DNS MIB Extensions

30–June–1992

DNS Working Group

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

Status of this Memo

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

This draft document will be submitted to the RFC editor as a proposed extension to the SNMP MIB. Distribution of this document is unlimited. Please send comments or corrections to the 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 1

2 Selected Objects 1

3 Objects 2
3.1 Format of Definitions 2
3.2 Textual Conventions 2

4 Object Definitions 2

5 Acknowledgements 36

6 References 36

1 Introduction

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

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

New DNS 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
- Resolver Configuration Group
- Server Configuration Group
- Resolver Counters Group
- Server Counters Group
- Records Group
- Resolver Cache Group
- Resolver Negative Cache Group
- Resolver Statistics 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

Several datatypes have been introduced as a textual conventions in this DNS MIB document. These additions will facilitate the common understanding of information used by the DNS. No changes to the SMI or the SNMP are necessary to support these conventions which are described in 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 next two octets represent tenths and
-- hundreths of a second.  The final 5 octets provide a mechanism
-- to relate the host's time to Coordinated Universal Time.  Tuesday
-- May 26, 1992 at 1:30 PM EDT would be represented as:
-- 199205261330.00-500.

DnsName ::= OCTET STRING
-- A DNS name is a sequence of octets with internal length encoding.
-- While it is often the case that the printed form of a DNS name is
-- a dotted-format Internet hostname, this is not always the case.
-- All occurrences of DNS names in this MIB us the internal format
-- used by the DNS protocols embedded within an OCTET STRING.

DnsClass ::= INTEGER (0..65535)
-- This data type is used to represent the class values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new types of Class records to be defined.
-- Existing classes are listed in RFC 1035.

DnsType ::= INTEGER (0..65535)
-- This data type is used to represent the type values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new record types to be defined.
-- Existing types are listed in RFC 1035.

DnsQClass ::= INTEGER (0..65535)
-- This data type is used to represent the QClass values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new types of QClass records to be defined.
-- Existing Q classes are listed in RFC 1035.

DnsQType ::= INTEGER (0..65535)
-- This data type is used to represent the QType values which appear
-- in Resource Records in the DNS.  A 16-bit unsigned integer is
-- used to allow room for new record types to be defined.
-- Existing QTypes are listed in RFC 1035.

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

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


-- groups in the dns mib
```

```
       dnsGenConfig               OBJECT IDENTIFIER ::= { dns 1 }
       dnsResConfig               OBJECT IDENTIFIER ::= { dns 2 }
       dnsServerConfig            OBJECT IDENTIFIER ::= { dns 3 }
       dnsResCounters             OBJECT IDENTIFIER ::= { dns 4 }
       dnsServerCounters          OBJECT IDENTIFIER ::= { dns 5 }
       dnsRecords                 OBJECT IDENTIFIER ::= { dns 6 }
       dnsResCache                OBJECT IDENTIFIER ::= { dns 7 }
       dnsResNCache               OBJECT IDENTIFIER ::= { dns 8 }
       dnsResStats                OBJECT IDENTIFIER ::= { dns 9 }
       dnsServerManagement        OBJECT IDENTIFIER ::= { dns 10 }

-- General Configuration Group

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

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

-- Resolver Configuration Group

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

dnsResConfigId OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
          "Name and version identification of DNS resolver software "
     ::= { dnsResConfig 1 }

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

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

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

dnsResSealtBeltEntry  OBJECT-TYPE
     SYNTAX  DnsResSeatBeltEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
      "An entry in the resolver's seatbelt table."
     INDEX { dnsResSeatBeltAddr,
             dnsResSeatBeltSubTree,
             dnsResSeatBeltClass }
     ::= { dnsResSeatBeltTable 1 }

DnsResSeatBeltEntry ::=
     SEQUENCE {
        dnsResSeatBeltAddr
     IpAddress,
        dnsResSeatBeltName
            DnsName,
        dnsResSeatBeltTOS
            INTEGER,
        dnsResSeatBeltPref
            INTEGER,
        dnsResSeatBeltSubTree
            DnsName,
        dnsResSeatBeltClass
            DnsClass,
        dnsResSeatBeltFile
            OCTET STRING,
        dnsResSeatBeltDate
            DnsDate,
        dnsResSeatBeltValid
            DnsValid
     }
dnsResSeatBeltAddr OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "IP address of a SeatBelt name server."
     ::= { dnsResSeatBeltEntry 1 }
```

