TN3270E Working Group
INTERNET DRAFT: <draft-ietf-tn3270e-tn3270-mib-07.txt>
Expiration Date: November, 1998

Kenneth White
IBM Corp.
Robert Moore
IBM Corp.

May 1998

Base Definitions of Managed Objects for TN3270E Using SMIv2

<draft-ietf-tn3270e-tn3270-mib-07.txt>

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 Internet Draft. Distribution of this document is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1998). All Rights Reserved.

Abstract

This memo defines a Management Information Base (MIB) for configuring and managing TN3270E servers. TN3270E, defined by RFC 1647 [15], refers to the enhancements made to the Telnet 3270 (TN3270) terminal emulation practices. Refer to RFC 1041 [14], RFC 854 [12], and RFC 860 [13] for a sample of what is meant by TN3270 practices.

The MIB defined by this memo provides generic support for both host and gateway TN3270E server implementations. A TN3270E server connects a Telnet client performing 3270 emulation to a target SNA host over both a client-side network (client to TN3270E server) and an SNA Network (TN3270E server to target SNA host). The client-side network is

typically TCP/IP, but it need not be.

A host TN3270E server refers to an implementation where the TN3270E server is collocated with the Systems Network Architecture (SNA) System

Services Control Point (SSCP) for the dependent Secondary Logical Units (SLUs) that the server makes available to its clients for connecting into a SNA network. A gateway TN3270E server resides on an SNA node other than an SSCP, either an SNA type 2.0 node, a boundary-function-attached type 2.1 node, or an APPN node acting in the role of a Dependent LU Requester (DLUR). Host and gateway TN3270E server implementations typically differ greatly as to their internal implementation and system definition (SYSDEF) methods.

It is the intent that the MIB defined herein be extended by subsequent memos. For example, one such extension enables collection of TN3270E response time data. The specification of this MIB uses the Structure of Management Information (SMI) for Version 2 of the Simple Network Management Protocol (refer to RFC1902 [3]).

Table of Contents

1.0 Introduction 3

2.0 The SNMP Network Management Framework 3

3.0 Structure of the MIB 3
3.1 TN3270E Server Control 4
3.1.1 tn3270eSrvrConfTable 5
3.1.2 tn3270eSrvrPortTable 6
3.1.3 tn3270eSrvrStatsTable 6
3.2 TN3270E Server Resource Configuration 6
3.3 Resource Name / Client Address Mappings 7
3.3.1 tn3270eSnaMapTable 7
3.3.2 tn3270eResMapTable 7
3.3.3 tn3270eTcpConnTable 8
3.4 Advisory Spin Lock Usage 8
3.5 Row Persistence 8
3.6 IANA Considerations 9

4.0 Definitions 9

5.0 Security Considerations 45

6.0 Intellectual Property 46

7.0 Acknowledgments 47

8.0 References 47

9.0 Authors' Addresses 48

10.0 Full Copyright Statement 48

1.0 Introduction

This document is a product of the TN3270E Working Group. Its purpose is to define a MIB module for support by a TCP/IP implementation for configuration and management of TN3270E servers.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, reference [18].

2.0 The SNMP Network Management Framework

The SNMP Network Management Framework presently consists of six major components. They are:

- the overall architecture, described in RFC 2271 [7].
- the SMI, described in RFC 1902 [3], - the mechanisms used for describing and naming objects for the purpose of management.
- the MIB-II, STD 17, RFC 1213 [2], - the core set of managed objects for the Internet suite of protocols.
- the protocol, RFC 1157 [1] and/or RFC 1905 [6] and/or RFC 2272 [8] -- the protocol for accessing managed information.
- the user-based security model defined in RFC 2274 [10].
- the view-based access control model defined in RFC 2275 [11].

Textual conventions are defined in RFC 1903 [4], and conformance statements are defined in RFC 1904 [5]. Common applications are defined in RFC 2273 [9].

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

This memo specifies a MIB module that is compliant to the SMIv2. A MIB conforming to the SMIv1 can be produced through the appropriate translation.

3.0 Structure of the MIB

The TN3270E-MIB is split into the following components:

- TN3270E Server Control
- TN3270E Server Resource Configuration
- Resource/Client Address Mappings

There are three additional sections to address:

- Advisory Spin Lock Usage
- Row Persistence

o IANA Considerations

The TN3270E-MIB is defined primarily for TN3270E servers. This memo does not explicitly address use of the MIB by TN3270 servers that do not support the TN3270E protocol. Even though a significant number of the objects in the MIB do apply in the TN3270-only case, the case was not addressed, since it is unlikely that a TN3270-only server would implement this MIB.

The SYSAPPL-MIB, reference [20], contains the Utf8String textual convention (TC) that the TN3270E-MIB imports. This TC, which is used for some MIB objects containing textual information, enables internationalization of text strings, whereas the DisplayString TC does not. The SNMP-FRAMEWORK-MIB, reference [7], contains the SnmpAdminString TC that the TN3270E-MIB also imports. Like the Utf8String TC, this TC also enables internationalization of text strings; in addition, it provides some guidelines on the length and content of the strings.

It is important to note that implementation of the SYSAPPL-MIB is not actually a prerequisite for implementing the TN3270E-MIB. On the other hand, implementation of the TN3270E-MIB does not preclude implementing the SYSAPPL-MIB as well. When both MIBs are implemented, the primary index into most of the TN3270E-MIB tables, tn3270eSrvrConfIndex, SHOULD equal one of the SYSAPPL-MIB's sysApplElmtRunIndex values. In this case the entry in the sysApplElmtRunTable provides additional information on a TN3270E server.

The MIB defined by this memo supports use of both IPv4 and IPv6 addressing. Two textual conventions, IANATn3270eAddrType and Tn3270eTAddress, are defined for this purpose. IANATn3270eTAddress is essentially equivalent to the TAddress TC, defined by RFC 1903. The difference between the two is that IANATn3270eTAddress allows a zero-length octet string, while TAddress doesn't. It is important that IANATn3270eTAddress allow for the absence of an address, because some objects with this syntax are used as table indexes, and have special meanings when they contain zero-length strings.

The IANATn3270eAddrType textual convention is used rather than the TDomain TC (defined by RFC 1903) for identifying the contents of a tn3270eTAddress object. TDomain uses an OID to characterize the contents of an associated TAddress object. IANATn3270eAddrType was chosen over TDomain because, with a SYNTAX of Unsigned32 (enumeration type), it is much simpler to use as a component in an instance identifier. It was placed in the IANA-administered module to allow for the addition of values to cover cases (such as proxy servers) not covered by the TN3270E-MIB itself.

3.1 TN3270E Server Control

This group of objects provides for TN3270E server configuration and control. It consists of three tables:

- tn3270eSrvrConfTable
- tn3270eSrvrPortTable

o tn3270eSrvrStatsTable

The tn3270eSrvrConfTable is the primary table within the entire TN3270E-MIB. As section 3.1.1 indicates, each TN3270E server is represented by an entry in this table, indexed by tn3270eSrvrConfIndex. Most of the other tables defined by the TN3270E-MIB have tn3270eSrvrConfIndex as their primary index. Entries in these tables MUST NOT exist for a TN3270E server when it does not have a tn3270eSrvrConfigEntry.

