clock followed by a
-- decimal point.  The final two octets represent tenths and
-- hundreths of a second.  This data type is derived from one of the
-- ASN.1 representations of date and time called GeneralizedTime.

-- groups in the dns mib

      genConfig               OBJECT IDENTIFIER ::= { dns 1 }
      clientConfig            OBJECT IDENTIFIER ::= { dns 2 }
      serverConfig            OBJECT IDENTIFIER ::= { dns 3 }
      resolverCounters        OBJECT IDENTIFIER ::= { dns 4 }
      serverCounters          OBJECT IDENTIFIER ::= { dns 5 }
      fullSvcResolvCounters   OBJECT IDENTIFIER ::= { dns 6 }
      recursServerCounters    OBJECT IDENTIFIER ::= { dns 7 }
      nonRecursServerCounters OBJECT IDENTIFIER ::= { dns 8 }
      aRecords                OBJECT IDENTIFIER ::= { dns 9 }
      nsRecords               OBJECT IDENTIFIER ::= { dns 10 }
      cnameRecords            OBJECT IDENTIFIER ::= { dns 11 }
      wksRecords              OBJECT IDENTIFIER ::= { dns 16 }
      ptrRecords              OBJECT IDENTIFIER ::= { dns 17 }
      hinfoRecords            OBJECT IDENTIFIER ::= { dns 18 }
      mxRecords               OBJECT IDENTIFIER ::= { dns 19 }
      UndefRecors             OBJECT IDENTIFIER ::= { dns 21 }
      serverManagement        OBJECT IDENTIFIER ::= { dns 22 }

-- General Configuration Group

-- The implementation of the General Configuration group is
-- mandatory for all systems which implement any DNS software
-- functions.
```

```
dnsType OBJECT-TYPE
     SYNTAX  INTEGER {
          full_service_resolver (1),
          stub_resolver (2),
          server (3),
          recurs_server (4)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This value indicates the type of DNS service available
      on the system. A value of 1 indicates that the system
      will process DNS requests locally, while a value of 2
      means that the request will be sent to another system
      for processing.  The only difference between a value of
      3 and 4 is that 4 indicates that recursion will be
      performed on request."
     ::= { general 1 }

dnsDomain OBJECT-TYPE
     SYNTAX  DisplayString (SIZE (0..255) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The fully qualified domain name of the system."
     ::= { general 2 }

dnsResponseCaching OBJECT-TYPE
     SYNTAX  INTEGER {
          negative (1),
          positive (2),
          no_caching (3)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "A value of 1 indicates that negative response caching
      has been enabled on this system.  This means that both
      the data and the name of the source of the data are
      stored.  A value of 2 indicates that data only is
      cached."
     ::= { general 3 }

dnsCacheSeatBelt OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The default cache that is used to find other servers
      on startup.  Not all systems will store this
      information, so a null response indicates that there is
      no static cache information stored."
    ::= { general 4 }

-- Client Configuration Group

-- The implementation of the Client Configuration group is
-- mandatory for all systems which implement DNS client software
-- functions.

-- Client Configuration Table
```

```
dnsClientConfigTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsClientConfigEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Information about default servers used by the DNS
      client software at system start."
    ::= { clientConfig 1 }

dnsClientConfigEntry OBJECT-TYPE
     SYNTAX  dnsClientConfigEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains default server information for
      each server configured for the client."
    INDEX  { dnsClientConfigIndex }
    ::= { dnsClientConfigTable 1 }

DnsClientConfigEntry ::=
    SEQUENCE {
        dnsClientConfigIndex
            INTEGER,
        dnsClientConfigServer
            DisplayString,
        dnsClientConfigServerPreference
            INTEGER (1..255),
        dnsClientConfigServerAddr
            IPAddress
     }
dnsClientConfigIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value for each entry in the
      dnsClientConfigTable."
    ::= { dnsClientConfigEntry 1 }

dnsClientConfigServer OBJECT-TYPE
     SYNTAX  DisplayString ( SIZE (0..255) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The fully qualified domain name of the server
      described by this entry."
    ::= { dnsClientConfigEntry 2 }

dnsClientConfigServerPreference OBJECT-TYPE
     SYNTAX  INTEGER (1..255)
     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 higher the preference for the particular
      server."
    ::= {dnsClientConfigEntry 3 }

dnsClientConfigServerAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The IP address configured for the server."
     ::= { dnsClientConfigEntry 4 }
```

```
-- Client Seat Belt Table

dnsClientSeatBeltTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsClientSeatBeltEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Information about default information read in at start
      up time by the resolver software."
    ::= { clientConfig 2 }

dnsClientSeatBeltEntry OBJECT-TYPE
     SYNTAX  dnsClientSeatBeltEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains default information for startup of
      the resolver software."
    INDEX  { dnsClientSeatBeltIndex }
    ::= { dnsClientSeatBeltTable 1 }

DnsClientSeatBeltEntry ::=
    SEQUENCE {
        dnsClientSeatBeltIndex
            INTEGER,
        dnsClientSeatBeltBootLoad
            DisplayString,
        dnsClientSeatBeltBootDate
            DnsDate
     }
dnsClientSeatBeltIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value for each entry in the
      dnsClientSeatBeltTable."
    ::= { dnsClientSeatBeltEntry 1 }

dnsClientSeatBeltBootLoad OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name of any file that is read for initial client
      information."
     ::= { dnsClientSeatBeltEntry 2  }

dnsClientSeatBeltBootDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the date information found in
      dnsClientSeatBeltBootLoad was last changed."
     ::= { dnsClientSeatBeltEntry 3 }

-- Additional Client Configuration Objects
```

```
dnsClientConfigRecurs  OBJECT-TYPE
     SYNTAX  INTEGER {
          on (1),
          off (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "This represents the recursion status of the client
      requests on queries.  The values are: on Requests
      recursion on queries.  off Recursion is not requested
      on queries."
     ::= { clientConfig 3 }

dnsClientConfigDefaultDomain OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Default domain name appended to unqualified names for
      DNS queries. "
     ::= { clientConfig 4 }

dnsClientConfigDnsUse 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."
     ::= { clientConfig 5 }