```
dnsResSeatBeltName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS name of a SeatBelt nameserver, if known, otherwise
             it is null."
     ::= { dnsResSeatBeltEntry 2 }

dnsResSeatBeltTOS OBJECT-TYPE
     SYNTAX  INTEGER {
          iterative (1),
          recursive (2)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Type-Of-Service resolver expects from seatbelt
             nameserver.  ITERATIVE indicates that resolver will be
             directing iterative queries to this name server (RD
             bit turned off), while RECURSIVE indicates that
             resolver will be directing recursive queries to this
             name server (RD bit turned on)."
     ::= { dnsResSeatBeltEntry 3 }

dnsResSeatBeltPref OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This value identifies the preference for the server
             identified in this row of the table.  The lower the
             value, the higher the preference for the particular
             server."
     ::= { dnsResSeatBeltEntry 4 }

dnsResSeatBeltSubTree OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The DNS sub tree name for queries sent to the name
             server identified by dnsResSeatBeltName."
     ::= { dnsResSeatBeltEntry 5 }

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

dnsResSeatBeltFile OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The name of the source file which contains information
             read by the resolver at initialization."
     ::= { dnsResSeatBeltEntry 7 }
```

```
dnsResSeatBeltDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This is the date information found in
             dnsResSeatBeltFile was last changed."
     ::= { dnsResSeatBeltEntry 8 }

dnsResSeatBeltValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this variable to CLEAR deletes this SeatBelt
             server."
     ::= { dnsResSeatBeltEntry 9 }

-- 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"
    ::= { dnsServerConfig 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."
     ::= { dnsServerConfig 2 }

-- Resolver Counters Group

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

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

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

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

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

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

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

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

```
dnsResCounterTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsResCounterTableEntry
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Table of the current count of resolver queries and
             answers."
    ::= { dnsResCounter 8 }

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

DnsResCounterTableEntry ::=
     SEQUENCE {
         dnsResCounterOpCode
              INTEGER,
         dnsResCounterQueries
              Counter,
         dnsResCounterResponses
              Counter
     }
dnsResCounterOpCode OBJECT-TYPE
     SYNTAX  INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The index to this table.  The opcodes that have
             already been defined are found in RFC1035."
     ::= { dnsResCounterTableEntry 1 }

dnsResCounterQueries
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The total number of queries that have sent out by the
             resolver since initialization for the opcode which is
             index to this row of the table."
     ::= { dnsResCounterTableEntry 2 }

dnsResCounterResponses
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The total number of responses that have been received
             by the resolver since initialization for the opcode
             which is the index to this row of the table."
     ::= { dnsResCounterTableEntry 3 }

-- Additional DNS Resolver Counter Objects
```

```
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."
     ::= { dnsResCounter 9 }

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."
     ::= { dnsResCounter 10 }

dnsResCounterMartians OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were received from
             servers that the resolver does not think it asked."
     ::= { dnsResCounter 11 }

dnsResCounterResponsesRecd OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received to all queries."
     ::= { dnsResCounter 12 }

dnsResCounterDuplicates OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were duplicates and
             ignored by the resolver."
     ::= { dnsResCounter 13 }

dnsResCounterUnparse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which were unparseable."
     ::= { dnsResCounter 14 }

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

```
dnsResCounterOtherFail OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses received which where for all other
             bad responses."
     ::= { dnsResCounter 16 }

dnsResCounterTCPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of attempts over TCP made by the resolver
             to connect with root servers."
     ::= { dnsResCounter 17 }

dnsResCounterUDPAttempts OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of out bound UDP packets sent by the resolver
             resolver to root servers."
     ::= { dnsResCounter 18 }

dnsResCounterNoLocalData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that have been sent because
             they could not be answered with local information by
             the resolver."
     ::= { dnsResCounter 19 }

dnsResCounterReferals OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses which were received from
             servers redirecting query to another server."
     ::= { dnsResCounter 20 }

dnsResCounterRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number requests retransmitted for all reasons."
     ::= { dnsResCounter 21 }

dnsResCounterNoResponse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that were retransmitted because of
             no response."
     ::= { dnsResCounter 22 }
```