3.1.1 tn3270eSrvrConfTable

The tn3270eSrvrConfTable contains a set of objects primarily used for configuring and managing TN3270E servers. As with most of the other tables in the TN3270E-MIB, this table is indexed by an unsigned integer, tn3270eSrvrConfIndex. This primary index element enables support of multiple TN3270E servers by a single SNMP agent. Within the set of MIB objects returned by one SNMP agent, tn3270eSrvrConfIndex values MUST be unique, and need not be contiguous.

The tn3270eSrvrConfInactivityTimer object defines the inactivity period for user traffic on TN3270 and TN3270E sessions.

The four objects:

o tn3270eSrvrConfActivityCheck
o tn3270eSrvrConfTmNopInterval
o tn3270eSrvrConfTmNopInactTime
o tn3270eSrvrConfTmTimeout

define the parameters for performing the "Telnet Timing Mark Option" as defined by RFC 860 [13]. The object tn3270eSrvrConfActivityCheck allows a Management Station to select either a NOP command or a TIMING-MARK command. Sending a NOP command results in less overhead then a TIMING-MARK command, since a client doesn't send a reply.

The objects tn3270eSrvrConfAdminStatus and tn3270eSrvrConfOperStatus enable remote starting and stopping of a TN3270E server, and report the current state of the server. The object tn3270eSrvrConfFunctionsSupported indicates which of the TN3270 and TN3270E options a server supports. The object tn3270eSrvrConfSessionTermState defines as a TN3270E server-wide option what SHOULD occur when the SNA portion of a TN3270 or TN3270E session terminates with respect to the associated TCP connection. The object tn3270eSrvrConfSrvrType indicates whether the TN3270E server represented by a tn3270eSrvrConfEntry is a host or a gateway server. The object tn3270eSrvrConfContact provides a scratch pad area for a TN3270E server administrator to store information for later retrieval. The object tn3270eSrvrConfLastActTime reports the DateAndTime when the server was

most recently activated. The special value of all '00'Hs indicates that the server has never been active.

The object tn3270eSrvrConfRowStatus provides the capability to perform row creation and deletion operations on this table.

3.1.2 tn3270eSrvrPortTable

The tn3270eSrvrPortTable represents the local TCP ports associated with a TN3270E server. This information is important because some TN3270E server implementations support usage of multiple local ports. A tn3270eSrvrPortEntry is indexed by:

- tn3270eSrvrConfIndex
- tn3270eSrvrConfPort
- tn3270eSrvrConfPortAddrType
- tn3270eSrvrConfPortAddress

Certain TN3270E server implementations restrict a local TCP port to a particular local IP address, instead of allowing connections for any local IP address to occur via the port. tn3270eSrvrConfPortAddrType and tn3270eSrvrConfPortAddress allow this restriction to be represented in the MIB. A TN3270E server that doesn't restrict connections over a port to a local IP Address SHALL use the value unknown(0) for tn3270eSrvrConfPortAddrType, and a zero-length octet string for tn3270eSrvrConfPortAddress.

3.1.3 tn3270eSrvrStatsTable

The tn3270eSrvrStatsTable defines a series of objects that provide general usage statistics for a TN3270E server. An entry can represent the total activity for a server, or it can represent the activity occurring at the server on either a port or a port-and-local-address basis.

An implementation of this table MUST use only one of the three levels of refinement that the indexing of this table supports for the entries associated with a single TN3270E server.

The objects in this table reporting maximum, in-use, and spare LUs for terminals and printers presuppose an implementation where terminal resources and printer resources come from disjoint, dedicated pools. An implementation where resources for the two types of LUs come from a single shared pool should return the following values:

- maximum: maximum size of the shared pool
- in-use: number currently in use as this type of LU
- spare: maximum - (terminal in-use + printer in-use)

3.2 TN3270E Server Resource Configuration

The following three tables provide for configuration of resources at a TN3270E server:

- tn3270eClientGroupTable
- tn3270eResPoolTable

o   tn3270eClientResMapTable

tn3270eClientGroupTable and tn3270eResPoolTable enable implementations to define groupings of both client addresses and resource pools for

mapping client addresses to resources. The tn3270eClientResMapTable provides a mapping from a client group to a resource pool.

3.3 Resource Name / Client Address Mappings

The TN3270E-MIB contains three tables for mapping resource names to client addresses, and client addresses to resource names:

- tn3270eSnaMapTable
- tn3270eResMapTable
- tn3270eTcpConnTable

3.3.1 tn3270eSnaMapTable

The tn3270eSnaMapTable is a read-only table that maps a secondary LU's SNA network name to the name by which it is known locally at the TN3270E server. For correlation with data from the SNA network, the name of the associated primary LU also appears in a tn3270eSnaMapEntry. An entry in this table is created when the Activate LU (ACTLU) request carrying the SNA network name of the SLU is received from the SSCP. The entry is deleted when the SLU is deactivated.

A TN3270E server provides a client with access to an SNA application by associating a TCP connection from the client with an SNA secondary LU (SLU) at the server. This SLU in turn has an SNA session with a primary LU (PLU) running on an SNA host. This PLU represents the application with which the client is communicating. The TN3270E-MIB includes two tables for mapping back and forth among the SNA name identifying the PLU, the SNA name identifying the SLU, and the TCP connection with the client.

In order to understand how these name mappings work, it is necessary to understand a subtlety involving the names of the SLUs at the TN3270E server: these names are often different from the names by which the SLUs are known in the rest of the SNA network. In the TN3270E-MIB, these two types of SLU names are termed "local names" and "SSCP-supplied names"; the latter term indicates that the name by which the SLU is known in the SNA network comes to the TN3270E server from the SNA System Services Control Point.

SSCPs don't always send SLU names down to secondary LUs; in some cases this capability must be turned on. In the case of SLUs served by a Dependent LU Requester (DLUR), an SSCP always sends SLU names to the DLUR. It is necessary, however, to enable the DLUR's PU/LU Network Name Forwarding function, so that it forwards the SLU names it receives from the SSCP down to the PUs that it serves.

For SLUs associated with an SNA type 2.0 node (or with a boundary-function-attached type 2.1 node) not served by a DLUR, inclusion of SLU names on ACTLU must be enabled explicitly at the SSCP

via local configuration.

3.3.2 tn3270eResMapTable

The tn3270eResMapTable is a read-only table that maps a resource name to a client's address. An entry in this table is created when a TCP connection is received by a TN3270E server and mapped to a resource. The entry is deleted when the resource-to-address association is no longer valid.

3.3.3 tn3270eTcpConnTable

The TCP Connection Table is currently defined by RFC 2012 (Refer to reference [16], TCP-MIB Definitions). It contains the following objects:

- tcpConnState (INTEGER)
- tcpConnLocalAddress (IpAddress)
- tcpConnLocalPort (INTEGER)
- tcpConnRemAddress (IpAddress)
- tcpConnRemPort (INTEGER)

It is indexed by: tcpConnLocalAddress, tcpConnLocalPort, tcpConnRemAddress, and tcpConnRemPort.