-- Server Configuration Group

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

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

```
dnsServerConfigRecurs  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 from
      clients which are on an access control list unavailable
      Recursion is not available."
     ::= { serverConfig 2 }

-- Resolver Counters Group

-- The implementation of the Resolver Counters Group is mandatory for
-- all systems which implement either stub or full service resolver
-- software functions.

dnsResCounterBootTime OBJECT-TYPE
     SYNTAX  TimeTicks
     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."
    ::= { resCounter 1 }

dnsResCounterResetTime OBJECT-TYPE
     SYNTAX  TimeTicks
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "For those resolvers with a persistent state, this
      value will be the time elapsed since the last reset
      event.  A reset event occurs when resolver process is
      caused to reload it's configuration data.  For those
      resolvers without a persistent state, the value
      returned will be 0."
    ::= { resCounter 2 }

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

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

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

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

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

dnsResCounterRegReqsSent OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of regular (QUERY) queries that have been sent
      by the resolver."
     ::= { resCounter 8 }

dnsResCounterInvReqsSent OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of inverse (IQUERY) queries that have been sent
      by the resolver."
     ::= { resCounter 9 }

dnsResCounterNonAuthData 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 made by the
      local server."
     ::= { resCounter 10 }

dnsResCounterNonAuthNoData 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 was
      made (empty answer) by the local server."
     ::= { resCounter 11 }
```

```
dnsResCounterIAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to inverse queries made by the
      resolver which have been answered with local
      information by the local server."
     ::= { resCounter 12 }

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."
     ::= { resCounter 13 }

dnsResCounterResponsesRecd OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received to all queries."
     ::= { resCounter 14 }

dnsResCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were duplicates and
      ignored"
     ::= { resCounter 15 }

dnsResCounterUnparse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were unparseable."
     ::= { resCounter 16 }

dnsResCounterServerFail OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were server failed
      responses."
     ::= { resCounter 17 }

dnsResCounterOtherFail OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which where for all other
      bad responses."
     ::= { resCounter 18 }
```

```
dnsResCounterLocalRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests received for relative names from
      the resolver on the local host."
     ::= { resCounter 19 }

-- 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'.

dnsServCounterBootTime OBJECT-TYPE
     SYNTAX  TimeTicks
     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."
    ::= { servCounter 1 }

dnsServCounterResetTime OBJECT-TYPE
     SYNTAX  TimeTicks
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "For those servers with a persistent state, this value
      will be the time elapsed since the last reset event.  A
      reset event occurs when the server process is caused to
      reload it configuration data.  For those servers
      without a persistent state, the value returned will be
      0."
    ::= { servCounter 2 }
```

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

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

dnsServCounterTCPConns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of TCP connections received by the server
      processes(s)."
    ::= { servCounter 5 }

dnsServCounterTCPInitiated OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of attempt to initiate TCP connections with
      other servers."
    ::= { servCounter 6 }

dnsServCounterTCPMesgsIn OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of received DNS messages over TCP by the server
      process."
    ::= { servCounter 7 }

dnsServCounterTCPMesgsOut OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of out bound DNS messages over TCP sent by the
      server."
    ::= { servCounter 8 }

dnsServCounterRegReqsRecd OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of regular (QUERY) queries that have been
      received."
     ::= { servCounter 9 }
```

```
dnsServCounterRegReqsSent OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of regular (QUERY) queries that have been sent
      by the server."
     ::= { servCounter 10 }

dnsServCounterInvReqsRecd OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of inverse (IQUERY) queries that have been
      received by the server."
     ::= { servCounter 11 }

dnsServCounterAuthData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests by the local host that have been
      answered locally with authoritative data."
     ::= { servCounter 12 }

dnsServCounterAuthNoName OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests made by the local host for which
      there has been an authoritative no such name answer
      (NXDOMAIN) given by the local server."
     ::= { servCounter 13 }

dnsServCounterAuthNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests made by the local host for which
      there has been an authoritative no such data answer
      (empty answer) made by the local server."
     ::= { servCounter 14 }

dnsServCounterNonAuthData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests made by the local host for which a
      non- authoritative answer (cached data) was made."
     ::= { servCounter 15 }

dnsServCounterNonAuthNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests made by the local host for which a
      non- authoritative answer - no such data response was
      made (empty answer)."
     ::= { servCounter 16 }
```

```
dnsServCounterCAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to C_ANY queries that have been made
      by the local host answered by the local server."
     ::= { servCounter 17 }

dnsServCounterIAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to inverse queries made by the local
      host which have been answered with local information by
      this server process."
     ::= { servCounter 18 }

dnsServCounterNonRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of non recursive queries by the local host
      which have been referred to other servers by the local
      server process."
     ::= { servCounter 19 }

dnsServCounterErrorResponse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries by the local host which have been
      answered with error responses (e.g., malformed query)
      by the local server process."
     ::= { servCounter 20 }

dnsServCounterZoneAuth OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating in the local zone which
      were authoritatively answered."
     ::= { servCounter 21 }

dnsServCounterZoneNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating in the local zone for
      which authoritative no such name (NXDOMAIN) responses
      were made."
     ::= { servCounter 22 }

dnsServCounterZoneNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating in the local zone for
      which authoritative no such data (empty answer)
      responses were made."
     ::= { servCounter 23 }
```

```
dnsServCounterZoneCached OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating in the local zone which
      were non-authoritatively answered (cached data)."
     ::= { servCounter 24 }

dnsServCounterZoneCachedNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating in the local zone for
      were non-authoritatively answered with no such data
      (empty answer)."
     ::= { servCounter 25 }

dnsServCounterZoneAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to C_ANY queries which originated in
      the local zone."
     ::= { servCounter 26 }

dnsServCounterZoneInverse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to inverse queries which originated
      in the local zone."
     ::= { servCounter 27 }

dnsServCounterZoneForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which originated in the local zone
      that were forwarded to other servers."
     ::= { servCounter 28 }

dnsServCounterZoneRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of non recursive queries which originated in
      the local zone which have been referred to other
      servers."
     ::= { servCounter 29 }

dnsServCounterZoneErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries which originated in the local zone
      which were answered with error responses."
     ::= { servCounter 30 }
```

```
dnsServCounterRemoteAuth OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating outside the local zone
      which were authoritatively answered."
     ::= { servCounter 31 }

dnsServCounterRemoteNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating outside the local zone
      for which authoritative no such name (NXDOMAIN)
      responses were made."
     ::= { servCounter 32 }

dnsServCounterRemoteNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating outside the local zone
      for which authoritative no such data (empty answer)
      responses were made."
     ::= { servCounter 33 }

dnsServCounterRemoteCached OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating outside the local zone
      which were non-authoritatively answered (cached data)."
     ::= { servCounter 34 }

dnsServCounterRemoteCachedNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries originating outside the local zone
      which were non-authoritatively answered with no such
      data (empty answer)."
     ::= { servCounter 35 }

dnsServCounterRemoteAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to C_ANY queries which originated
      outside the local zone."
     ::= { servCounter 36 }

dnsServCounterRemoteInverse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of answers to inverse queries which originated
      outside the local zone."
     ::= { servCounter 37 }
```

```
dnsServCounterRemoteForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which originated outside the local
      zone that were forwarded to other servers."
     ::= { servCounter 38 }

dnsServCounterRemoteRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of non recursive queries which originated
      outside the local zone which have been referred to
      other servers."
     ::= { servCounter 39 }

dnsServCounterRemoteErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries which originated outside the local
      zone which were answered with error responses."
     ::= { servCounter 40 }

dnsServCounterReqRef OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of DNS requests refused by the local server
      process."
     ::= { servCounter 41 }

dnsServCounterReqUnparse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests received which were unparseable."
     ::= { servCounter 42 }

 OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were aborted for other
      (local) server errors."
     ::= { servCounter 43 }

dnsServCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of duplicate requests which were received and
      not forwarded."
     ::= { servCounter 44 }
```