```
dnsResCounterRootRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that were retransmitted that were to
             root servers."
     ::= { dnsResCounter 23 }

dnsResCounterLateRetrans OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that were retransmitted as a result
             of late responses."
     ::= { dnsResCounter 24 }

dnsReSCounterLocalTrunc OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of truncated responses due to local truncation
             for forwarded queries."
     ::= { dnsResCounter 25 }

dnsResSCounterTruncRemote OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of truncated responses from remote servers for
             forwarded queries."
     ::= { dnsResCounter 26 }

dnsResCounterInternal OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests internally generated by the
             resolver."
     ::= { dnsResCounter 27 }

dnsResCounterInternalTimeOut OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests internally generated which timed
             out."
     ::= { dnsResCounter 28 }

-- Server Counter Group

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

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

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

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

dnsServCounterResetTime OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed time since cache was last reloaded for any
             reason."
    ::= { dnsServCounter 2 }

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

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

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

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

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

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

dnsServCounterOpCodeTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsServCounterOpCodeTableEntry
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
          "Table of the current count of queries answered by the
          server."
    ::= { dnsServCounter 9 }

dnsServCounterOpCodeTableEntry OBJECT-TYPE
     SYNTAX  DnsServCounterOpCodeTableEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Entry in the server counter OpCode table.  Entries
             are indexed by dns opcode."
     INDEX { dnsServCounterOpCode }
     ::= { dnsServCounterOpCodeTable 1 }
```

```
DnsServCounterOpCodeTableEntry ::=
     SEQUENCE {
         dnsServCounterOpCode
              INTEGER,
         dnsServCounterQueries
              Counter,
         dnsServCounterResponses
              Counter
     }
dnsServCounterOpCode OBJECT-TYPE
     SYNTAX  INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The index to this table.  Opcodes that have
             already been defined are found in RFC1035."
     ::= { dnsServCounterOpCodeTableEntry 1 }

dnsServCounterQueries OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that have been received
             since initialization for the type of query
             identified on this row of the table."
     ::= { dnsServCounterOpCodeTableEntry 2 }

dnsServCounterResponses OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of responses made by the server since
             initialization for the type of query identified
             on this row of the table."
     ::= { dnsServCounterOpCodeTableEntry 3 }

-- Additional DNS Server Counter Objects

dnsServCounterSelfAuth OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed with
             authoritative data which originated from a resolver
             on the same host."
     ::= { dnsServCounter 10 }

dnsServCounterSelfAuthNoName OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been an authoritative no such name answer
             (NXDOMAIN) given."
     ::= { dnsServCounter 11 }
```

```
dnsServCounterSelfAuthNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             there has been an authoritative no such data answer
             (empty answer) made."
     ::= { dnsServCounter 12 }

dnsServCounterSelfNonAuthData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             a non-authoritative answer (cached data) was made."
     ::= { dnsServCounter 13 }

dnsServCounterSelfNonAuthNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host for which
             a non-authoritative answer - no such data response was
             made (empty answer)."
     ::= { dnsServCounter 14 }

dnsServCounterSelfCAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host which
             were answers to C_ANY queries."
     ::= { dnsServCounter 15 }

dnsServCounterSelfIAns OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of inverse queries the server has processed which
             originated from a resolver on the same host which
             were answered with local information."
     ::= { dnsServCounter 16 }

dnsServCounterSelfNonRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of non recursive queries the server has
             processed which originated from a resolver on the
             same host and were referred to other servers."
     ::= { dnsServCounter 17 }
```

```
dnsServCounterSelfErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests the server has processed which
             originated from a resolver on the same host which
             have been answered with error responses (e.g.,
             malformed query)."
     ::= { dnsServCounter 18 }

dnsServCounterSelfRelNames OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received for relative names the
             server has processed which originated from a resolver
             on the same host."
     ::= { dnsServCounter 19 }

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

dnsServCounterFriendsNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends, for
             which authoritative no such name (NXDOMAIN) responses
             were made.  The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 21 }

dnsServCounterFriendsZoneNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends for
             which authoritative no such data (empty answer)
             responses were made.  The definition of friends is
             a locally defined matter."
     ::= { dnsServCounter 22 }

dnsServCounterFriendsCached OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends which
             were non-authoritatively answered (cached data).
             The definition of friends is a locally defined matter."
     ::= { dnsServCounter 23 }
```