The tn3270eTcpConnTable contains objects for keeping a list of the current set of TN3270 and TN3270E sessions at a TN3270E server. The relationship between the tcpConnTable and the Tn3270eTcpConnTable is not one-to-one, since the tn3270eTcpConnTable contains information pertaining only to TN3270(E) sessions.

The tn3270eTcpConnTable has a different indexing structure from that of the tcpConnTable. Instead of using IpAddress objects, Tn3270eTAddress and IANATn3270eAddrType object pairs are used to specify client addresses (both local and remote). This enables support of IPv6 addresses. In addition, the remote address pair precedes the local address pair in the index clause, in order to enable a GET-NEXT operation using only the remote address pair.

3.4 Advisory Spin Lock Usage

Within the TN3270E-MIB, tn3270eConfSpinLock is defined as an advisory lock that allows cooperating TN3270E-MIB applications to coordinate their use of the tn3270eSrvrConfTable, the tn3270eSrvrPortTable, the tn3270eClientGroupTable, the tn3270eResPoolTable, and the tn3270eClientResMapTable. When creating a new entry or altering an existing entry in any of these tables, an application SHOULD make use of tn3270eConfSpinLock to serialize application changes or additions. Since this is an advisory lock, its use by management applications SHALL NOT be enforced by agents. Agents MUST, however, implement the tn3270eConfSpinLock object.

3.5 Row Persistence

The following tables enable remote creation of their entries by including RowStatus objects:

- tn3270eSrvrConfTable
- tn3270eSrvrPortTable

- tn3270eClientGroupTable
- tn3270eResPoolTable
- tn3270eClientResMapTable

An implementation SHOULD NOT retain SNMP-created entries in these tables across reIPLs (Initial Program Loads) of the corresponding TN3270E server, since management applications need to see consistent behavior with respect to the persistence of the table entries that they create.

It is expected that local, implementation-dependent configuration information will be used to define the initial and persistent configurations for TN3270E server usage. Thus it is not necessary to enable persistence of table entries by adding StorageType (refer to RFC 1903 [4]) objects to these tables.

3.6 IANA Considerations

The tn3270eSrvrFunctionsSupported, tn3270eTcpConnFunctions, tn3270eTcpConnClientIdFormat, and tn3270eTcpConnLogInfo objects, as well as a number of objects identifying the types of various addresses, use textual conventions imported from the IANATn3270eTC-MIB. The purpose of defining these textual conventions in a separate MIB module is to allow additional values to be defined without having to issue a new version of this document. The Internet Assigned Number Authority (IANA) is responsible for the assignment of all Internet numbers, including various SNMP-related numbers; it will administer the values associated with these textual conventions.

The rules for additions or changes to the IANATn3270eTC-MIB are outlined in the DESCRIPTION clause associated with its MODULE-IDENTITY statement.

The current version of the IANATn3270eTC-MIB can be accessed from the IANA home page at: "http://www.iana.org/iana/".

4.0 Definitions

```
  TN3270E-MIB DEFINITIONS ::= BEGIN

  IMPORTS
      MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, TimeTicks,
      IpAddress, Counter32, Gauge32, Counter64
          FROM SNMPv2-SMI
      TEXTUAL-CONVENTION, RowStatus, TestAndIncr, DateAndTime,
      TimeStamp
          FROM SNMPv2-TC
      MODULE-COMPLIANCE, OBJECT-GROUP
          FROM SNMPv2-CONF
      snanauMIB
          FROM SNA-NAU-MIB
```

```
      Utf8String
          FROM SYSAPPL-MIB
      SnmpAdminString
          FROM SNMP-FRAMEWORK-MIB
      IANATn3270eAddrType, IANATn3270eClientType,
```

```
      IANATn3270eLogData, IANATn3270Functions
          FROM IANATn3270eTC-MIB;

    tn3270eMIB MODULE-IDENTITY
        LAST-UPDATED "9805120000Z" -- May 12, 1998
        ORGANIZATION "TN3270E Working Group"
        CONTACT-INFO
            "Kenneth White (kennethw@vnet.ibm.com)
             IBM Corp. - Dept. BRQA/Bldg. 501/G114
             P.O. Box 12195
             3039 Cornwallis
             RTP, NC 27709-2195
             USA

             Robert Moore (remoore@us.ibm.com)
             IBM Corp. - Dept. BRQA/Bldg. 501/G114
             P.O. Box 12195
             3039 Cornwallis
             RTP, NC 27709-2195
             USA
             +1-919-254-4436"
        DESCRIPTION
            "This module defines a portion of the management
            information base (MIB) for managing TN3270E servers."
        ::= { snanauMIB 8 }

  -- Textual Conventions

  SnaResourceName ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
          "The textual convention for defining an SNA resource
           name. A fully qualified SNA resource name, consisting
           of a 1 to 8 character network identifier (NetId), a
           period ('.'), and a 1 to 8 character resource name
           (ResName).

           The NetId and ResName are constructed from the
           uppercase letters 'A' - 'Z' and the numerics '0' - '9',
           all encoded in ASCII, with the restriction that the
           first character of each must be a letter.  Blanks are
           not allowed.

           Earlier versions of SNA permitted three additional
           characters in NetIds and ResNames:  '#', '@', and '$'.
           While this use of these characters has been retired,
           a Management Station should still accept them for
           backward compatibility.
```

```
            Note: This Textual Convention is not subject to
            internationalization, and does not use the character
            encodings used by the Utf8String Textual Convention."
        SYNTAX     OCTET STRING (SIZE(0..17))

    Tn3270ResourceType ::= TEXTUAL-CONVENTION
```

```
      STATUS      current
      DESCRIPTION
          "The type of resource defined by a resource pool.  Refer
          to tn3270eResPoolTable."
      SYNTAX      INTEGER {
                            other(0),
                            terminal(1),
                            printer(2),
                            terminalOrPrinter(3)
                          }

  Tn3270eTAddress ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
            "Denotes a client address.  The type of client
            address is determined by use of the IANATn3270eAddrType
            textual convention.  The length in octets of a
            Tn3270eTAddress object is:

            IANATn3270eAddrType       Address Length
                 unknown(0)       not specified or unknown;
                                  the actual length of the
                                  Tn3270eTAddress octet string
                                  indicates if an address
                                  is present
                   ipv4(1)        4 OCTETS
                   ipv6(2)        16 OCTETS

            This textual convention is similar to the TAddress
            TC defined by RFC1903 except that it allows a
            zero-length octet string and is not a full transport
            layer address."
      SYNTAX       OCTET STRING (SIZE (0..255))

  Tn3270DeviceTypes ::= TEXTUAL-CONVENTION
      STATUS      current
      DESCRIPTION
          "This textual convention defines the list of device
          types that can be set, as defined by RFC 1647."
      SYNTAX      INTEGER { -- terminals
                    unknown(0),
                    ibm3278d2(1),    -- (24 row x 80 col display)
                    ibm3278d2E(2),   -- (24 row x 80 col display)
                    ibm3278d3(3),    -- (32 row x 80 col display)
                    ibm3278d3E(4),   -- (32 row x 80 col display)
                    ibm3278d4(5),    -- (43 row x 80 col display)
                    ibm3278d4E(6),   -- (43 row x 80 col display)
                    ibm3278d5(7),    -- (27 row x 132 col display)
                    ibm3278d5E(8),   -- (27 row x 132 col display)
```

```
                    ibmDynamic(9),   -- (no pre-defined display size)
                    ibm3287d1(10)    -- printers
            }

  Tn3270eTraceData ::= TEXTUAL-CONVENTION
      STATUS      current
```

```
      DESCRIPTION
          "An octet string representing trace data from the
          Telnet half of a TN3270E session, from the SNA half,
          or from both.  The octet string contains a sequence
          of trace elements, with the trace elements in the
          string ordered from earliest to latest.

          Each trace element has the following form:

                  +---+---+----+----------------------+
                  !length !type!data                  !
                  +---+---+----+----------------------+

            where:

            length = two-octet length of the data portion of the
                     trace element, not including the length and
                     type octets

            type   = one-octet code point characterizing the data;
                     defined values are:

                     X'01' telnet PDU from the server to the client
                     X'02' telnet PDU from the client to the server
                     X'03' SNA data from the server to the SNA host
                     X'04' SNA data from the SNA host to the server

            data   = initial part of a PDU.

          It is left to implementations to determine how
          much of each PDU to return in a trace element.

          The zero-length string indicates that no trace
          data is available."
      SYNTAX OCTET STRING (SIZE (0 | 3..4096))

  -- Top-level structure of the MIB

  tn3270eNotifications  OBJECT IDENTIFIER ::= { tn3270eMIB 0 }
  tn3270eObjects        OBJECT IDENTIFIER ::= { tn3270eMIB 1 }
  tn3270eConformance    OBJECT IDENTIFIER ::= { tn3270eMIB 3 }

  -- MIB Objects

  tn3270eSrvrConfTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eSrvrConfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table defines the configuration elements for
```