```
dnsServCounterReforward OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were received from other
      servers redirecting query to another server."
     ::= { servCounter 45 }

dnsServCounterTruncLocal OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses due to local truncation
      for local queries."
     ::= { servCounter 46 }

dnsServCounterLocalRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests received for relative names from
      the local host."
     ::= { servCounter 47 }

dnsServCounterRelInZone OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests received for relative names from
      other hosts within the servers zone."
     ::= { servCounter 48 }

dnsServCounterRelRemote OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests received for relative names form
      hosts outside of the server zone."
     ::= { servCounter 49 }

-- DNS Server Counter Table

dnsCounterTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Counter information based on DNS record type."
    ::= { servCounter 50 }

dnsServCounterEntry OBJECT-TYPE
     SYNTAX  dnsServCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains count information for each DNS
      record type and zone known to the server."
    INDEX  { dnsServCounterIndex }
    ::= { dnsServCounterTable 1 }
```

```
DnsServCounterEntry ::=
    SEQUENCE {
        dnsServCounterIndex
            INTEGER,
        dnsServCounterClass,
            INTEGER,
        dnsServCounterQueryType
            INTEGER,
        dnsServCounterQuerySource
            INTEGER,
        dnsServCounterTransport
            INTEGER,
        dnsServCounterNumber
            INTEGER
        }

dnsServCounterIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value for each entry in the
      dnsServCounter Table."
     ::= { dnsServCounterEntry 1 }

dnsServCounterClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the record type reported in this row of
      the table.  For classes defined in RFC 1035, the values
      are the same.  ::= { dnsServCounterEntry 2 }
      dnsServCounterQueryType OBJECT-TYPE SYNTAX INTEGER
      ACCESS read-only STATUS mandatory DESCRIPTION "The type
      of record being reported for this row in the table.
      For types defined in RFC 1035, the values are the same
      as those specified in the RFC."
     ::= { dnsServCounterEntry 3 }

dnsServCounterQuerySource OBJECT-TYPE
     SYNTAX  INTEGER {
         local (1),
         remote (2)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A value of 1 indicates that the queries reported on
      this row came from within the servers zone.  A value of
      2 indicates the queries are from outside of the servers
      zone."
     ::= { dnsServCounterEntry 4 }

dnsServCounterTransport OBJECT-TYPE
     SYNTAX  INTEGER {
        udp (1),
        tcp (2)
     }
     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."
     ::= { dnsServCounterEntry 5 }
```

```
dnsServCounterNumber OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that have been recorded which fit
      the 'query profile described by this row of the table."
     ::= { dnsServCounterEntry 6 }

-- Full Service Resolver Counter Group

-- The implementation of the Full Service Resolver Counter Group is
-- mandatory only for systems which offer full service resolver
-- functions.

dnsFSResolvCounterTCPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of attempts over TCP made by the full service
      resolver to connect with root servers."
     ::= { fsResolvCounter 1 }

dnsFSResolvCounterUDPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of out bound UDP packets sent by the full
      service resolver to root servers."
     ::= { fsResolvCounter 2 }

dnsFSResolvCounterForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that have been forwarded because
      they could not be answered with local information by
      the resolver."
     ::= { fsResolvCounter 3 }

dnsFSResolvCounterReforward OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were received from other
      servers redirecting query to another server."
     ::= { fsResolvCounter 4 }

dnsFSResolvCounterRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number requests retransmitted for all reasons."
     ::= { fsResolvCounter 5 }
```

```
dnsFSCounterNoResponse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted because of
      no response."
     ::= { fsResolvCounter 6 }

dnsFSCounterRootRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted that were to
      root servers."
     ::= { fsResolvCounter 7 }

dnsFSCounterLateRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted as a result
      of late responses."
     ::= { fsResolvCounter 8 }

dnsFSCounterExpiredRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted as a result
      of computed timers expiring."
     ::= { fsResolvCounter 9 }

dnsFSCounterRetransNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted as a result
      of the expiration of static timers (no previous data)."
     ::= { fsResolvCounter 10 }

dnsFSCounterRetransSmall OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmissions where an adjusted value was
      used instead of a computed value because the computed
      value was implausibly to small."
     ::= { fsResolvCounter 11 }

dnsFSCounterRetransLarge OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmissions where an adjusted value was
      used instead of a computed value because the computed
      value was implausibly to large."
     ::= { fsResolvCounter 12 }
```

```
dnsFSCounterComputed OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries based on computed
      estimates."
     ::= { fsResolvCounter 13 }

dnsFSCounterStaticNoPrev OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries based on static
      estimates - with no previous data."
     ::= { fsResolvCounter 14 }

dnsFSCounterEstSmall OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries based on static
      estimates - with computed value too small."
     ::= { fsResolvCounter 15 }

dnsFSCounterEstLarge OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries based on static
      estimates - with computed value too large."
     ::= { fsResolvCounter 16 }

dnsFSCounterEstFor OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries made based on static
      estimates that used forwarders."
     ::= { fsResolvCounter 17 }

dnsFSCounterEstPrim OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of retransmitted queries made based on static
      estimates - priming slowly."
     ::= { fsResolvCounter 18 }

dnsFSCounterLocalForward OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses due to local truncation
      for forwarded queries."
     ::= { fsResolvCounter 19 }
```

```
dnsFSCounterTruncRemote OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses from remote servers for
      forwarded queries."
     ::= { fsResolvCounter 20 }

dnsFSCounterTruncIntermed OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses from remote servers for
      queries with intermediate results."
     ::= { fsResolvCounter 21 }

dnsFSCounterInternal OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests internally generated by the
      server."
     ::= { fsResolvCounter 22 }

dnsFSCounterInternalTimeOut OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests internally generated which timed
      out."
     ::= { fsResolvCounter 23 }

-- Recursive Server Counter Group

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

dnsRSCounterTCPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of attempts over TCP made by the recursive
      server to connect with root servers."
     ::= { rsCounter 1 }

dnsRSCounterUDPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of out bound UDP packets sent by the recursive
      server to root servers."
     ::= { rsCounter 2 }
```

```
dnsRSCounterInvReqsSent OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of inverse (IQUERY) queries that have been sent
      by the server."
     ::= { rsCounter 3 }

dnsRSCounterForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that have been forwarded because
      they could not be answered with local information by
      the server."
     ::= { rsCounter 4 }

dnsRSCounterForwardAuth OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and for which
      authoritative responses were made."
     ::= { rsCounter 5 }

dnsRSCounterForwardNoName OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and for which
      authoritative no such name (NXDOMAIN) responses were
      made."
     ::= { rsCounter 6 }

dnsRSCounterForwardNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and for which
      an authoritative no such data (empty answer) response
      was made.  This means that the name exists, but there
      are not resource records of the type requested for that
      name (response code in the response header is OK but
      the number of records in the answer section is zero)."
     ::= { rsCounter 7 }

dnsRSCounterForwardCached OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and for which
      non- authoritative responses were made.  That is, the
      request was forwarded to a server which was only
      authoritative for a parent domain of the name in
      question, but that server had the answer in its cache."
     ::= { rsCounter 8 }
```