```
dnsServCounterFriendsCachedNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from friends for
             were non-authoritatively answered with no such data
             (empty answer)."
     ::= { dnsServCounter 24 }

dnsServCounterFriendsCAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to C_ANY queries which originated
             from friends.  The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 25 }

dnsServCounterFriendsInverse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to inverse queries which originated
             from friends. The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 26 }

dnsServCounterFriendsForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests which originated from friends
             that were forwarded to other servers.  The definition
             of friends is a locally defined matter."
     ::= { dnsServCounter 27 }

dnsServCounterFriendsRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of non recursive queries which originated
             from friends which have been referred to other
             servers.  The definition of friends is a locally
             defined matter."
     ::= { dnsServCounter 28 }

dnsServCounterFriendsErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries which originated from friends
             which were answered with error responses.  The
             definition of friends is a locally defined matter."
     ::= { dnsServCounter 29 }
```

```
dnsServCounterFriendsRel OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests received for relative names from
             friends."
     ::= { dnsServCounter 30 }

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

dnsServCounterRemoteNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, for which
             authoritative no such name (NXDOMAIN) responses
             were made."
     ::= { dnsServCounter 32 }

dnsServCounterRemoteNoData OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, for which
             authoritative no such data (empty answer)
             responses were made."
     ::= { dnsServCounter 33 }

dnsServCounterRemoteCached OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, which were
             non-authoritatively answered (cached data)."
     ::= { dnsServCounter 34 }

dnsServCounterRemoteCachedNoSuch OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries originating from systems which
             have not been defined as friends, which were
             non-authoritatively answered with no such data
             (empty answer)."
     ::= { dnsServCounter 35 }
```

```
dnsServCounterRemoteAny OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to C_ANY queries which originated
             from systems which have not been defined as friends."
     ::= { dnsServCounter 36 }

dnsServCounterRemoteInverse OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of answers to inverse queries which originated
             from systems which have not been defined as friends."
     ::= { dnsServCounter 37 }

dnsServCounterRemoteForwards OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of requests which originated from systems which
             have not been defined as friends that were forwarded to
             other servers."
     ::= { dnsServCounter 38 }

dnsServCounterRemoteRecurs OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of non recursive queries which originated
             from systems which have not been defined as friends
             that have been referred other servers."
     ::= { dnsServCounter 39 }

dnsServCounterRemoteErrors OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries which originated from systems which
             have not been defined as friends that were answered
             with error responses."
     ::= { dnsServCounter 40 }

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

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

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

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

dnsServCounterTruncLocal OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of truncated responses due to local truncation
             for local queries."
     ::= { dnsServCounter 45 }

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."
     ::= { dnsServCounter 46 }

-- DNS Server Counter Table

dnsCounterTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Counter information based on DNS class and
             record type."
    ::= { dnsServCounter 47 }

dnsServCounterEntry OBJECT-TYPE
     SYNTAX  dnsServCounterEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "This table contains count information for each DNS
             record and class type known to the server.  The index
             allows management software to to create indices to the
             table to get the specific information desired, e.g.,
             number or A record queries over UDP which came to this
             server from a specific IP address."
    INDEX  { dnsServCounterClass, dnsServCounterQueryType,
             dnsServCounterQuerySource, dnsServCounterTransport }
    ::= { dnsServCounterTable 1 }
```

```
DnsServCounterEntry ::=
    SEQUENCE {
        dnsServCounterClass
            DnsClass,
        dnsServCounterQueryType
            DnsType,
        dnsServCounterQuerySource
            INTEGER,
        dnsServCounterTransport
            INTEGER,
        dnsServCounterNumber
            Counter
        }
dnsServCounterClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The class of the record type being counted in
             this row the table.  For classes defined in RFC 1035,
             the values are the same.
     ::= { dnsServCounterEntry 1 }

dnsServCounterQueryType OBJECT-TYPE
     SYNTAX DnsType -- INTEGER (0..65535)
     ACCESS read-only
     STATUS mandatory
     DESCRIPTION
             "The type of record which is being counted in this row
             in the table. For types defined in RFC 1035, the
             values are the same as those specified in the RFC."
     ::= { dnsServCounterEntry 2 }