TN3270E servers. The number of entries in this table is expected to vary depending on the location of the table. A particular TN3270E server is expected to have a single entry. Modeling of the configuration elements as a table allows multiple TN3270E

```
          servers to be serviced by the same SNMP agent.

          An implementation SHOULD NOT retain an SNMP-created
          entry in this table across re-IPLs (Initial Program
          Loads) of the corresponding TN3270E server."

      ::= { tn3270eObjects 1 }

  tn3270eSrvrConfEntry OBJECT-TYPE
      SYNTAX      Tn3270eSrvrConfEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of the configuration elements for a single
          TN3270E server."
      INDEX       { tn3270eSrvrConfIndex }
      ::= { tn3270eSrvrConfTable 1 }

  Tn3270eSrvrConfEntry ::= SEQUENCE {
      tn3270eSrvrConfIndex             Unsigned32,
      tn3270eSrvrConfInactivityTimeout Unsigned32,
      tn3270eSrvrConfActivityCheck     INTEGER,
      tn3270eSrvrConfTmNopInactTime    Unsigned32,
      tn3270eSrvrConfTmNopInterval     Unsigned32,
      tn3270eSrvrFunctionsSupported    IANATn3270Functions,
      tn3270eSrvrConfAdminStatus       INTEGER,
      tn3270eSrvrConfOperStatus        INTEGER,
      tn3270eSrvrConfSessionTermState  INTEGER,
      tn3270eSrvrConfSrvrType          INTEGER,
      tn3270eSrvrConfContact           SnmpAdminString,
      tn3270eSrvrConfRowStatus         RowStatus,
      tn3270eSrvrConfLastActTime       DateAndTime,
      tn3270eSrvrConfTmTimeout         Unsigned32
    }

  tn3270eSrvrConfIndex  OBJECT-TYPE
      SYNTAX      Unsigned32 (1..4294967295)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "Identifier for a single TN3270E server.

         tn3270eSrvrConfIndex values need not be
         contiguous."
      ::= { tn3270eSrvrConfEntry 1 }

  tn3270eSrvrConfInactivityTimeout OBJECT-TYPE
      SYNTAX      Unsigned32 (0..99999999)
      UNITS "seconds"
```

```
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "The inactivity time-out specified in seconds.  When a
          connection has been inactive for the number of seconds
          specified by this object it is closed.  Only user traffic
```

```
          is considered when determining whether there has been
          activity on a connection.

          The default value 0 means that no inactivity time-out is
          in effect."
      DEFVAL { 0 }
      ::= { tn3270eSrvrConfEntry 2 }

  tn3270eSrvrConfActivityCheck OBJECT-TYPE
      SYNTAX      INTEGER {
                            noCheck(0),
                            timingMark(1),
                            nop(2)
                          }
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object enables TIMING-MARK processing, NOP
          processing, or neither for a TN3270E server."
      DEFVAL { noCheck }
      ::= { tn3270eSrvrConfEntry 3 }

  tn3270eSrvrConfTmNopInactTime OBJECT-TYPE
      SYNTAX      Unsigned32 (1..86400) -- 1 second to 24 hours
      UNITS "seconds"
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "The amount of time a connection must have had no
         traffic on it in order for a TIMING-MARK or NOP request
         to be sent on the connection.  This value applies only
         when connections are being examined for recent activity
         on a scan interval controlled by the value of the
         tn3270eSrvrConfTmNopInterval object."
      DEFVAL { 600 }   -- 10 minutes
      ::= { tn3270eSrvrConfEntry 4 }

  tn3270eSrvrConfTmNopInterval OBJECT-TYPE
      SYNTAX      Unsigned32 (1..86400) -- 1 second to 24 hours
      UNITS "seconds"
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "The scan interval to be used by a TN3270E server when
         it examines its Telnet connections for recent activity.
         The server scans its Telnet connections on the interval
         provided by this object, looking for ones that have been
         idle for more than the value provided by the
         tn3270eSrvrConfTmNopInactTime object.  A TIMING-MARK or
```

```
        NOP request is sent for each connection that has
        exhibited no activity for this period of time."
    DEFVAL { 120 }   -- 2 minutes
    ::= { tn3270eSrvrConfEntry 5 }

  tn3270eSrvrFunctionsSupported OBJECT-TYPE
```

```
      SYNTAX    IANATn3270Functions
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "This object indicates the functions supported by a
          TN3270E server."
      DEFVAL { { scsCtlCodes, dataStreamCtl,
                 responses, bindImage, sysreq } }
      ::= { tn3270eSrvrConfEntry 6 }

  tn3270eSrvrConfAdminStatus OBJECT-TYPE
      SYNTAX  INTEGER {
                        up(1),
                        down(2),
                        stopImmediate(3)
                      }
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "The desired state of the TN3270E server represented
           by this entry in the table:

           up(1)            - Activate this TN3270E server.
           down(2)          - Informs the associated TN3270E
                              server to gracefully terminate
                              its processing.
           stopImmediate(3) - Informs the associated TN3270E
                              server to terminate itself
                              immediately.

           When a managed system creates an entry in this table,
           tn3270eSrvrConfAdminStatus and tn3270eSrvrConfOperStatus
           are initialized as up(1) by default.

           The exact behavior of a server in response to a down(2)
           or stopImmediate(3) command is left implementation-
           dependent.  A TN3270E server that is capable of it
           SHOULD close all of its TN3270 and TN3270E sessions
           during a graceful termination.

           Often the function enabled via stopImmediate(3) is used
           as a last resort by a system administrator, to attempt
           to either bring down a hung TN3270E server or free up
           its resources immediately to aid in general system
           availability, or to shut down a TN3270E server that is
           not recognizing a down(2) request.

           A TN3270E server that does not distinguish between
           down(2) or stopImmediate(3) transitions should not
```

```
           support stopImmediate(3)."
       DEFVAL { up }
       ::= { tn3270eSrvrConfEntry 7 }

   tn3270eSrvrConfOperStatus OBJECT-TYPE
       SYNTAX  INTEGER {
```

```
                        up(1),
                        down(2),
                        busy(3),
                        shuttingDown(4)
                       }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "The current operational state of a TN3270E server.
         The following values are defined:

          up(1)            - the server is active and accepting
                             new client connections
          down(2)          - the server is not active
          busy(3)          - the server is active, but is not
                             accepting new client connections
                             because it lacks the resources to
                             do so
          shuttingDown(4)  - the server is active, but is not
                             accepting new client connections
                             because it is in the process of
                             performing a graceful shutdown."
      DEFVAL { up }
      ::= { tn3270eSrvrConfEntry 8 }

  tn3270eSrvrConfSessionTermState OBJECT-TYPE
      SYNTAX  INTEGER {
                        terminate(1),
                        luSessionPend(2),
                        queueSession(3)
                       }
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
         "This object determines what a TN3270E server
          should do when a TN3270 Session terminates:

         terminate(1)     => Terminate the TCP connection.
         luSessionPend(2) => Do not drop the TCP connection
                             associated with a client when its
                             TN3270 session ends.  Processing
                             should redrive session initialization
                             as if the client were first connecting.
         queueSession(3)  => This value relates to the Close
                             Destination PASS (CLSDST PASS) operation
                             in VTAM.  An example provides the
                             easiest explanation.  Suppose a TN3270E
                             client is in session with APPL1, and
                             APPL1 does a CLSDST PASS of the client's
```

```
                        session to APPL2.  queueSession(3)
                        specifies that the TN3270E server must
                        keep the TCP connection with the client
                        active after it receives the UNBIND from
                        APPL1, waiting for the BIND from APPL2."
      DEFVAL { terminate }
```

```
      ::= { tn3270eSrvrConfEntry 9 }

  tn3270eSrvrConfSrvrType OBJECT-TYPE
      SYNTAX      INTEGER {
                            host(1),
                            gateway(2)
                          }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "This object indicates the type of TN3270E server.
          The existence of MIB tables and objects that will be
          defined by follow-on MIBs may be predicated on whether
          the TN3270E server can be local to the same host as a
          target application (host(1)) or will always be remote
          (gateway(2)).

          A host TN3270E server refers to an implementation where
          the TN3270E server is collocated with the Systems
          Network Architecture (SNA) System Services Control Point
          (SSCP) for the dependent Secondary Logical Units (SLUs)
          that the server makes available to its clients for
          connecting into an SNA network.

          A gateway TN3270E server resides on an SNA node other
          than an SSCP, either an SNA type 2.0 node or an APPN node
          acting in the role of a Dependent LU Requester (DLUR).

          Host and gateway TN3270E server implementations typically
          differ greatly as to their internal implementation and
          system definition (SYSDEF) requirements."
      ::= { tn3270eSrvrConfEntry 10 }

  tn3270eSrvrConfContact OBJECT-TYPE
      SYNTAX      SnmpAdminString
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "This object provides a scratch pad for a TN3270E
          server administrator for storing information for
          later retrieval."
      DEFVAL { ''H }   -- the empty string
      ::= { tn3270eSrvrConfEntry 11 }

  tn3270eSrvrConfRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
```

