

DNS Resolver MIB Extensions

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Table of Contents

1. Introduction	1
2. The SNMPv2 Network Management Framework	2
2.1 Object Definitions	2
3. Overview	2
3.1 Resolvers	3
3.2 Name Servers	3
3.3 Selected Objects	4
3.4 Textual Conventions	4
4. Definitions	5
5. Acknowledgements	30
6. References	30
7. Security Considerations	32
8. Authors' Addresses	32

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a set of extensions which instrument DNS resolver functions. This memo was produced by the DNS working group.

With the adoption of the Internet-standard Network Management Framework [4,5,6,7], and with 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 was 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 resolver software in an integrated fashion.

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

2. The SNMPv2 Network Management Framework

The SNMPv2 Network Management Framework consists of four major components. They are:

- o RFC 1442 which defines the SMI, the mechanisms used for describing and naming objects for the purpose of management.
- o STD 17, RFC 1213 defines MIB-II, the core set of managed objects for the Internet suite of protocols.
- o RFC 1445 which defines the administrative and other architectural aspects of the framework.
- o RFC 1448 which defines the protocol used for network access to managed objects.

The Framework permits new objects to be defined for the purpose of experimentation and evaluation.

2.1. Object Definitions

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the subset of Abstract Syntax Notation One (ASN.1) defined in the SMI. In particular, each object type is named by an OBJECT IDENTIFIER, an administratively assigned name. The object type together with an object instance serves to uniquely identify a specific instantiation of the object. For human convenience, we often use a textual string, termed the descriptor, to refer to the object type.

3. Overview

In theory, the DNS world is pretty simple. There are two kinds of entities: resolvers and name servers. Resolvers ask questions. Name servers answer them. The real world, however, is not so simple.

Implementors have made widely differing choices about how to divide DNS functions between resolvers and servers. They have also constructed various sorts of exotic hybrids. The most difficult task in defining this MIB was to accommodate this wide range of entities without having to come up with a separate MIB for each.

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

3.1. Resolvers

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

3.2. Name Servers

A name server is a program (or piece thereof) that provides resource records to resolvers. All references in this document to "a name server" imply "the name server's role"; in some cases the name server's role and the resolver's role might be combined into a single program. A name server receives DNS protocol queries and sends DNS protocol replies. A name server neither sends queries nor receives replies. As a consequence, name servers do not have caches. Normally, a name server would expect to receive only those queries to which it could respond with authoritative information. However, if a

name server receives a query that it cannot respond to with purely authoritative information, it may choose to try to obtain the necessary additional information from a resolver which may or may not be a separate process.

3.3. Selected Objects

Many of the objects included in this memo have been created from information contained in the DNS specifications [1,2], as amended and clarified by subsequent host requirements documents [3]. Other objects have been created based on experience with existing DNS management tools, expected operational needs, the statistics generated by existing DNS implementations, and the configuration files used by existing DNS implementations. These objects have been ordered into groups as follows:

- o Resolver Configuration Group
- o Resolver Counter Group
- o Resolver Lame Delegation Group
- o Resolver Cache Group
- o Resolver Negative Cache Group
- o Resolver Optional Counter Group

This information has been converted into a standard form using the SNMPv2 SMI defined in [9]. For the most part, the descriptions are influenced by the DNS related RFCs noted above. For example, the descriptions 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 [2].

3.4. Textual Conventions

Several conceptual data types have been introduced as a textual conventions in the DNS Server MIB document and have been imported into this MIB module. 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.

Readers familiar with MIBs designed to manage entities in the lower layers of the Internet protocol suite may be surprised at the number of non-enumerated integers used in this MIB to represent values such as DNS RR class and type numbers. The reason for this choice is simple: the DNS itself is designed as an extensible protocol,

allowing new classes and types of resource records to be added to the protocol without recoding the core DNS software. Using non-enumerated integers to represent these data types in this MIB allows the MIB to accommodate these changes as well.

4. Definitions

```
DNS-RESOLVER-MIB DEFINITIONS ::= BEGIN
```

IMPORTS

```
    MODULE-IDENTITY, OBJECT-TYPE, IpAddress, Counter32, Integer32
        FROM SNMPv2-SMI
    TEXTUAL-CONVENTION, RowStatus, DisplayString
        FROM SNMPv2-TC
    MODULE-COMPLIANCE, OBJECT-GROUP
        FROM SNMPv2-CONF
    dns, DnsName, DnsNameAsIndex, DnsClass, DnsType, DnsQClass,
    DnsQType, DnsTime, DnsOpCode, DnsRespCode
        FROM DNS-SERVER-MIB;
```

```
-- DNS Resolver MIB
```

dnsResMIB MODULE-IDENTITY

```
    LAST-UPDATED "9401282250Z"
```

```
    ORGANIZATION "IETF DNS Working Group"
```