```
dnsRSCounterForwardCacheNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and for which
      the answer is non-authoritative and is not a referral,
      but the answer section is empty anyway.  These events
      should not occur."
     ::= { rsCounter 9 }

dnsRSCounterForwardCAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of C_ANY requests which were forwarded."
     ::= { rsCounter 10 }

dnsRSCounterAbortRemote OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and were
      aborted with no (good) responses from the remote
      server."
     ::= { rsCounter 11 }

dnsRSCounterAbortOther OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests which were forwarded and were
      aborted due to local server errors."
     ::= { rsCounter 12 }

dnsRSCounterRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number requests retransmitted for all reasons by the
      server."
     ::= { rsCounter 13 }

dnsRSCounterNoResponse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted because of
      no response by the server."
     ::= { rsCounter 14 }

dnsRSCounterRootRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted by the
      server that were to root servers."
     ::= { rsCounter 15 }
```

```
dnsRSCounterLateRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted by the
      server as a result of late responses."
     ::= { rsCounter 16 }

dnsRSCounterExpiredRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted as a result
      of computed timers expiring in the server."
     ::= { rsCounter 17 }

dnsRSCounterRetransNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of queries that were retransmitted as a result
      of the expiration of static timers (no previous data)."
     ::= { rsCounter 18 }

dnsRSCounterRetransSmall OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions where an adjusted
      value was used instead of a computed value because the
      computed value was implausibly to small."
     ::= { rsCounter 19 }

dnsRSCounterRetransLarge OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions where an adjusted
      value was used instead of a computed value because the
      computed value was implausibly to large."
     ::= { rsCounter 20 }

dnsRSCounterComputed OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on computed
      estimates by the server."
     ::= { rsCounter 21 }

dnsRSCounterStaticNoPrev OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on static
      estimates - with no previous data."
     ::= { rsCounter 22 }
```

```
dnsRSCounterEstSmall OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on static
      estimates - with computed value too small."
     ::= { rsCounter 23 }

dnsRSCounterEstLarge OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on static
      estimates - with computed value too large."
     ::= { rsCounter 24 }

dnsRSCounterEstFor OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on static
      estimates that used forwarders."
     ::= { rsCounter 25 }

dnsRSCounterEstPrim OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of server retransmissions based on static
      estimates - priming slowly."
     ::= { rsCounter 26 }

dnsRSCounterLocalForward OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses due to local server
      truncation for forwarded queries."
     ::= { rsCounter 27 }

dnsRSCounterTruncRemote OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses from remote servers for
      forwarded queries from this server."
     ::= { rsCounter 28 }

dnsRSCounterTruncIntermed OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of truncated responses from remote servers for
      queries with intermediate results."
     ::= { rsCounter 29 }
```

```
dnsRSCounterMartians OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were received from
      servers that the server does not think it asked."
     ::= { rsCounter 30 }

dnsRSCounterResponsesRecd OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received to all queries from the
      recursive server."
     ::= { rsCounter 31 }

dnsRSCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were duplicates and
      ignored by the recursive server."
     ::= { rsCounter 31 }

dnsRSCounterUnparse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were unparseable."
     ::= { rsCounter 32 }

dnsRSCounterServerFail OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received which were server failed
      responses."
     ::= { rsCounter 33 }

dnsRSCounterOtherFail OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of responses received by the recursive server
      which where for all other bad responses."
     ::= { rsCounter 34 }

dnsRSCounterInternal OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests internally generated by the
      server."
     ::= { rsCounter 35 }
```

```
dnsRSCounterInternalTimeOut OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of requests internally generated which timed
      out."
     ::= { rsCounter 36 }


-- Non Recursive Server Counter Group

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

dnsNRCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "Number of duplicate requests which were received and
      not forwarded.  The value will be 0 if the number of
      duplicate requests received and not forwarded is not
      known."
     ::= { nrCounter 1 }

-- A Records Group

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

-- A Record Default Table

-- The A record default table contains information about A records
-- that the DNS software reads at initialization.

dnsARecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsARecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all A records for the
      DNS software."
    ::= { aRecord 1 }

dnsARecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsARecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains A record configuration
      information."
     INDEX { dnsARecDefaultIndex }
     ::= { dnsARecDefaultTable 1 }
```

```
DnsARecDefaultEntry ::=
    SEQUENCE {
        dnsARecDefaultIndex
            INTEGER,
 dnsARecDefaultZone
            OCTET STRING,
        dnsARecDefaultClass
     INTEGER,
        dnsARecDefaultTTL
            INTEGER,
        dnsARecDefaultName
     OCTET STRING,
        dnsARecDefaultAddr
            IpAddress
        }

dnsARecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsARecDefaultEntry 1 }

dnsARecDefaultZone OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The zone to which this entry belongs."
     ::= { dnsARecDefaultEntry 2 }

dnsARecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the A record contained in this row For
      classes defined in RFC 1035, the values are the same."
     ::= { dnsARecDefaultEntry 3 }

dnsARecDefaultTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The default Time to Live value for this A record which
      is read in on initialization."
     ::= { dnsARecDefaultEntry 4 }

dnsARecDefaultName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name of the host to which the entry applies."
     ::= { dnsARecDefaultEntry 5 }

dnsARecDefaultAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The IP Address of the entry which corresponds to the
      dnsARecDefaultName."
     ::= { dnsARecDefaultEntry 6 }
```

```
-- A Record Cache Table

-- The A record cache table contains information the DNS software
-- has about currently known A records.  This information may be
-- different than that which is found in the default table.

dnsARecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsARecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all A records for the DNS
      software."
    ::= { aRecord 2 }

dnsARecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsARecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains A record cache information."
    INDEX  { dnsARecCacheIndex }
    ::= { dnsARecCacheTable 1 }

DnsARecCacheEntry ::=
    SEQUENCE {
        dnsARecCacheIndex
            INTEGER,
        dnsARecCacheZone
            OCTET STRING,
        dnsARecCacheClass
            INTEGER,
        dnsARecCacheName
            OCTET STRING,
        dnsARecCacheAddr
            IpAddress,
        dnsARecCacheSource
            OCTET STRING,
        dnsARecCacheTTL
            INTEGER
        }

dnsARecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsARecCacheEntry 1 }

dnsARecCacheZone OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The zone to which this cache entry belongs. A NULL
      value indicates that the server does not have the name
      of the zone as in the case of a boundary between a host
      and zone name."
     ::= { dnsARecCacheEntry 2 }
```

```
dnsARecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the A record contained in this cache
      entry.  The value is the same as that defined in RFC
      1035 for objects that are defined in that document"
     ::= { dnsARecCacheEntry 3 }

dnsARecCacheName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The fully qualified domain name of the host to which
      this cache entry applies."
     ::= { dnsARecCacheEntry 4 }

dnsARecCacheAddr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The protocol address of the entry which corresponds to
      the dnsARecCacheName."
     ::= { dnsARecCacheEntry 5 }

dnsARecCacheSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The IP address of the server from which this cache
      entry came."
     ::= { dnsARecCacheEntry 6 }

dnsARecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this entry."
     ::= { dnsARecCacheEntry 7 }


-- NS Records Group

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

-- NS Record Default Table

-- The NS record default table contains information about NS
-- records that the DNS software reads at initialization.

dnsNSRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsNSRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all NS records for the
      DNS software."
    ::= { nsRecord 1 }
```