"This object allows entries to be created and deleted
in the tn3270eSrvrConfTable. Entries may also be
created and deleted as a result of implementation-
dependent operations.

With the exception of tn3270eSrvrConfSrvrType, which

```
            an implementation can easily fill in for itself, all
            the columnar objects in this table have DEFVALs
            associated with them.  Consequently, a Management
            Station can create a conceptual row via a SET
            operation that specifies a value only for this object.

            When a tn3270eSrvrConfEntry is deleted (by setting
            this object to destroy(6)), this has the side-effect
            of removing all the associated entries (i.e., those
            having the same tn3270eSrvrConfIndex) from the
            tn3270eSrvrPortTable, the tn3270eSrvrStatsTable, the
            tn3270eClientGroupTable, the tn3270eResPoolTable,
            the tn3270eSnaMapTable, the tn3270eClientResMapTable,
            and the tn3270eResMapTable.  All entries in the
            tn3270eTcpConnTable that belong to a TN3270E server
            that has been deleted MUST also be removed.

            In other words, a tn3270eSrvrConfEntry must exist for
            a TN3270E server in order for it to have entries in
            any of the other tables defined by this MIB."
        REFERENCE
            "RFC 1903, 'Textual Conventions for version 2 of the
            Simple Network Management Protocol (SNMPv2).'"
        ::= { tn3270eSrvrConfEntry 12 }

    tn3270eSrvrConfLastActTime OBJECT-TYPE
        SYNTAX      DateAndTime
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object reports the DateAndTime when a TN3270E
            server was most recently activated.

            The special value of all '00'Hs indicates that the
            server has never been active, i.e., that the value of
            tn3270eSrvrOperStatus has never been anything other
            than down(2)."
        DEFVAL { '0000000000000000'H }
        ::= { tn3270eSrvrConfEntry 13 }

    tn3270eSrvrConfTmTimeout OBJECT-TYPE
        SYNTAX      Unsigned32 (1..600) -- 1 second to 10 minutes
        UNITS "seconds"
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
           "The TIMING-MARK time-out, specified in seconds."
        DEFVAL { 5 }   -- 5 seconds
        ::= { tn3270eSrvrConfEntry 14 }
```

```
  tn3270eSrvrPortTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eSrvrPortEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
```

```
          "This table defines the TCP ports associated with
          TN3270E servers.  No entry in this table shall exist
          without a corresponding (same tn3270eSrvrConfIndex)
          entry in the tn3270eSrvrConfTable existing.

          An implementation SHOULD NOT retain SNMP-created
          entries in this table across re-IPLs (Initial Program
          Loads) of the corresponding TN3270E server."
      ::= { tn3270eObjects 2 }

  tn3270eSrvrPortEntry OBJECT-TYPE
      SYNTAX      Tn3270eSrvrPortEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of a single TCP port assignment to a
           TN3270E server.  Assignment of a port on a local
           address basis is enabled though use of
           tn3270eSrvrPortAddrType and tn3270eSrvrPortAddress.

           A TCP port assignment that is not restricted to
           a local address SHALL specify a tn3270eSrvrPortAddrType
           of unknown(0), and SHALL use a zero-length octet string
           for the tn3270eSrvrPortAddress."
      INDEX       {
                    tn3270eSrvrConfIndex,
                    tn3270eSrvrPort,
                    tn3270eSrvrPortAddrType,
                    tn3270eSrvrPortAddress
                  }
      ::= { tn3270eSrvrPortTable 1 }

  Tn3270eSrvrPortEntry ::= SEQUENCE {
      tn3270eSrvrPort                 Unsigned32,
      tn3270eSrvrPortAddrType         IANATn3270eAddrType,
      tn3270eSrvrPortAddress          Tn3270eTAddress,
      tn3270eSrvrPortRowStatus        RowStatus
    }

  tn3270eSrvrPort OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65535)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
         "Indicates a port assigned to a server."
      ::= { tn3270eSrvrPortEntry 1 }

  tn3270eSrvrPortAddrType OBJECT-TYPE
      SYNTAX      IANATn3270eAddrType
```

```
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Indicates the type of an address local to the host on
          which the TN3270E server resides that is represented
          in tn3270eSrvrPortAddress.  A value of unknown(0)
```

```
          SHALL be used for this object when the port is not
          to be restricted to a local address."
      ::= { tn3270eSrvrPortEntry 2 }

  tn3270eSrvrPortAddress OBJECT-TYPE
      SYNTAX      Tn3270eTAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A local address on the host that a TN3270E server
           resides on that is associated with a TCP port that
           is to be used or is in use by a TN3270E server.
           tn3270eClientGroupAddrType indicates the
           address type (IPv4 or IPv6, for example).

           A zero-length octet string SHALL be used as the
           value of this object when a local address isn't
           being specified."
      ::= { tn3270eSrvrPortEntry 3 }

  tn3270eSrvrPortRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "This object allows entries to be created and deleted
          in the tn3270eSrvrPortTable.  Entries may also be
          created and deleted as a result of implementation-
          dependent operations.

          Since this is the only accessible object in this table,
          a Management Station can create a conceptual row via a SET
          operation that specifies a value only for this object.

          An entry in this table is deleted by setting this object
          to destroy(6).  Deletion of a tn3270eSrvrPortEntry has
          no effect on any other table entry defined by this MIB."
      REFERENCE
          "RFC 1903, 'Textual Conventions for version 2 of the
          Simple Network Management Protocol (SNMPv2).'"
      ::= { tn3270eSrvrPortEntry 4 }

  tn3270eSrvrStatsTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eSrvrStatsEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table defines a set of statistics concerning