CONTACT-INFO

```
    "      Rob Austein
    Postal: Epilogue Technology Corporation
            268 Main Street, Suite 283
            North Reading, MA 10864
            US
    Tel: +1 617 245 0804
    Fax: +1 617 245 8122
    E-Mail: sra@epilogue.com
```

```
    Jon Saperia
    Postal: Digital Equipment Corporation
            110 Spit Brook Road
            ZK01-3/H18
            Nashua, NH 03062-2698
            US
    Tel: +1 603 881 0480
    Fax: +1 603 881 0120
    E-mail: saperia@zko.dec.com"
```

DESCRIPTION

```
    "The MIB module for entities implementing the client
    (resolver) side of the Domain Name System (DNS)
    protocol."
```

```

 ::= { dns 2 }

dnsResMIBObjects          OBJECT IDENTIFIER ::= { dnsResMIB 1 }

-- (Old-style) groups in the DNS resolver MIB.

dnsResConfig              OBJECT IDENTIFIER ::= { dnsResMIBObjects 1 }
dnsResCounter             OBJECT IDENTIFIER ::= { dnsResMIBObjects 2 }
dnsResLameDelegation      OBJECT IDENTIFIER ::= { dnsResMIBObjects 3 }
dnsResCache               OBJECT IDENTIFIER ::= { dnsResMIBObjects 4 }
dnsResNCache              OBJECT IDENTIFIER ::= { dnsResMIBObjects 5 }
dnsResOptCounter          OBJECT IDENTIFIER ::= { dnsResMIBObjects 6 }

-- Resolver Configuration Group

dnsResConfigImplementIdent OBJECT-TYPE
    SYNTAX      DisplayString
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "The implementation identification string for the
         resolver software in use on the system, for example;
         'RES-2.1'"
    ::= { dnsResConfig 1 }

dnsResConfigService OBJECT-TYPE
    SYNTAX      INTEGER { recursiveOnly(1),
                          iterativeOnly(2),
                          recursiveAndIterative(3) }
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "Kind of DNS resolution service provided:

         recursiveOnly(1) indicates a stub resolver.

         iterativeOnly(2) indicates a normal full service
         resolver.

         recursiveAndIterative(3) indicates a full-service
         resolver which performs a mix of recursive and iterative
         queries."
    ::= { dnsResConfig 2 }

dnsResConfigMaxCnames OBJECT-TYPE
    SYNTAX      INTEGER (0..2147483647)
    MAX-ACCESS   read-write

```

STATUS current
 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."

REFERENCE
 "RFC-1035 section 7.1."
 ::= { dnsResConfig 3 }

-- DNS Resolver Safety Belt Table

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

dnsResConfigSbeltEntry OBJECT-TYPE
 SYNTAX DnsResConfigSbeltEntry
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION
 "An entry in the resolver's Sbelt table. Rows may be created or deleted at any time by the DNS resolver and by SNMP SET requests. Whether the values changed via SNMP are saved in stable storage across 'reset' operations is implementation-specific."
 INDEX { dnsResConfigSbeltAddr,
 dnsResConfigSbeltSubTree,
 dnsResConfigSbeltClass }
 ::= { dnsResConfigSbeltTable 1 }

DnsResConfigSbeltEntry ::=
 SEQUENCE {
 dnsResConfigSbeltAddr
 IpAddress,
 dnsResConfigSbeltName
 DnsName,
 dnsResConfigSbeltRecursion
 INTEGER,
 dnsResConfigSbeltPref
 INTEGER,
 dnsResConfigSbeltSubTree

```

        DnsNameAsIndex,
    dnsResConfigSbeltClass
        DnsClass,
    dnsResConfigSbeltStatus
        RowStatus
    }

dnsResConfigSbeltAddr OBJECT-TYPE
    SYNTAX      IPAddress
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The IP address of the Sbelt name server identified by
         this row of the table."
    ::= { dnsResConfigSbeltEntry 1 }

dnsResConfigSbeltName OBJECT-TYPE
    SYNTAX      DnsName
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "The DNS name of a Sbelt nameserver identified by this
         row of the table.  A zero-length string indicates that
         the name is not known by the resolver."
    ::= { dnsResConfigSbeltEntry 2 }

dnsResConfigSbeltRecursion OBJECT-TYPE
    SYNTAX      INTEGER { iterative(1),
                           recursive(2),
                           recursiveAndIterative(3) }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Kind of queries resolver will be sending to the name
         server identified in this row of the table:

         iterative(1) indicates that resolver will be directing
         iterative queries to this name server (RD bit turned
         off).

         recursive(2) indicates that resolver will be directing
         recursive queries to this name server (RD bit turned
         on).

         recursiveAndIterative(3) indicates that the resolver
         will be directing both recursive and iterative queries
         to the server identified in this row of the table."
    ::= { dnsResConfigSbeltEntry 3 }

```

dnsResConfigSbeltPref OBJECT-TYPE

SYNTAX INTEGER (0..2147483647)

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This value identifies the preference for the name server identified in this row of the table. The lower the value, the more desirable the resolver considers this server."