```
dnsNSRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsNSRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains NS record configuration
      information."
    INDEX  { dnsNSRecDefaultIndex }
    ::= { dnsNSRecDefaultTable 1 }

DnsNSRecDefaultEntry ::=
    SEQUENCE {
        dnsNSRecDefaultIndex
            INTEGER,
        dnsNSRecDefaultZone
            OCTET STRING,
        dnsNSRecDefaultClass
            INTEGER,
        dnsNSRecDefaultName
            OCTET STRING
        }

dnsNSRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsNSRecDefaultEntry 1 }

dnsNSRecDefaultZone OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The zone to which this NS entry belongs."
     ::= { dnsNSRecDefaultEntry 2 }

dnsNSRecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the NS record contained in this row.  The
      primary purpose for this object is to allow room for
      future expansion.  For those objects in this row whose
      class is defined in RFC 1035, the value is the same."
     ::= { dnsNSRecDefaultEntry 3 }

dnsNSRecDefaultName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The fully qualified name in the form of
      labserver.abc.edu. for example."
     ::= { dnsNSRecDefaultEntry 4 }

-- NS Record Cache Table

-- The NS record cache table contains information the DNS software
-- has about currently known NS records.  This information may be
-- different than that which is found in the default table.
```

```
dnsNSRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsNSRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all NS records for the DNS
      software."
    ::= { nsRecord 2 }

dnsNSRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsNSRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains NS record cache information."
    INDEX  { dnsNSRecCacheIndex }
    ::= { dnsNSRecCacheTable 1 }

DnsNSRecCacheEntry ::=
    SEQUENCE {
        dnsNSRecCacheIndex
            INTEGER,
        dnsNSRecCacheZone
            OCTET STRING,
        dnsNSRecCacheClass
            INTEGER,
        dnsNSRecCacheName
            OCTET STRING,
        dnsNSRecCacheSource
            OCTET STRING,
        dnsNSRecCacheTTL
            INTEGER
        }

dnsNSRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsNSRecCacheEntry 1 }

dnsNSRecCacheZone OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The zone that the server identified by the NS record
      provides DNS service for which is shown in this entry."
     ::= { dnsNSRecCacheEntry 2 }

dnsNSRecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the NS record contained in this cache
      entry.  This is provided to allow for the addition of
      new class NS records.  Integer values consistent for
      class types defined in RFC 1035 are used for objects
      defined in that document."
     ::= { dnsNSRecCacheEntry 3 }
```

```
dnsNSRecCacheName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name of the server to which this cache entry
      applies."
     ::= { dnsNSRecCacheEntry 4 }

dnsNSRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsNSRecCacheEntry 5 }

dnsNSRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this entry."
     ::= { dnsNSRecCacheEntry 6 }


-- CNAME Records Group

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

-- CNAME Record Default Table

-- The CNAME record default table contains information about CNAME
-- records that the DNS software reads at initialization.

dnsCnameRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsCnameRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all CNAME records for
      the DNS software."
    ::= { cnameRecord 1 }

dnsCnameRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsCnameRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains CNAME record configuration
      information."
    INDEX  { dnsCnameRecDefaultIndex }
    ::= { dnsCnameRecDefaultTable 1 }
```

```
DnsCnameRecDefaultEntry ::=
    SEQUENCE {
        dnsCnameRecDefaultIndex
            INTEGER,
        dnsCnameRecDefaultDomaine
            OCTET STRING,
        dnsCnameRecDefaultClass
            INTEGER,
        dnsCnameRecDefaultName
            OCTET STRING,
        dnsCnameDefaultCname
            OCTET STRING
        }

dnsCnameRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsCnameRecDefaultEntry 1 }

dnsCnameRecDefaultDomaine OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The domaine to which this CNAME entry belongs."
     ::= { dnsCnameRecDefaultEntry 2 }

dnsCnameRecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the CNAME record contained in this row.
      The primary purpose for this object is to allow room
      for future expansion.  See RFC 1035 for definitions of
      existing class values."
     ::= { dnsCnameRecDefaultEntry 3 }

dnsCnameRecDefaultName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name which corresponds to the CNAME entry which is
      contained in dnsCnameRecDefaultCname."
     ::= { dnsCnameRecDefaultEntry 4 }

dnsCnameRecDefaultCname OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A canonical name for the system identified by this
      entry."
     ::= { dnsCnameRecDefaultEntry 5 }

-- CNAME Record Cache Table

-- The CNAME record cache table contains information the DNS
-- software has about currently known CNAME records.  This
-- information may be different than that which is found in the
-- default table.
```

```
dnsCnameRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsCnameRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all CNAME records for the DNS
      software."
    ::= { cnameRecord 2 }

dnsCnameRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsCnameRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains CNAME record cache information."
    INDEX  { dnsCnameRecCacheIndex }
    ::= { dnsCnameRecCacheTable 1 }

DnsCnameRecCacheEntry ::=
    SEQUENCE {
        dnsCnameRecCacheIndex
            INTEGER,
        dnsCnameRecCacheDomaine
            OCTET STRING,
        dnsCnameRecCacheClass
            INTEGER,
        dnsCnameRecCacheData
            OCTET STRING,
        dnsCnameRecCacheSource
            OCTET STRING,
        dnsCnameRecCacheTTL
            INTEGER
        }

dnsCnameRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsCnameRecCacheEntry 1 }

dnsCnameRecCacheDomaine OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The domaine to which the host identified by the CNAME
      record belongs."
     ::= { dnsCnameRecCacheEntry 2 }

dnsCnameRecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the CNAME record contained in this cache
      entry.  This is provided to allow for the addition of
      new class CNAME records.  The value returned will be
      consistent with the the values specified in RFC 1035
      for those objects whose class has been defined in that
      document."
     ::= { dnsCnameRecCacheEntry 3 }
```

```
dnsCnameRecCacheName OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name of the system to which this cache entry
      applies."
     ::= { dnsCnameRecCacheEntry 4 }

dnsCnameRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsCnameRecCacheEntry 5 }

dnsCnameRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this entry."
     ::= { dnsCnameRecCacheEntry 6 }


-- WKS Records Group

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

-- WKS Record Default Table

-- The WKS record default table contains information about WKS
-- records that the DNS software reads at initialization.

dnsWksRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsWksRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all WKS records for
      the DNS software."
    ::= { wksRecord 1 }

dnsWksRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsWksRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains WKS record configuration
      information."
    INDEX  { dnsWksRecDefaultIndex }
    ::= { dnsWksRecDefaultTable 1 }

DnsWksRecDefaultEntry ::=
    SEQUENCE {
        dnsWksRecDefaultIndex
            INTEGER,
        dnsWksRecDefaultAddr
            IpAddress,
        dnsWksRecDefaultProto
            INTEGER,
        dnsWksRecDefaultBitMap
            OCTET STRING
        }
```