          TN3270E server performance.
```

```
          No entry in this table shall exist without
          a corresponding (same tn3270eSrvrConfIndex) entry in
          the tn3270eSrvrConfTable existing."
      ::= { tn3270eObjects 3 }
```

```
  tn3270eSrvrStatsEntry OBJECT-TYPE
      SYNTAX      Tn3270eSrvrStatsEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A collection of statistical and maximum usage objects
          for a single TN3270 server.  An entry can represent the total
          activity of the server, or it can represent the activity
          occurring at the server on either a port or a
          port-and-local-address basis.

          Collection of the statistics represented by the objects
          in this table is not mandatory.   An implementation
          of this table MUST use only one of the three levels of
          refinement that this table supports for the entries
          associated with each TN3270E server.

          The indexing for a row that represents total server
          statistics is as follows:

               tn3270eSrvrConfIndex      value identifying the server
               tn3270eSrvrPort           0
               tn3270eSrvrPortAddrType   unknown(0)
               tn3270eSrvrPortAddress    zero-length octet string.

          On a port basis:

               tn3270eSrvrConfIndex      value identifying the server
               tn3270eSrvrPort           > 0
               tn3270eSrvrPortAddrType   unknown(0)
               tn3270eSrvrPortAddress    zero-length octet string.

          On a port-and-local-address basis:

               tn3270eSrvrConfIndex      value identifying the server
               tn3270eSrvrPort           > 0
               tn3270eSrvrPortAddrType   valid value other than unknown(0)
               tn3270eSrvrPortAddress    non-zero-length octet string.

         "
      INDEX       {
                    tn3270eSrvrConfIndex,
                    tn3270eSrvrPort,
                    tn3270eSrvrPortAddrType,
                    tn3270eSrvrPortAddress
                  }
      ::= { tn3270eSrvrStatsTable 1 }

  Tn3270eSrvrStatsEntry ::= SEQUENCE {
      tn3270eSrvrStatsUpTime          TimeStamp,
```

```
      tn3270eSrvrStatsMaxTerms        Unsigned32,
      tn3270eSrvrStatsInUseTerms      Gauge32,
      tn3270eSrvrStatsSpareTerms      Gauge32,
      tn3270eSrvrStatsMaxPtrs         Unsigned32,
      tn3270eSrvrStatsInUsePtrs       Gauge32,
```

```
      tn3270eSrvrStatsSparePtrs       Gauge32,
      tn3270eSrvrStatsInConnects      Counter32,
      tn3270eSrvrStatsConnResrceRejs  Counter32,
      tn3270eSrvrStatsDisconnects     Counter32,
      tn3270eSrvrStatsHCInOctets      Counter64,
      tn3270eSrvrStatsInOctets        Counter32,
      tn3270eSrvrStatsHCOutOctets     Counter64,
      tn3270eSrvrStatsOutOctets       Counter32,
      tn3270eSrvrStatsConnErrorRejs   Counter32
    }

  tn3270eSrvrStatsUpTime OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The value of the sysUpTime object the last time
          the TN3270E server was re-initialized.

          Server re-initialization is the only discontinuity
          event for the counters in this table.  Even if table
          entries are on a port or port-and-local-address
          basis, port deactivation and reactivation do not
          result in counter discontinuities."
      ::= { tn3270eSrvrStatsEntry 2 }

  tn3270eSrvrStatsMaxTerms OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "LUs"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the maximum number of terminal LUs available
         for use at a TN3270E server for the granularity of this
         conceptual row (server-wide, port, or
         port-and-local-address)."
      ::= { tn3270eSrvrStatsEntry 3 }

  tn3270eSrvrStatsInUseTerms OBJECT-TYPE
      SYNTAX      Gauge32
      UNITS       "LUs"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of terminal LUs currently in
         use at a TN3270E server for the granularity of this
         conceptual row (server-wide, port, or
         port-and-local-address)."
      ::= { tn3270eSrvrStatsEntry 4 }
```

```
  tn3270eSrvrStatsSpareTerms OBJECT-TYPE
      SYNTAX      Gauge32
      UNITS       "LUs"
      MAX-ACCESS  read-only
      STATUS      current
```

```
      DESCRIPTION
         "Indicates the number of free terminal LUs at a TN3270E
         server for the granularity of this conceptual row
         (server-wide, port, or port-and-local-address).

         It is possible that the difference between
         tn3270eSrvrStatsMaxTerms and tn3270eSrvrStatsInUseTerms
         in a conceptual row does not equal the value of
         tn3270eSrvrStatsSpareTerms in that row:  an LU may
         exist but not be usable by a client connection.

         Alternatively, the administrative ceiling represented
         by tn3270eSrvrStatsMaxTerms may have been lowered to
         a point where it is less than the current value of
         tn3270eSrvrStatsInUseTerms.  In this case
         tn3270eSrvrStatsSpareTerms returns the value 0."

      ::= { tn3270eSrvrStatsEntry 5 }

  tn3270eSrvrStatsMaxPtrs OBJECT-TYPE
      SYNTAX      Unsigned32
      UNITS       "Printer Resources"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the maximum number of printer resources
         available for use by a TN3270E server for the
         granularity of this conceptual row (server-wide,
         port, or port-and-local-address)."
      ::= { tn3270eSrvrStatsEntry 6 }

  tn3270eSrvrStatsInUsePtrs OBJECT-TYPE
      SYNTAX      Gauge32
      UNITS       "Printer Resources"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of printer resources
         currently in use by a TN3270E server for the
         granularity of this conceptual row (server-wide,
         port, or port-and-local-address)."
      ::= { tn3270eSrvrStatsEntry 7 }

  tn3270eSrvrStatsSparePtrs OBJECT-TYPE
      SYNTAX      Gauge32
      UNITS       "Spare Printer Resources"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

"Indicates the number of free printer resources at
a TN3270E server for the granularity of this conceptual
row (server-wide, port, or port-and-local-address).

It is possible that the difference between
tn3270eSrvrStatsMaxPtrs and tn3270eSrvrStatsInUsePtrs