::= { dnsResConfigSbeltEntry 4 }

dnsResConfigSbeltSubTree OBJECT-TYPE

SYNTAX DnsNameAsIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Queries sent to the name server identified by this row of the table are limited to those for names in the name subtree identified by this variable. If no such limitation applies, the value of this variable is the name of the root domain (a DNS name consisting of a single zero octet)."

::= { dnsResConfigSbeltEntry 5 }

dnsResConfigSbeltClass OBJECT-TYPE

SYNTAX DnsClass

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The class of DNS queries that will be sent to the server identified by this row of the table."

::= { dnsResConfigSbeltEntry 6 }

dnsResConfigSbeltStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Row status column for this row of the Sbelt table."

::= { dnsResConfigSbeltEntry 7 }

dnsResConfigUpTime OBJECT-TYPE

SYNTAX DnsTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the resolver has a persistent state (e.g., a process), this value will be the time elapsed since it

started. For software without persistent state, this value will be 0."
 ::= { dnsResConfig 5 }

dnsResConfigResetTime OBJECT-TYPE

SYNTAX DnsTime

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"If the resolver has a persistent state (e.g., a process) and supports a 'reset' operation (e.g., can be told to re-read configuration files), this value will be the time elapsed since the last time the resolver was 'reset.' For software that does not have persistence or does not support a 'reset' operation, this value will be zero."

::= { dnsResConfig 6 }

dnsResConfigReset OBJECT-TYPE

SYNTAX INTEGER { other(1),
 reset(2),
 initializing(3),
 running(4) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status/action object to reinitialize any persistent resolver state. When set to reset(2), any persistent resolver state (such as a process) is reinitialized as if the resolver had just been started. This value will never be returned by a read operation. When read, one of the following values will be returned:

other(1) - resolver in some unknown state;
 initializing(3) - resolver (re)initializing;
 running(4) - resolver currently running."

::= { dnsResConfig 7 }

-- Resolver Counters Group

-- Resolver Counter Table

dnsResCounterByOpcodeTable OBJECT-TYPE

SYNTAX SEQUENCE OF DnsResCounterByOpcodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of the current count of resolver queries and

```

        answers."
 ::= { dnsResCounter 3 }

dnsResCounterByOpcodeEntry OBJECT-TYPE
    SYNTAX      DnsResCounterByOpcodeEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "Entry in the resolver counter table.  Entries are
         indexed by DNS OpCode."
    INDEX        { dnsResCounterByOpcodeCode }
    ::= { dnsResCounterByOpcodeTable 1 }

DnsResCounterByOpcodeEntry ::=
    SEQUENCE {
        dnsResCounterByOpcodeCode
            DnsOpCode,
        dnsResCounterByOpcodeQueries
            Counter32,
        dnsResCounterByOpcodeResponses
            Counter32
    }

dnsResCounterByOpcodeCode OBJECT-TYPE
    SYNTAX      DnsOpCode
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The index to this table.  The OpCodes that have already
         been defined are found in RFC-1035."
    REFERENCE
        "RFC-1035 section 4.1.1."
    ::= { dnsResCounterByOpcodeEntry 1 }

dnsResCounterByOpcodeQueries OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current
    DESCRIPTION
        "Total number of queries that have sent out by the
         resolver since initialization for the OpCode which is
         the index to this row of the table."
    ::= { dnsResCounterByOpcodeEntry 2 }

dnsResCounterByOpcodeResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS   read-only
    STATUS       current

```

DESCRIPTION

"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."

::= { dnsResCounterByOpcodeEntry 3 }

-- Resolver Response Code Counter Table

dnsResCounterByRcodeTable OBJECT-TYPE

SYNTAX SEQUENCE OF DnsResCounterByRcodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of the current count of responses to resolver queries."

::= { dnsResCounter 4 }

dnsResCounterByRcodeEntry OBJECT-TYPE

SYNTAX DnsResCounterByRcodeEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Entry in the resolver response table. Entries are indexed by DNS response code."

INDEX { dnsResCounterByRcodeCode }

::= { dnsResCounterByRcodeTable 1 }

DnsResCounterByRcodeEntry ::=

SEQUENCE {

dnsResCounterByRcodeCode

DnsRespCode,

dnsResCounterByRcodeResponses

Counter32

}

dnsResCounterByRcodeCode OBJECT-TYPE

SYNTAX DnsRespCode

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The index to this table. The Response Codes that have already been defined are found in RFC-1035."