```
dnsWksRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsWksRecDefaultEntry 1 }

dnsWksRecDefaultAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The 32 bit Internet address of the entity described by
      this entry."
     ::= { dnsWksRecDefaultEntry 2 }

dnsWksRecDefaultProto OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This describes the specific IP protocol whose services
      are listed in dnsWksRecDefaultBitMap.  See RFC 1035,
      page 21 for additional information."
     ::= { dnsWksRecDefaultEntry 3 }

dnsWksRecDefaultBitMap OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is a bit map of the services provided for the IP
      protocol shown in dnsWksRecDefaultProto.  RFC 1035
      explains that the bit map has one bit per port of the
      specified protocol.  The first bit corresponds to port
      0, the second to port 1, etc.  If the bit map does not
      include a bit for a protocol of interest, that bit is
      assumed zero."
     ::= { dnsWksRecDefaultEntry 4 }

-- WKS Record Cache Table

-- The WKS record cache table contains information the DNS software
-- has about currently known WKS records.  This information may be
-- different than that which is found in the default table.

dnsWksRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsWksRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all WKS records for the DNS
      software."
    ::= { wksRecord 2 }

dnsWksRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsWksRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains WKS record cache information."
    INDEX  { dnsWksRecCacheIndex }
    ::= { dnsWksRecCacheTable 1 }
```

```
DnsWksRecCacheEntry ::=
    SEQUENCE {
        dnsWksRecCacheIndex
            INTEGER,
        dnsWksRecCacheAddr
            IpAddress,
        dnsWksRecCacheProto
            INTEGER,
        dnsWksRecCacheBitMap
            OCTET STRING,
        dnsWksRecCacheSource
            OCTET STRING,
        dnsWksRecCacheTTL
            INTEGER
        }

dnsWksRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsWksRecCacheEntry 1 }

dnsWksRecCacheAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The Internet address in cache of the entity described
      in this row of the table."
     ::= { dnsWksRecCacheEntry 2 }

dnsWksRecCacheProto OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This describes the specific IP protocol whose services
      are listed in dnsWksRecCacheBitMap."
     ::= { dnsWksRecCacheEntry 3 }

dnsWksRecCacheBitMap OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is a bit map of the services provided for the IP
      protocol shown in dnsWksRecCacheProto."
     ::= { dnsWksRecCacheEntry 4 }

dnsWksRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information about this WKS RR."
     ::= { dnsWksRecCacheEntry 5 }
```

```
dnsWksRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this WKS cache entry."
     ::= { dnsWksRecCacheEntry 6 }

-- PTR Records Group

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

-- PTR Record Default Table

-- The PTR record default table contains information about PTR
-- records that the DNS software reads at initialization.

dnsPtrRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsPtrRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all PTR records for
      the DNS software."
    ::= { ptrRecord 1 }

dnsPtrRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsPtrRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "This table contains PTR record configuration information."
    INDEX  { dnsPtrRecDefaultIndex }
    ::= { dnsPtrRecDefaultTable 1 }

DnsPtrRecDefaultEntry ::=
    SEQUENCE {
        dnsPtrRecDefaultIndex
            INTEGER,
        dnsPtrRecDefaultPtr
            OCTET STRING,
        dnsPtrRecDefaultClass
            INTEGER,
        dnsPtrRecDefaultData
            OCTET STRING
        }

dnsPtrRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsPtrRecDefaultEntry 1 }
```

```
dnsPtrRecDefaultPtr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The pointer to the item identified in
      dnsPtrRecDefaultData.  In the case of in-addr.arpa type
      pointer records, an example would be
      1.0.1.16.IN-ADDR.ARPA. for the host decwrl.pa.dec.com.
      The 'value' decwrl.pa.dec.com would be contained in
      dnsPtrRecDefaultData."
     ::= { dnsPtrRecDefaultEntry 2 }

dnsPtrRecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the PTR record contained in this row.
      The primary purpose for this object is to allow room
      for future expansion.  See RFC 1035 for definitions of
      existing class values."
     ::= { dnsPtrRecDefaultEntry 3 }

dnsPtrRecDefaultData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The data which corresponds to the PTR entry which is
      contained in dnsPtrRecDefaultPtr."
     ::= { dnsPtrRecDefaultEntry 4 }

-- PTR Record Cache Table

-- The PTR record cache table contains information the DNS software
-- has about currently known PTR records.  This information may be
-- different than that which is found in the default table.

dnsPtrRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsPtrRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all PTR records for the DNS
      software."
    ::= { ptrRecord 2 }

dnsPtrRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsPtrRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains PTR record cache information."
    INDEX  { dnsPtrRecCacheIndex }
    ::= { dnsPtrRecCacheTable 1 }
```

```
DnsPtrRecCacheEntry ::=
    SEQUENCE {
        dnsPtrRecCacheIndex
            INTEGER,
        dnsPtrRecCachePtr
            OCTET STRING,
        dnsPtrRecCacheClass
            INTEGER,
        dnsPtrRecCacheData
            OCTET STRING,
        dnsPtrRecCacheSource
            OCTET STRING,
        dnsPtrRecCacheTTL
            INTEGER
        }

dnsPtrRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsPtrRecCacheEntry 1 }

dnsPtrRecCachePtr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The pointer to the item identified in
      dnsPtrRecCacheData.  In the case of in-addr.arpa type
      pointer records, an example would be
      1.0.1.16.IN-ADDR.ARPA. for the host decwrl.pa.dec.com.
      The 'value' decwrl.pa.dec.com would be contained in
      dnsPtrRecCacheData."
     ::= { dnsPtrRecCacheEntry 2 }

dnsPtrRecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the PTR record contained in this cache
      entry.  This is provided to allow for the addition of
      new class PTR records.  The value returned will be
      consistent with the the values specified in RFC 1035
      for those objects whose class has been defined in that
      document."
     ::= { dnsPtrRecCacheEntry 3 }

dnsPtrRecCacheData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The data which corresponds to the PTR entry which is
      contained in dnsPtrRecCachePtr."
     ::= { dnsPtrRecCacheEntry 4 }
```

```
dnsPtrRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsPtrRecCacheEntry 5 }

dnsPtrRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this cache entry."
     ::= { dnsPtrRecCacheEntry 6 }

-- HINFO Records Group

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

-- HINFO Record Default Table

-- The HINFO record default table contains information about HINFO
-- records that the DNS software reads at initialization.

dnsHinfoRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsHinfoRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all HINFO records for
      the DNS software."
    ::= { hinfoRecord 1 }

dnsHinfoRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsHinfoRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains HINFO record configuration
      information."
    INDEX  { dnsHinfoRecDefaultIndex }
    ::= { dnsHinfoRecDefaultTable 1 }

DnsHinfoRecDefaultEntry ::=
    SEQUENCE {
        dnsHinfoRecDefaultIndex
            INTEGER,
        dnsHinfoRecDefaultDomaine
            OCTET STRING,
        dnsHinfoRecDefaultCpu
            OCTET STRING,
        dnsHinfoRecDefaultOs
            OCTET STRING
        }

dnsHinfoRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsHinfoRecDefaultEntry 1 }
```

```
dnsHinfoRecDefaultDomaine OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The fully qualified domaine name of the system who's
      default host information is detailed by this record."
     ::= { dnsHinfoRecDefaultEntry 2 }

dnsHinfoRecDefaultCpu OBJECT-TYPE
     SYNTAX  OCTET STRING ( SIZE (0..40) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The CPU type of the system identified by the domain
      name in dnsHinfoRecDefaultDomaine.  Existing types have
      been detailed in RFC 1010."
     ::= { dnsHinfoRecDefaultEntry 3 }

dnsHinfoRecDefaultOs OBJECT-TYPE
     SYNTAX  OCTET STRING ( SIZE (0..40) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The operating system in use on the system identified
      in dnsHinfoRecDefaultDomaine.  Existing types have been
      detailed in RFC 1010."
     ::= { dnsHinfoRecDefaultEntry 4 }