```
        in a conceptual row does not equal the value of
        tn3270eSrvrStatsSparePtrs in that row:  a printer
        resource may exist but not be usable by a client
        connection.

        Alternatively, the administrative ceiling represented
        by tn3270eSrvrStatsMaxPtrs may have been lowered to
        a point where it is less than the current value of
        tn3270eSrvrStatsInUsePtrs.  In this case
        tn3270eSrvrStatsSparePtrs returns the value 0."
    ::= { tn3270eSrvrStatsEntry 8 }

  tn3270eSrvrStatsInConnects OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "connections"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
        "Indicates the number of client (TCP) connections
        that succeeded at a TN3270E server for the
        granularity of this conceptual row (server-wide,
        port, or port-and-local-address).

        The tn3270eSrvrStatsConnResrceRejs and
        tn3270eSrvrStatsConnErrorRejs objects provide a count
        of failed connection attempts.

        A Management Station can detect discontinuities in
        this counter by monitoring the tn3270eSrvrStatsUpTime
        object."
    ::= { tn3270eSrvrStatsEntry 9 }

  tn3270eSrvrStatsConnResrceRejs OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "connection attempts"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
        "Indicates the number of (TCP) connections rejected
        during connection setup at a TN3270E server for the
        granularity of this conceptual row (server-wide,
        port, or port-and-local-address) due to a lack of
        resources at the server.  An example of when this
        counter would be incremented is when no terminal
        or printer resource is available to associate with a
        client's TCP connection.

        A Management Station can detect discontinuities in
        this counter by monitoring the tn3270eSrvrStatsUpTime
```

```
         object."
      ::= { tn3270eSrvrStatsEntry 10 }

  tn3270eSrvrStatsDisconnects OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "disconnections"
```

```
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of (TCP) connections that were
         disconnected at a TN3270E server for the
         granularity of this conceptual row (server-wide,
         port, or port-and-local-address).

         A Management Station can detect discontinuities in
         this counter by monitoring the tn3270eSrvrStatsUpTime
         object."
      ::= { tn3270eSrvrStatsEntry 11 }

  tn3270eSrvrStatsHCInOctets OBJECT-TYPE
      SYNTAX      Counter64
      UNITS       "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of octets received from TN3270
         and TN3270E clients for the granularity of this
         conceptual row (server-wide, port, or
         port-and-local-address).

         A Management Station can detect discontinuities in
         this counter by monitoring the tn3270eSrvrStatsUpTime
         object."
      ::= { tn3270eSrvrStatsEntry 12 }

  tn3270eSrvrStatsInOctets OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Low-order 32 bits of tn3270eSrvrStatsHCInOctets for
         this conceptual row.

         A Management Station can detect discontinuities in
         this counter by monitoring the tn3270eSrvrStatsUpTime
         object."
      ::= { tn3270eSrvrStatsEntry 13 }

  tn3270eSrvrStatsHCOutOctets OBJECT-TYPE
      SYNTAX      Counter64
      UNITS       "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
```

"Indicates the number of octets sent to TN3270
and TN3270E clients for the granularity of this
conceptual row (server-wide, port, or
port-and-local-address).

A Management Station can detect discontinuities in

```
        this counter by monitoring the tn3270eSrvrStatsUpTime
        object."
    ::= { tn3270eSrvrStatsEntry 14 }

  tn3270eSrvrStatsOutOctets OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Low-order 32 bits of tn3270eSrvrStatsHCOutOctets for
         this conceptual row.

         A Management Station can detect discontinuities in
         this counter by monitoring the tn3270eSrvrStatsUpTime
         object."
      ::= { tn3270eSrvrStatsEntry 15 }

  tn3270eSrvrStatsConnErrorRejs OBJECT-TYPE
      SYNTAX      Counter32
      UNITS       "connection attempts"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
         "Indicates the number of (TCP) connections rejected
         during connection setup at a TN3270E server for the
         granularity of this conceptual row (server-wide,
         port, or port-and-local-address) due to an error
         of some type.  An example of when this counter would
         be incremented is when the client and the server
         cannot agree on a common set of TN3270E functions for
         the connection.

         A Management Station can detect discontinuities in
         this counter by monitoring the tn3270eSrvrStatsUpTime
         object."
      ::= { tn3270eSrvrStatsEntry 16 }

  tn3270eClientGroupTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eClientGroupEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table defines client address groupings for use
          by a TN3270E server.