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

```
dnsServCounterTransport OBJECT-TYPE
     SYNTAX  INTEGER {
        udp (1),
        tcp (2),
        other (3)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "A value of 1 indicates that the queries reported on
             this row were sent using UDP.  A value of 2 indicates
             that TCP was used.  3 is for any transport other than
             TCP or UDP."
     ::= { dnsServCounterEntry 4 }

dnsServCounterNumber OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of queries that have been recorded in this row
             of the table."
     ::= { dnsServCounterEntry 5 }

-- Records Group

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

-- Records Table

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

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

dnsRecDefaultEntry  OBJECT-TYPE
     SYNTAX  DnsRecDefaultEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "This table contains Resource Record configuration
             information."
     INDEX { dnsRecDefaultIndex }
     ::= { dnsRecDefaultTable 1 }
```

```
DnsRecDefaultEntry ::=
    SEQUENCE {
        dnsRecDefaultIndex
            INTEGER,
        dnsRecDefaultDomain
            OCTET STRING,
        dnsRecDefaultClass
            DnsClass,
        dnsRecDefaultType
            DnsType,
        dnsRecDefaultTTL
            INTEGER,
        dnsRecDefaultLength
            INTEGER,
        dnsRecDefaultData
            OCTET STRING
        }

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

dnsRecDefaultDomain OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The DNS domain to which this entry belongs."
     ::= { dnsRecDefaultEntry 2 }

dnsRecDefaultClass OBJECT-TYPE
     SYNTAX  DnsClasss -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The class of the record contained in this row.  For
             classes defined in RFC 1035, the values are the same."
     ::= { dnsRecDefaultEntry 3 }

dnsRecDefaultType OBJECT-TYPE
     SYNTAX  DnsType -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The type of the record contained in this row.  For
             types defined in RFC 1035, the values are the same."
     ::= { dnsRecDefaultEntry 3 }

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

```
dnsRecDefaultLength OBJECT-TYPE
     SYNTAX  INTEGER (1..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "The length in octets of the data in dnsRecDefaultData."
     ::= { dnsRecDefaultEntry 5 }

dnsRecDefaultData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Variable length of octets.  The length is defined in
             dnsRecDefaultLength.  The Type and Class information
             provided in this row of the table along with this data
             tell the management station how to interpret the date.
             For information on the details of DNS Resource Records
             and their formats, see RFC1035."
     ::= { dnsRecDefaultEntry 6 }


-- Resolver Cache Group

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

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

dnsResCacheValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this to CLEAR deletes the resolver's entire
             cache."
     ::= { dnsResCache 2 }

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

-- Resolver Cache Table

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

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

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

DnsResCacheEntry ::=
     SEQUENCE {
         dnsResCacheIndex,
             INTEGER,
         dnsResCacheRRName
             DnsName,
         dnsResCacheRRClass
             DnsClass,
         dnsResCacheRRType
             DnsType,
         dnsResCacheRRTTL
             DnsTime,
         dnsResCacheRRElapsedTTL
             DnsTime,
         dnsResCacheRRSource
             IpAddress,
         dnsResCacheRRRData
             OCTET STRING,
         dnsResCacheRRValid
             DnsValid
     }
dnsResCacheRRIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "A unique index value which indentifies each entry in
             this table."
     ::= { dnsResCacheEntry 1 }

dnsResCacheRRName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Owner name of the Resource Record in the cache which
             is identified in this row of the table.  As described
             in RFC1034, the owner of the record is the domain name
             were the RR is found."
     ::= { dnsResCacheEntry 2 }
```

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

dnsResCacheRRType OBJECT-TYPE
     SYNTAX  DnsType -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS type of the Resource Record in the cache which
             is identified in this row of the table."
     ::= { dnsResCacheEntry 4 }

dnsResCacheRRTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Time-To-Live of RR in DNS cache.  This is the initial
             TTL value which was received with the RR when it was
             originally received."
     ::= { dnsResCacheEntry 5 }

dnsResCacheRRElapsedTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since RR was received."
     ::= { dnsResCacheEntry 6 }

dnsResCacheRRSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Host from which RR was received, 0.0.0.0 if unknown."
     ::= { dnsResCacheEntry 7 }

dnsResCacheRRRData OBJECT-TYPE
     SYNTAX  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "RDATA portion of a cached RR."
     ::= { dnsResCacheEntry 8 }

dnsResCacheRRValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Set to CLEAR to delete this RR entry from the
             cache table."
     ::= { dnsResCacheEntry 9 }
```

```
-- Resolver Negative Cache Group

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

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

dnsResNCacheValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "Setting this to CLEAR deletes the resolver's entire
             negative response cache."
     ::= { dnsNResCache 2 }

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

-- Resolver Negative Cache Table

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

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

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