REFERENCE

"RFC-1035 section 4.1.1."

::= { dnsResCounterByRcodeEntry 1 }

```
dnsResCounterByRcodeResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of responses the resolver has received for the
         response code value which identifies this row of the
         table."
    ::= { dnsResCounterByRcodeEntry 2 }

-- Additional DNS Resolver Counter Objects

dnsResCounterNonAuthDataResps OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of requests made by the resolver for which a
         non-authoritative answer (cached data) was received."
    ::= { dnsResCounter 5 }

dnsResCounterNonAuthNoDataResps OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of requests made by the resolver for which a
         non-authoritative answer - no such data response (empty
         answer) was received."
    ::= { dnsResCounter 6 }

dnsResCounterMartians OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of responses received which were received from
         servers that the resolver does not think it asked."
    ::= { dnsResCounter 7 }

dnsResCounterRecdResponses OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of responses received to all queries."
    ::= { dnsResCounter 8 }
```

dnsResCounterUnparseResps OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of responses received which were unparseable."

::= { dnsResCounter 9 }

dnsResCounterFallbacks OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of times the resolver had to fall back to its
seat belt information."

::= { dnsResCounter 10 }

-- Lame Delegation Group

dnsResLameDelegationOverflows OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of times the resolver attempted to add an entry
to the Lame Delegation table but was unable to for some
reason such as space constraints."

::= { dnsResLameDelegation 1 }

-- Lame Delegation Table

dnsResLameDelegationTable OBJECT-TYPE

SYNTAX SEQUENCE OF DnsResLameDelegationEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Table of name servers returning lame delegations.

A lame delegation has occurred when a parent zone
delegates authority for a child zone to a server that
appears not to think that it is authoritative for the
child zone in question."

::= { dnsResLameDelegation 2 }

dnsResLameDelegationEntry OBJECT-TYPE

SYNTAX DnsResLameDelegationEntry

MAX-ACCESS not-accessible

```

STATUS      current
DESCRIPTION
    "Entry in lame delegation table.  Only the resolver may
    create rows in this table.  SNMP SET requests may be used
    to delete rows."
INDEX       { dnsResLameDelegationSource,
              dnsResLameDelegationName,
              dnsResLameDelegationClass }
 ::= { dnsResLameDelegationTable 1 }

DnsResLameDelegationEntry ::=
SEQUENCE {
    dnsResLameDelegationSource
        IPAddress,
    dnsResLameDelegationName
        DnsNameAsIndex,
    dnsResLameDelegationClass
        DnsClass,
    dnsResLameDelegationCounts
        Counter32,
    dnsResLameDelegationStatus
        RowStatus
}

dnsResLameDelegationSource OBJECT-TYPE
SYNTAX      IPAddress
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Source of lame delegation."
 ::= { dnsResLameDelegationEntry 1 }

dnsResLameDelegationName OBJECT-TYPE
SYNTAX      DnsNameAsIndex
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "DNS name for which lame delegation was received."
 ::= { dnsResLameDelegationEntry 2 }

dnsResLameDelegationClass OBJECT-TYPE
SYNTAX      DnsClass
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "DNS class of received lame delegation."
 ::= { dnsResLameDelegationEntry 3 }

```

dnsResLameDelegationCounts OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"How many times this lame delegation has been received."

::= { dnsResLameDelegationEntry 4 }

dnsResLameDelegationStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status column for the lame delegation table. Since only the agent (DNS resolver) creates rows in this table, the only values that a manager may write to this variable are active(1) and destroy(6)."

::= { dnsResLameDelegationEntry 5 }

-- Resolver Cache Group**dnsResCacheStatus OBJECT-TYPE**

SYNTAX INTEGER { enabled(1), disabled(2), clear(3) }

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status/action for the resolver's cache.

enabled(1) means that the use of the cache is allowed. Query operations can return this state.

disabled(2) means that the cache is not being used. Query operations can return this state.

Setting this variable to clear(3) deletes the entire contents of the resolver's cache, but does not otherwise change the resolver's state. The status will retain its previous value from before the clear operation (i.e., enabled(1) or disabled(2)). The value of clear(3) can NOT be returned by a query operation."