-- HINFO Record Cache Table

-- The HINFO record cache table contains information the DNS
-- software has about currently known HINFO records.  This
-- information may be different than that which is found in the
-- default table.

dnsHinfoRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsHinfoRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all HINFO records for the DNS
      software."
    ::= { hinfoRecord 2 }

dnsHinfoRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsHinfoRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains HINFO record cache information."
    INDEX  { dnsHinfoRecCacheIndex }
    ::= { dnsHinfoRecCacheTable 1 }
```

```
DnsHinfoRecCacheEntry ::=
    SEQUENCE {
        dnsHinfoRecCacheIndex
            INTEGER,
        dnsHinfoRecCacheDomaine
            OCTET STRING,
        dnsHinfoRecCacheCpu
            INTEGER,
        dnsHinfoRecCacheOs
            OCTET STRING,
        dnsHinfoRecCacheSource
            OCTET STRING,
        dnsHinfoRecCacheTTL
            INTEGER
        }

dnsHinfoRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsHinfoRecCacheEntry 1 }

dnsHinfoRecCacheDomaine OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The fully qualified domaine name of the system who's
      cache host information is detailed by this record."
     ::= { dnsHinfoRecCacheEntry 2 }

dnsHinfoRecCacheCpu OBJECT-TYPE
     SYNTAX  OCTET STRING ( SIZE (0..40) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The CPU type of the system identified by the domain
      name in dnsHinfoRecCacheDomaine.  Existing types have
      been detailed in RFC 1010."
     ::= { dnsHinfoRecCacheEntry 3 }

dnsHinfoRecCacheOs OBJECT-TYPE
     SYNTAX  OCTET STRING ( SIZE (0..40) )
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The operating system in use on the system identified
      in dnsHinfoRecCacheDomaine.  Existing types have been
      detailed in RFC 1010."
     ::= { dnsHinfoRecCacheEntry 4 }

dnsHinfoRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsHinfoRecCacheEntry 5 }
```

```
dnsHinfoRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this cache entry."
     ::= { dnsHinfoRecCacheEntry 6 }

-- MX Records Group

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

-- MX Record Default Table

-- The MX record default table contains information about MX
-- records that the DNS software reads at initialization.

dnsMxRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsMxRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Configuration information about all MX records for the
      DNS software."
    ::= { mxRecord 1 }

dnsMxRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsMxRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains MX record configuration
      information."
    INDEX  { dnsMxRecDefaultIndex }
    ::= { dnsMxRecDefaultTable 1 }

DnsMxRecDefaultEntry ::=
    SEQUENCE {
        dnsMxRecDefaultIndex
            INTEGER,
        dnsMxRecDefaultClass
            INTEGER,
        dnsMxRecDefaultOwner
            OCTET STRING,
        dnsMxRecDefaultPref
            INTEGER,
        dnsMxRecDefaultExch
            OCTET STRING
        }

dnsMxRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsMxRecDefaultEntry 1 }
```

```
dnsMxRecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the MX record contained in this default
      entry.  This is provided to allow for the addition of
      new class MX records.  The value returned will be
      consistent with the the values specified in RFC 1035
      for those objects whose class has been defined in that
      document."
     ::= { dnsMxRecDefaultEntry 2 }

dnsMxRecDefaultOwner OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The host name for whom the MX record on this row of
      the table applies."
     ::= { dnsMxRecDefaultEntry 3 }

dnsMxRecDefaultPref OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the 16 bit integer for the preference given to
      this RR entry for the host specified in
      dnsMxRecDefaultOwner."
     ::= { dnsMxRecDefaultEntry 4 }

dnsMxRecDefaultExch OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the fully qualified domain name of the host
      which is acting as a mail exchange for the
      dnsMxRecDefaultOwner."
     ::= { dnsMxRecDefaultEntry 5 }


-- MX Record Cache Table

-- The MX record cache table contains information the DNS software
-- has about currently known MX records.  This information may be
-- different than that which is found in the default table.

dnsMxRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsMxRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all MX records for the DNS
      software."
    ::= { mxRecord 2 }

dnsMxRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsMxRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains MX record cache information."
    INDEX  { dnsMxRecCacheIndex }
    ::= { dnsMxRecCacheTable 1 }
```

```
DnsMxRecCacheEntry ::=
    SEQUENCE {
        dnsMxRecCacheIndex
            INTEGER,
        dnsMxRecCacheClass
            INTEGER,
        dnsMxRecCacheOwner
            OCTET STRING,
        dnsMxRecCachePref
            INTEGER,
        dnsMxRecCacheExch
            OCTET STRING,
        dnsMxRecCacheSource
            OCTET STRING,
        dnsMxRecCacheTTL
            INTEGER
        }

dnsMxRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsMxRecCacheEntry 1 }

dnsMxRecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the MX record contained in this cache
      entry.  This is provided to allow for the addition of
      new class MX records.  The value returned will be
      consistent with the the values specified in RFC 1035
      for those objects whose class has been defined in that
      document."
     ::= { dnsMxRecCacheEntry 2 }

dnsMxRecCacheOwner OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The host name for whom the MX record on this row of
      the cache table applies."
     ::= { dnsMxRecCacheEntry 3 }

dnsMxRecCachePref OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the 16 bit integer for the preference given to
      this RR cache entry for the host specified in
      dnsMxRecCacheOwner.  ::= { dnsMxRecCacheEntry 4 }
      dnsMxRecDefaultExch OBJECT-TYPE SYNTAX OCTET STRING
      ACCESS read-only STATUS mandatory DESCRIPTION "This is
      the fully qualified domain name of the host which is
      acting as a mail exchange for the dnsMxRecCacheOwner."
     ::= { dnsMxRecCacheEntry 5 }
```

```
dnsMxRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsMxRecCacheEntry 6 }

dnsMxRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this cache entry."
     ::= { dnsMxRecCacheEntry 7 }


-- Undefined Records Group

-- The implementation of the Undefined Records Group is optional
-- for all systems which implement resource records with either
-- type or class definitions not found in RFC 1035.