```
DnsResNCacheEntry ::=
     SEQUENCE {
         dnsResNCacheIndex,
             INTEGER,
         dnsResNCacheErrQName
             DnsName,
         dnsResNCacheErrQClass
             DnsQClass,
         dnsResNCacheErrQType
             DnsQType,
         dnsResNCacheErrTTL
             DnsTime,
         dnsResNCacheErrElapsedTTL
             DnsTime,
         dnsResNCacheErrSource
             IpAddress,
         dnsResNCacheErrCode
             INTEGER,
         dnsResNCacheErrValid
             DnsValid
     }
dnsResNCacheIndex OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "A unique index value which indentifies each entry in
             the Negative Cache table."
     ::= { dnsResNCacheEntry 1 }

dnsResNCacheErrQName OBJECT-TYPE
     SYNTAX  DnsName  --  OCTET STRING
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "QNAME associated with a cached authoritative error."
     ::= { dnsResNCacheEntry 2 }

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

dnsResNCacheErrQType OBJECT-TYPE
     SYNTAX  DnsQType -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS QTYPE associated with a cached authoritative error."
     ::= { dnsResNCacheEntry 4 }

dnsResNCacheErrTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Time-To-Live of a cached authoritative error.
             This should be the TTL as copied from the MINIMUM
             field of the SOA that accompanied the authoritative
             error."
     ::= { dnsResNCacheEntry 5 }
```

```
dnsResNCacheErrElapsedTTL OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since authoritative error was received."
     ::= { dnsResNCacheEntry 6 }

dnsResNCacheErrSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Host which sent the authoritative error, 0.0.0.0 if
             unknown."
     ::= { dnsResNCacheEntry 7 }

dnsResNCacheErrCode OBJECT-TYPE
     SYNTAX  INTEGER {
         nonexistantName (1),
         noData (2),
         other (3)
     }
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
            "The authoritative error that has been cached.
             NonexistantName indicates a cached authoritative name
             error.  NoData indicates a cached authoritative
             response with no error and no relevant data."
     ::= { dnsResNCacheEntry 8 }

dnsResNCacheErrValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
            "Set to CLEAR to delete this cached error."
     ::= { dnsResNCacheEntry 9 }

-- Resolver Statistics Group

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

dnsResStatGoodCaches OBJECT-TYPE
     SYNTAX  Counter
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Number of RRs the resolver has cached successfully."
     ::= { dnsResStats 1 }

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

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

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

-- Lame Delegation Table

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

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

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

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

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

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

dnsResLameDelegationValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
            "Set to CLEAR to delete this lame delegation entry."
     ::= { dnsResLameDelegationEntry 5 }

-- Server Management Group

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

dnsMgmtVersion OBJECT-TYPE
     SYNTAX  DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Name and version information for name server software."
     ::= { dnsMgmt 1 }

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

-- DNS Managment Zone Configuration Table
```

```
dnsMgmtZoneTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsNsZoneEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Table of zones for which this name server is
             configured.  If name server doesn't load any zones,
             this table is empty."
     ::= { dnsMgmt 3 }

dnsMgmtZoneEntry OBJECT-TYPE
     SYNTAX  DnsMgmtZoneEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "An entry in the name server zone table."
     INDEX { dnsMgmtZoneName,
             dnsMgmtZoneClass }
     ::= { dnsMgmtZoneTable 1 }

DnsMgmtZoneEntry ::=
     SEQUENCE {
         dnsMgmtZoneName
             DnsName,
         dnsMgmtZoneLoad
             DisplayString,
         dnsMgmtZoneDate
             DnsDate,
         dnsMgmtZoneClass
             DnsClass,
         dnsMgmtZoneLastReload
             DnsTime,
         dnsMgmtZoneLastReloadAttempt
             DnsTime,
         dnsMgmtZoneLastSource
             IpAddress,
         dnsMgmtZoneSerial
             INTEGER,
         dnsMgmtZoneRefresh
             DnsTime
         dnsMgmtZoneRetry
             DnsTime,
         dnsMgmtZoneExpire
             DnsTime,
         dnsMgmtZoneMin,
             INTEGER,
         dnsMgmtZoneValid
             DnsValid
     }
dnsMgmtZoneName OBJECT-TYPE
     SYNTAX  DnsName
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS name of SOA RR defining top of this zone."
     ::= { dnsMgmtZoneEntry 1 }

dnsMgmtZoneLoad 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."
    ::= { dnsMgmtZoneEntry 2 }
```