::= { dnsResCache 1 }

dnsResCacheMaxTTL OBJECT-TYPE

SYNTAX DnsTime

MAX-ACCESS read-write

STATUS current

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 zero."
 ::= { dnsResCache 2 }

dnsResCacheGoodCaches OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of RRs the resolver has cached successfully."
 ::= { dnsResCache 3 }

dnsResCacheBadCaches OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of RRs the resolver has refused to cache because
        they appear to be dangerous or irrelevant.  E.g., RRs
        with suspiciously high TTLs, unsolicited root
        information, or that just don't appear to be relevant to
        the question the resolver asked."
 ::= { dnsResCache 4 }

-- Resolver Cache Table

dnsResCacheRRTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF DnsResCacheRREntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains information about all the resource
        records currently in the resolver's cache."
 ::= { dnsResCache 5 }

dnsResCacheRREntry OBJECT-TYPE
    SYNTAX      DnsResCacheRREntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "An entry in the resolvers's cache.  Rows may be created
        only by the resolver.  SNMP SET requests may be used to
        delete rows."
    INDEX      { dnsResCacheRRName,
                 dnsResCacheRRClass,
                 dnsResCacheRRType,
                 dnsResCacheRRIndex }

```

```

 ::= { dnsResCacheRRTable 1 }

DnsResCacheRREntry ::=
  SEQUENCE {
    dnsResCacheRRName
      DnsNameAsIndex,
    dnsResCacheRRClass
      DnsClass,
    dnsResCacheRRType
      DnsType,
    dnsResCacheRRTTL
      DnsTime,
    dnsResCacheRRElapsedTTL
      DnsTime,
    dnsResCacheRRSource
      IPAddress,
    dnsResCacheRRData
      OCTET STRING,
    dnsResCacheRRStatus
      RowStatus,
    dnsResCacheRRIndex
      Integer32,
    dnsResCacheRRPrettyName
      DnsName
  }

dnsResCacheRRName OBJECT-TYPE
  SYNTAX      DnsNameAsIndex
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "Owner name of the Resource Record in the cache which is
     identified in this row of the table. As described in
     RFC-1034, the owner of the record is the domain name
     were the RR is found."
  REFERENCE
    "RFC-1034 section 3.6."
  ::= { dnsResCacheRREntry 1 }

dnsResCacheRRClass OBJECT-TYPE
  SYNTAX      DnsClass
  MAX-ACCESS  not-accessible
  STATUS      current
  DESCRIPTION
    "DNS class of the Resource Record in the cache which is
     identified in this row of the table."
  ::= { dnsResCacheRREntry 2 }

```

dnsResCacheRRType OBJECT-TYPE

SYNTAX DnsType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION

"DNS type of the Resource Record in the cache which is identified in this row of the table."

::= { dnsResCacheRREntry 3 }

dnsResCacheRRTTL OBJECT-TYPE

SYNTAX DnsTime
MAX-ACCESS read-only
STATUS current
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."

::= { dnsResCacheRREntry 4 }

dnsResCacheRRElapsedTTL OBJECT-TYPE

SYNTAX DnsTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"Elapsed seconds since RR was received."

::= { dnsResCacheRREntry 5 }

dnsResCacheRRSource OBJECT-TYPE

SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"Host from which RR was received, 0.0.0.0 if unknown."

::= { dnsResCacheRREntry 6 }

dnsResCacheRRData OBJECT-TYPE

SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION

"RDATA portion of a cached RR. The value is in the format defined for the particular DNS class and type of the resource record."

REFERENCE

"RFC-1035 section 3.2.1."

::= { dnsResCacheRREntry 7 }

dnsResCacheRRStatus OBJECT-TYPE

SYNTAX RowStatus
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"Status column for the resolver cache table. Since only the agent (DNS resolver) creates rows in this table, the only values that a manager may write to this variable are active(1) and destroy(6)."

::= { dnsResCacheRREntry 8 }

dnsResCacheRRIndex OBJECT-TYPE

SYNTAX Integer32
 MAX-ACCESS not-accessible
 STATUS current
 DESCRIPTION

"A value which makes entries in the table unique when the other index values (dnsResCacheRRName, dnsResCacheRRClass, and dnsResCacheRRType) do not provide a unique index."

::= { dnsResCacheRREntry 9 }

dnsResCacheRRPrettyName OBJECT-TYPE

SYNTAX DnsName
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION

"Name of the RR at this row in the table. This is identical to the dnsResCacheRRName variable, except that character case is preserved in this variable, per DNS conventions."

REFERENCE

"RFC-1035 section 2.3.3."

::= { dnsResCacheRREntry 10 }

-- Resolver Negative Cache Group**dnsResNCacheStatus OBJECT-TYPE**

SYNTAX INTEGER { enabled(1), disabled(2), clear(3) }
 MAX-ACCESS read-write
 STATUS current
 DESCRIPTION

"Status/action for the resolver's negative response cache."

enabled(1) means that the use of the negative response cache is allowed. Query operations can return this state.

disabled(2) means that the negative response cache is not being used. Query operations can return this state.

Setting this variable to clear(3) deletes the entire contents of the resolver's negative response cache. The status will retain its previous value from before the clear operation (i.e., enabled(1) or disabled(2)). The value of clear(3) can NOT be returned by a query operation."