-- Undefined Record Default Table

-- The Undefined record default table contains information about
-- records that are not defined in RFC 1035 which the DNS software
-- reads at initialization.

dnsUndefRecDefaultTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsUndefRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Default information about all undefined records for
      the DNS software."
    ::= { undefRecord 1 }

dnsUndefRecDefaultEntry  OBJECT-TYPE
     SYNTAX  dnsUndefRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains default undefined record
      configuration information."
    INDEX  { dnsUndefRecDefaultIndex }
    ::= { dnsUndefRecDefaultTable 1 }

DnsUndefRecDefaultEntry ::=
    SEQUENCE {
        dnsUndefRecDefaultIndex
            INTEGER,
        dnsUndefRecDefaultAddr
            OCTET STRING,
        dnsUndefRecDefaultClass
            INTEGER,
        dnsUndefRecDefaultName
            OCTET STRING
        }
```

```
dnsUndefRecDefaultIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsUndefRecDefaultEntry 1 }

dnsUndefRecDefaultAddr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The protocol address which belong to the host
      identified by this entry."
     ::= { dnsUndefRecDefaultEntry 2 }

dnsUndefRecDefaultClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the undefined record contained in this
      row."
     ::= { dnsUndefRecDefaultEntry 3 }

dnsUndefRecDefaultData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The undefined record data."
     ::= { dnsUndefRecDefaultEntry 4 }

-- Undefined Record Cache Table

-- The undefined record cache table contains information the DNS
-- software has about currently known undefined records.  This
-- information may be different than that which is found in the
-- default table.

dnsUndefRecCacheTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsUndefRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Cache information about all undefined records for the
      DNS software."
    ::= { undefRecord 2 }

dnsUndefRecCacheEntry  OBJECT-TYPE
     SYNTAX  dnsUndefRecCacheEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains undefined record cache
      information."
    INDEX  { dnsUndefRecCacheIndex }
    ::= { dnsUndefRecCacheTable 1 }
```

```
DnsUndefRecCacheEntry ::=
    SEQUENCE {
        dnsUndefRecCacheIndex
            INTEGER,
        dnsUndefRecCacheAddr
            OCTET STRING,
        dnsUndefRecCacheClass
            INTEGER,
        dnsUndefRecCacheData
            OCTET STRING,
        dnsUndefRecCacheSource
            OCTET STRING,
        dnsUndefRecCacheTTL
            INTEGER
        }

dnsUndefRecCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value which identifies each entry in
      this table."
     ::= { dnsUndefRecCacheEntry 1 }

dnsUndefRecCacheAddr OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The protocol address which belong to the host
      identified by this entry ."
     ::= { dnsUndefRecCacheEntry 2 }

dnsUndefRecCacheClass OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The class of the undefined record contained in this
      cache entry."
     ::= { dnsUndefRecCacheEntry 3 }

dnsUndefRecCacheData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The undefined record data in cache."
     ::= { dnsUndefRecCacheEntry 4 }

dnsUndefRecCacheSource OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The source of this cache entry.  In most cases this
      will be the host name of the server which provided the
      information."
     ::= { dnsUndefRecCacheEntry 5 }
```

```
dnsUndefRecCacheTTL OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the Time To Live value currently in effect for
      this entry."
     ::= { dnsUndefRecCacheEntry 6 }

-- 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 from disk 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 - No action."
     ::= { serverMgmt 1 }

dnsServMgmtCacheData OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "The location from which the initial cache load is
      made."
    ::= { serverMgmt 2 }

-- DNS Server Boot Configuration Table

dnsServMgmtBootTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsServMgmtBootEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "Information default zones served by this name server
      process."
    ::= { serverMgmt 3 }

dnsServMgmtBootEntry OBJECT-TYPE
     SYNTAX  dnsServMgmtBootEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "This table contains default boot information about the
      zones served which is read in from disk at startup."
    INDEX  { dnsServMgmtIndex }
    ::= { dnsServMgmtBootTable 1 }
```

```
DnsServMgmtBootEntry ::=
    SEQUENCE {
        dnsServMgmtIndex
            INTEGER,
        dnsServMgmtBootDomain
            DisplayString,
        dnsServMgmtBootServerType
            INTEGER,
        dnsServMgmtBootLoad
            DisplayString,
        dnsServMgmtBootDate
            dnsDate,
        dnsServMgmtBootTransfer
            DisplayString,
        dnsServMgmtBootTransferDirection
            INTEGER,
        dnsServMgmtBootTransferDate
            DnsDate,
        dnsServMgmtAdministrator
            INTEGER,
        dnsServMgmtSerial
            INTEGER,
        dnsServMgmtRefresh
            INTEGER,
        dnsServMgmtRetry
            INTEGER,
        dnsServMgmtExpire
            INTEGER,
        dnsServMgmtMinimum
            INTEGER
     }
dnsServMgmtIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "A unique index value for each zone entry in the Server
      Management BootTable."
    ::= { dnsServMgmtBootEntry 1 }

dnsServMgmtBootDomain OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the domain for which this entry information
      applies.  In the case of the in-addr.arpa domains, they
      will be listed with the network address, for example:
      0.121.16.in-addr.arpa."
    ::= { dnsServMgmtBootEntry 2 }

dnsServMgmtBootServerType OBJECT-TYPE
     SYNTAX  INTEGER {
        primary (1),
        secondary (2),
        cache (3),
        slave (4),
        forwarding (5)
    }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This value specifies the type of service for the
      specified domain that is offered by the server
      specified."
    ::= { dnsServMgmtBootEntry 3 }
```

```
dnsServMgmtBootLoad OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the name of the file that is read in for the
      initial load of zone information on startup."
    ::= { dnsServMgmtBootEntry 4 }

dnsServMgmtBootDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the date that the information found in
      dnsServMgmtBootLoad was last changed."
    ::= { dnsServMgmtBootEntry 5 }

dnsServMgmtBootTransfer OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "The name of the server to/from which a zone transfer
      was last made."
    ::= { dnsServMgmtBootEntry 6 }

dnsServMgmtBootTransferDirection OBJECT-TYPE
     SYNTAX  INTEGER {
        send (1),
        receive (2)
    }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This value is direction of the last transfer.  A value
      of 1 indicates that the zone transfer was sent from
      this system to the system identified
      dnsServMgmtBootTransfer.  A value of 2 indicates that
      the transfer was from the system identified in
      dnsServMgmtBootTransfer."
    ::= { dnsServMgmtBootEntry 7 }

dnsServMgmtBootTransferDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
      "This is the date of the last zone transfer."
    ::= { dnsServMgmtBootEntry 8 }

dnsServMgmtAdministrator OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "The name of the domaine administrator found in the SOA
      record for the domain identified in this row."
     ::= { dnsServMgmtBootEntry 9 }
```

```
dnsServMgmtSerial OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "The current serial number of the SOA record in the
      file specified by dnsServMgmtBootLoad."
     ::= { dnsServMgmtBootEntry 10 }

dnsServMgmtRefresh OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "The time interval before the zone should be refreshed
      as found in the SOA record in the file specified by
      dnsServMgmtBootLoad."
     ::= { dnsServMgmtBootEntry 11 }

dnsServMgmtRetry OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "This value is the time interval that should elapse
      before failed refresh attempts are retried.  This value
      is taken from the SOA record specified by
      dnsServMgmtBootLoad."
     ::= { dnsServMgmtBootEntry 12 }

dnsServMgmtExpire OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "This value is the time value which specifies the upper
      limit on the time interval that can elapse before the
      zone is considered to be no longer authoritative. This
      value is taken from the SOA record specified by
      dnsServMgmtBootLoad"
    ::= { dnsServMgmtBootEntry 13 }

dnsServMgmtMinimum OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
      "This value is the TTL field which will be exported
      with any RR from this zone and is specified in the SOA
      record identified by dnsServMgmtBootLoad."
    ::= { dnsServMgmtBootEntry 14 }

END
```

5 Acknowledgements

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

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

6 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).