```
dnsMgmtZoneDate OBJECT-TYPE
     SYNTAX  DnsDate -- DisplayString
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "This is the date that the information found in
             dnsMgmtZoneLoad was last changed."
    ::= { dnsMgmtZoneEntry 3 }

dnsMgmtZoneClass OBJECT-TYPE
     SYNTAX  DnsClass -- INTEGER (0..65535)
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "DNS class of the RRs in this zone."
     ::= { dnsMgmtZoneEntry 4 }

dnsMgmtZoneLastReload OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since last successful reload of
              this zone."
     ::= { dnsMgmtZoneEntry 5 }

dnsMgmtZoneLastReloadAttempt OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "Elapsed seconds since last attempted reload of
             this zone."
     ::= { dnsMgmtZoneEntry 6 }

dnsMgmtZoneLastSource OBJECT-TYPE
     SYNTAX  IpAddress
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "IP address of host from which most recent reload of
              this zone was received.  If unknown or irrelevant,
              value should be 0.0.0.0."
     ::= { dnsMgmtZoneEntry 7 }

dnsMgmtSerial OBJECT-TYPE
     SYNTAX  INTEGER
     ACCESS  read-write
     STATUS  mandatory
     DESCRIPTION
             "The current serial number of the SOA record in the
             file specified by dnsServMgmtBootLoad."
     ::= { dnsMgmtZoneEntry 8 }

dnsMgmtZoneRefresh OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "REFRESH interval for zone.  This is the REFRESH field
              of the zone's SOA RR, if known, otherwise whatever
              value the name server is using in place of the SOA
              REFRESH value."
     ::= { dnsMgmtZoneEntry 9 }
```

```
dnsMgmtZoneRetry OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "RETRY interval for zone.  This is the RETRY field of
              the zone's SOA RR, if known, otherwise whatever value
              the name server is using in place of the SOA RETRY
              value."
     ::= { dnsMgmtZoneEntry 10 }

dnsMgmtZoneExpire OBJECT-TYPE
     SYNTAX  DnsTime
     ACCESS  read-only
     STATUS  mandatory
     DESCRIPTION
             "EXPIRE interval for zone.  This is the EXPIRE field of
              the zone's SOA RR, if known, otherwise whatever value
              the name server is using in place of the SOA EXPIRE
              value."
     ::= { dnsMgmtZoneEntry 11 }

dnsMgmtZoneMin 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 12 }

dnsMgmtZoneValid OBJECT-TYPE
     SYNTAX  DnsValid {
          valid (1),
          clear (2)
     }
     ACCESS  write-only
     STATUS  mandatory
     DESCRIPTION
             "Set to CLEAR to delete this zone entry."
     ::= { dnsMgmtZoneEntry 13 }

-- DNS Management Server Table

dnsMgmtServTable OBJECT-TYPE
     SYNTAX  SEQUENCE OF DnsMgmtServEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "Table of IP addresses of name server hosts from which
              this name server might obtain particular zones.
              Typically this indicates the addresses of the primary
              name servers of zones for which the SNMP agent's host
              is a secondary name server."
     ::= { dnsMgmt 4 }

dnsMgmtServEntry OBJECT-TYPE
     SYNTAX  DnsMgmtServEntry
     ACCESS  not-accessible
     STATUS  mandatory
     DESCRIPTION
             "An entry in the name server server (sic) table."
     INDEX { dnsMgmtServName,
             dnsMgmtServClass,
             dnsMgmtServAddr }
     ::= { dnsMgmtServTable 1 }
```

```
DnsMgmtServEntry ::=
     SEQUENCE {
         dnsMgmtServName
             DnsName,
         dnsMgmtServClass
             DnsClass,
         dnsMgmtServAddr
             IpAddress
     }

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

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

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

5 Acknowledgements

This document is the result of work undertaken the by DNS working group. Special thanks are due to Rob Austein of Epilogue Technology for the major contributions he has made in the organization of the MIB as well as authoring many of the objects. In addition, the contributions and comments of the following members are also specially acknowledged:

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

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).