```
::= { dnsResNCache 1 }
```

dnsResNCacheMaxTTL OBJECT-TYPE

SYNTAX DnsTime

MAX-ACCESS read-write

STATUS current

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 zero."

```
::= { dnsResNCache 2 }
```

dnsResNCacheGoodNCaches OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of authoritative errors the resolver has cached successfully."

```
::= { dnsResNCache 3 }
```

dnsResNCacheBadNCaches OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

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."

REFERENCE

"RFC-1034 section 4.3.4."

```
::= { dnsResNCache 4 }
```

-- Resolver Negative Cache Table

dnsResNCacheErrTable OBJECT-TYPE

SYNTAX SEQUENCE OF DnsResNCacheErrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The resolver's negative response cache. This table contains information about authoritative errors that have been cached by the resolver."

::= { dnsResNCache 5 }

dnsResNCacheErrEntry OBJECT-TYPE

SYNTAX DnsResNCacheErrEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the resolver's negative response cache table. Only the resolver can create rows. SNMP SET requests may be used to delete rows."

INDEX { dnsResNCacheErrQName,
dnsResNCacheErrQClass,
dnsResNCacheErrQType,
dnsResNCacheErrIndex }

::= { dnsResNCacheErrTable 1 }

DnsResNCacheErrEntry ::=

SEQUENCE {

dnsResNCacheErrQName

DnsNameAsIndex,

dnsResNCacheErrQClass

DnsQClass,

dnsResNCacheErrQType

DnsQType,

dnsResNCacheErrTTL

DnsTime,

dnsResNCacheErrElapsedTTL

DnsTime,

dnsResNCacheErrSource

IpAddress,

dnsResNCacheErrCode

INTEGER,

dnsResNCacheErrStatus

RowStatus,

dnsResNCacheErrIndex

Integer32,

dnsResNCacheErrPrettyName

DnsName

}

dnsResNCacheErrQName OBJECT-TYPE

SYNTAX DnsNameAsIndex

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION
 "QNAME associated with a cached authoritative error."
REFERENCE
 "RFC-1034 section 3.7.1."
 ::= { dnsResNCacheErrEntry 1 }

dnsResNCacheErrQClass OBJECT-TYPE
SYNTAX DnsQClass
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "DNS QCLASS associated with a cached authoritative
 error."
 ::= { dnsResNCacheErrEntry 2 }

dnsResNCacheErrQType OBJECT-TYPE
SYNTAX DnsQType
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION
 "DNS QTYPE associated with a cached authoritative error."
 ::= { dnsResNCacheErrEntry 3 }

dnsResNCacheErrTTL OBJECT-TYPE
SYNTAX DnsTime
MAX-ACCESS read-only
STATUS current
DESCRIPTION
 "Time-To-Live of a cached authoritative error at the time
 of the error, it should not be decremented by the number
 of seconds since it was received. This should be the
 TTL as copied from the MINIMUM field of the SOA that
 accompanied the authoritative error, or a smaller value
 if the resolver implements a ceiling on negative
 response cache TTLs."
REFERENCE
 "RFC-1034 section 4.3.4."
 ::= { dnsResNCacheErrEntry 4 }

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

dnsResNCacheErrSource OBJECT-TYPE

SYNTAX IpAddress

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Host which sent the authoritative error, 0.0.0.0 if unknown."

::= { dnsResNCacheErrEntry 6 }

dnsResNCacheErrCode OBJECT-TYPE

SYNTAX INTEGER { nonexistentName(1), noData(2), other(3) }

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The authoritative error that has been cached:

nonexistentName(1) indicates an authoritative name error (RCODE = 3).

noData(2) indicates an authoritative response with no error (RCODE = 0) and no relevant data.

other(3) indicates some other cached authoritative error. At present, no such errors are known to exist."

::= { dnsResNCacheErrEntry 7 }

dnsResNCacheErrStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"Status column for the resolver negative response cache table. Since only the agent (DNS resolver) creates rows in this table, the only values that a manager may write to this variable are active(1) and destroy(6)."

::= { dnsResNCacheErrEntry 8 }

dnsResNCacheErrIndex OBJECT-TYPE

SYNTAX Integer32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A value which makes entries in the table unique when the other index values (dnsResNCacheErrQName, dnsResNCacheErrQClass, and dnsResNCacheErrQType) do not provide a unique index."

::= { dnsResNCacheErrEntry 9 }

dnsResNCacheErrPrettyName OBJECT-TYPE

SYNTAX DnsName

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"QNAME associated with this row in the table. This is identical to the dnsResNCacheErrQName variable, except that character case is preserved in this variable, per DNS conventions."

REFERENCE

"RFC-1035 section 2.3.3."

::= { dnsResNCacheErrEntry 10 }

-- Resolver Optional Counters Group**dnsResOptCounterReferrals OBJECT-TYPE**

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of responses which were received from servers redirecting query to another server."

::= { dnsResOptCounter 1 }

dnsResOptCounterRetrans OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number requests retransmitted for all reasons."

::= { dnsResOptCounter 2 }

dnsResOptCounterNoResponses OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of queries that were retransmitted because of no response."

::= { dnsResOptCounter 3 }

dnsResOptCounterRootRetrans OBJECT-TYPE

SYNTAX Counter32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Number of queries that were retransmitted that were to

```

        root servers."
    ::= { dnsResOptCounter 4 }

dnsResOptCounterInternals OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of requests internally generated by the
        resolver."
    ::= { dnsResOptCounter 5 }

dnsResOptCounterInternalTimeOuts OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "Number of requests internally generated which timed
        out."
    ::= { dnsResOptCounter 6 }

-- SNMPv2 groups.

dnsResMIBGroups          OBJECT IDENTIFIER ::= { dnsResMIB 2 }

dnsResConfigGroup OBJECT-GROUP
    OBJECTS { dnsResConfigImplementIdent,
              dnsResConfigService,
              dnsResConfigMaxCnames,
              dnsResConfigSbeltAddr,
              dnsResConfigSbeltName,
              dnsResConfigSbeltRecursion,
              dnsResConfigSbeltPref,
              dnsResConfigSbeltSubTree,
              dnsResConfigSbeltClass,
              dnsResConfigSbeltStatus,
              dnsResConfigUpTime,
              dnsResConfigResetTime }
    STATUS      current
    DESCRIPTION
        "A collection of objects providing basic configuration
        information for a DNS resolver implementation."
    ::= { dnsResMIBGroups 1 }

dnsResCounterGroup OBJECT-GROUP
    OBJECTS { dnsResCounterByOpcodeCode,
              dnsResCounterByOpcodeQueries,

```

```

        dnsResCounterByOpcodeResponses,
        dnsResCounterByRcodeCode,
        dnsResCounterByRcodeResponses,
        dnsResCounterNonAuthDataResps,
        dnsResCounterNonAuthNoDataResps,
        dnsResCounterMartians,
        dnsResCounterRecdResponses,
        dnsResCounterUnparseResps,
        dnsResCounterFallbacks }
STATUS      current
DESCRIPTION
    "A collection of objects providing basic instrumentation
    of a DNS resolver implementation."
 ::= { dnsResMIBGroups 2 }

dnsResLameDelegationGroup OBJECT-GROUP
OBJECTS     { dnsResLameDelegationOverflows,
               dnsResLameDelegationSource,
               dnsResLameDelegationName,
               dnsResLameDelegationClass,
               dnsResLameDelegationCounts,
               dnsResLameDelegationStatus }
STATUS      current
DESCRIPTION
    "A collection of objects providing instrumentation of
    'lame delegation' failures."
 ::= { dnsResMIBGroups 3 }

dnsResCacheGroup OBJECT-GROUP
OBJECTS     { dnsResCacheStatus,
               dnsResCacheMaxTTL,
               dnsResCacheGoodCaches,
               dnsResCacheBadCaches,
               dnsResCacheRRName,
               dnsResCacheRRClass,
               dnsResCacheRRType,
               dnsResCacheRRTTL,
               dnsResCacheRRElapsedTTL,
               dnsResCacheRRSource,
               dnsResCacheRRData,
               dnsResCacheRRStatus,
               dnsResCacheRRIndex,
               dnsResCacheRRPrettyName }
STATUS      current
DESCRIPTION
    "A collection of objects providing access to and control
    of a DNS resolver's cache."

```

```

 ::= { dnsResMIBGroups 4 }

dnsResNCCacheGroup OBJECT-GROUP
  OBJECTS { dnsResNCCacheStatus,
             dnsResNCCacheMaxTTL,
             dnsResNCCacheGoodNCaches,
             dnsResNCCacheBadNCaches,
             dnsResNCCacheErrQName,
             dnsResNCCacheErrQClass,
             dnsResNCCacheErrQType,
             dnsResNCCacheErrTTL,
             dnsResNCCacheErrElapsedTTL,
             dnsResNCCacheErrSource,
             dnsResNCCacheErrCode,
             dnsResNCCacheErrStatus,
             dnsResNCCacheErrIndex,
             dnsResNCCacheErrPrettyName }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing access to and control
    of a DNS resolver's negative response cache."
  ::= { dnsResMIBGroups 5 }

dnsResOptCounterGroup OBJECT-GROUP
  OBJECTS { dnsResOptCounterReferrals,
            dnsResOptCounterRetrans,
            dnsResOptCounterNoResponses,
            dnsResOptCounterRootRetrans,
            dnsResOptCounterInternals,
            dnsResOptCounterInternalTimeOuts }
  STATUS      current
  DESCRIPTION
    "A collection of objects providing further
    instrumentation applicable to many but not all DNS
    resolvers."
  ::= { dnsResMIBGroups 6 }

-- Compliances.

dnsResMIBCompliances OBJECT IDENTIFIER ::= { dnsResMIB 3 }

dnsResMIBCompliance MODULE-COMPLIANCE
  STATUS      current
  DESCRIPTION
    "The compliance statement for agents implementing the DNS
    resolver MIB extensions."
  MODULE -- This MIB module

```

```
MANDATORY-GROUPS { dnsResConfigGroup, dnsResCounterGroup }
GROUP      dnsResCacheGroup
DESCRIPTION
    "The resolver cache group is mandatory for resolvers that
    implement a cache."
GROUP      dnsResNCacheGroup
DESCRIPTION
    "The resolver negative cache group is mandatory for
    resolvers that implement a negative response cache."
GROUP      dnsResLameDelegationGroup
DESCRIPTION
    "The lame delegation group is unconditionally optional."
GROUP      dnsResOptCounterGroup
DESCRIPTION
    "The optional counters group is unconditionally
    optional."
OBJECT     dnsResConfigMaxCnames
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResConfigSbeltName
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResConfigSbeltRecursion
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResConfigSbeltPref
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResConfigReset
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResCacheStatus
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResCacheMaxTTL
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
OBJECT     dnsResNCacheStatus
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
```

```
OBJECT dnsResNCacheMaxTTL
MIN-ACCESS      read-only
DESCRIPTION
    "This object need not be writable."
 ::= { dnsResMIBCompliances 1 }
```

END

5. Acknowledgements

This document is the result of work undertaken the by DNS working group. The authors would particularly like to thank the following people for their contributions to this document: Philip Almquist, Frank Kastenholz (FTP Software), Joe Peck (DEC), Dave Perkins (SynOptics), Win Treese (DEC), and Mimi Zohar (IBM).

6. References

- [1] Mockapetris, P., "Domain Names -- Concepts and Facilities", STD 13, RFC 1034, USC/Information Sciences Institute, November 1987.
- [2] Mockapetris, P., "Domain Names -- Implementation and Specification", STD 13, RFC 1035, USC/Information Sciences Institute, November 1987.
- [3] Braden, R., Editor, "Requirements for Internet Hosts -- Application and Support, STD 3, RFC 1123, USC/Information Sciences Institute, October 1989.
- [4] Rose, M., and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based internets", STD 16, RFC 1155, Performance Systems International, Hughes LAN Systems, May 1990.
- [5] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets", RFC 1156, Hughes LAN Systems, Performance Systems International, May 1990.
- [6] Case, J., Fedor, M., Schoffstall, M., and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, SNMP Research, Performance Systems International, Performance Systems International, MIT Laboratory for Computer Science, May 1990.
- [7] Rose, M., and K. McCloghrie, Editors, "Concise MIB Definitions", STD 16, RFC 1212, Performance Systems International, Hughes LAN Systems, March 1991.

- [8] McCloghrie, K., and M. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II", STD 17, RFC 1213, Hughes LAN Systems, Performance Systems International, March 1991.
- [9] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Structure of Management Information for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1442, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [10] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Textual Conventions for version 2 of the the Simple Network Management Protocol (SNMPv2)", RFC 1443, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [11] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Conformance Statements for version 2 of the the Simple Network Management Protocol (SNMPv2)", RFC 1444, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [12] Galvin, J., and K. McCloghrie, "Administrative Model for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1445, Trusted Information Systems, Hughes LAN Systems, April 1993.
- [13] Case, J., McCloghrie, K., Rose, M., and S. Waldbusser, "Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1448, SNMP Research, Inc., Hughes LAN Systems, Dover Beach Consulting, Inc., Carnegie Mellon University, April 1993.
- [14] "Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1)", International Organization for Standardization, International Standard 8824, December 1987.

7. Security Considerations

Security issues are not discussed in this memo.

8. Authors' Addresses

Rob Austein
Epilogue Technology Corporation
268 Main Street, Suite 283
North Reading, MA 01864
USA

Phone: +1-617-245-0804
Fax: +1-617-245-8122
EMail: sra@epilogue.com

Jon Saperia
Digital Equipment Corporation
110 Spit Brook Road
ZK01-3/H18
Nashua, NH 03062-2698
USA

Phone: +1-603-881-0480
Fax: +1-603-881-0120
EMail: saperia@zko.dec.com