          No entry in this table shall exist without
          a corresponding (same tn3270eSrvrConfIndex) entry in
          the tn3270eSrvrConfTable existing.
```

```
            An implementation SHOULD NOT retain SNMP-created
            entries in this table across re-IPLs (Initial Program
            Loads) of the corresponding TN3270E server."
        ::= { tn3270eObjects 4 }
```

```
  tn3270eClientGroupEntry OBJECT-TYPE
      SYNTAX      Tn3270eClientGroupEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of a single client address entry.  All
          entries with the same first two indexes,
          tn3270eSrvrConfIndex and tn3270eClientGroupName, are
          considered to be in the same client group."
      INDEX       { tn3270eSrvrConfIndex,
                    tn3270eClientGroupName,
                    tn3270eClientGroupAddrType,
                    tn3270eClientGroupAddress }
      ::= { tn3270eClientGroupTable 1 }

  Tn3270eClientGroupEntry ::= SEQUENCE {
      tn3270eClientGroupName           Utf8String,
      tn3270eClientGroupAddrType       IANATn3270eAddrType,
      tn3270eClientGroupAddress        Tn3270eTAddress,
      tn3270eClientGroupSubnetMask     IpAddress,
      tn3270eClientGroupPfxLength      Unsigned32,
      tn3270eClientGroupRowStatus      RowStatus
  }

  tn3270eClientGroupName OBJECT-TYPE
      SYNTAX      Utf8String (SIZE(1..24))
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of a client group.  Note: client group
          names are required to be unique only with respect
          to a single TN3270E server."
      ::= { tn3270eClientGroupEntry 1 }

  tn3270eClientGroupAddrType OBJECT-TYPE
      SYNTAX      IANATn3270eAddrType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Indicates the type of the address represented in
           tn3270eClientGroupAddress."
      ::= { tn3270eClientGroupEntry 2 }

  tn3270eClientGroupAddress OBJECT-TYPE
      SYNTAX      Tn3270eTAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The client address of a member of a client group.
```

```
          The value of tn3270eClientGroupAddrType indicates
          the address type (IPv4 or IPv6, for example)."
      ::= { tn3270eClientGroupEntry 3 }

  tn3270eClientGroupSubnetMask OBJECT-TYPE
      SYNTAX      IpAddress
```

```
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "The corresponding subnet mask associated with
           tn3270eClientGroupAddress.  A single IP address is
           represented by having this object contain the value
           of 255.255.255.255.

           This object's value is meaningful only if
           tn3270eClientGroupAddrType has a value of ipv4(1)."
      DEFVAL { 'FFFFFFFF'H }
      ::= { tn3270eClientGroupEntry 4 }

  tn3270eClientGroupPfxLength OBJECT-TYPE
      SYNTAX      Unsigned32 (0..128)
      UNITS       "bits"
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "The corresponding IPv6 network prefix length.

          This object's value is meaningful only if
          tn3270eClientGroupAddrType  has a value of ipv6(2)."
      DEFVAL { 0 }
      ::= { tn3270eClientGroupEntry 5 }

  tn3270eClientGroupRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
          "This object allows entries to be created and deleted
          in the tn3270eClientGroupTable.  Entries may also be
          created and deleted as a result of implementation-
          dependent operations.

          An entry in this table is deleted by setting this object
          to destroy(6).  When the number of entries in this table
          for a given client group becomes 0, this has the side-
          effect of removing any entries for the group in the
          tn3270eClientResMapTable."
      REFERENCE
          "RFC 1903, 'Textual Conventions for version 2 of the
          Simple Network Management Protocol (SNMPv2).'"
      ::= { tn3270eClientGroupEntry 6 }

  tn3270eResPoolTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eResPoolEntry
      MAX-ACCESS  not-accessible
```

```
      STATUS      current
      DESCRIPTION
          "This table defines resource groupings; the term
          'pool' is used as it is defined by RFC 1647.

          No entry in this table shall exist without
```

```
          a corresponding (same tn3270eSrvrConfIndex) entry in
          the tn3270eSrvrConfTable existing.

          An implementation SHOULD NOT retain SNMP-created
          entries in this table across re-IPLs (Initial Program
          Loads) of the corresponding TN3270E server."
      ::= { tn3270eObjects 5 }

  tn3270eResPoolEntry OBJECT-TYPE
      SYNTAX      Tn3270eResPoolEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of a single resource pool member.  All entries
          with the same first two indexes, tn3270eSrvrConfIndex and
          tn3270eResPoolName, are considered to be in the same pool."
      INDEX  { tn3270eSrvrConfIndex,
               tn3270eResPoolName,
               tn3270eResPoolElementName }
      ::= { tn3270eResPoolTable 1 }

  Tn3270eResPoolEntry ::= SEQUENCE {
      tn3270eResPoolName           Utf8String,
      tn3270eResPoolElementName    SnaResourceName,
      tn3270eResPoolElementType    Tn3270ResourceType,
      tn3270eResPoolRowStatus      RowStatus
  }

  tn3270eResPoolName OBJECT-TYPE
      SYNTAX      Utf8String (SIZE(1..24))
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of a resource pool."
      ::= { tn3270eResPoolEntry 1 }

  tn3270eResPoolElementName OBJECT-TYPE
      SYNTAX      SnaResourceName
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of a member of a resource pool."
      ::= { tn3270eResPoolEntry 2 }

  tn3270eResPoolElementType OBJECT-TYPE
      SYNTAX      Tn3270ResourceType
      MAX-ACCESS  read-create
      STATUS      current
      DESCRIPTION
```

```
          "The type of the entity in a resource pool."
      ::= { tn3270eResPoolEntry 3 }

  tn3270eResPoolRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
```

```
      STATUS      current
      DESCRIPTION
          "This object allows entries to be created and deleted
          in the tn3270eResPoolTable.  Entries may also be
          created and deleted as a result of implementation-
          dependent operations.

          An entry in this table is deleted by setting this object
          to destroy(6).  When all entries in this table associated
          with the same tn3270eResPoolElementName have been removed,
          then any associated (tn3270eResPoolElementName matching
          tn3270eClientResMapPoolName with same tn3270eSrvrConfIndex
          values) entries in the tn3270eClientResMapTable SHALL
          also be removed."
      REFERENCE
          "RFC 1903, 'Textual Conventions for version 2 of the
          Simple Network Management Protocol (SNMPv2).'"
      ::= { tn3270eResPoolEntry 4 }

  tn3270eSnaMapTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eSnaMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table provide a mapping from the name by which
          a secondary LU is known in the SNA network to the
          name by which it is known locally at the TN3270e
          server.  This latter name serves as an index into
          the tn3270eResPoolTable and the tn3270eResMapTable.

          No entry in this table shall exist without
          a corresponding (same tn3270eSrvrConfIndex) entry in
          the tn3270eSrvrConfTable existing."
      ::= { tn3270eObjects 6 }

  tn3270eSnaMapEntry OBJECT-TYPE
      SYNTAX      Tn3270eSnaMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of a single mapping from an SSCP-supplied
           SLU name to a local SLU name."
      INDEX  { tn3270eSrvrConfIndex,
               tn3270eSnaMapSscpSuppliedName }
      ::= { tn3270eSnaMapTable 1 }

  Tn3270eSnaMapEntry ::= SEQUENCE {
      tn3270eSnaMapSscpSuppliedName     SnaResourceName,
      tn3270eSnaMapLocalName            SnaResourceName,
```

```
      tn3270eSnaMapPrimaryLuName        SnaResourceName
  }

  tn3270eSnaMapSscpSuppliedName OBJECT-TYPE
      SYNTAX      SnaResourceName
      MAX-ACCESS  not-accessible
```

```
      STATUS      current
      DESCRIPTION
          "The name of the secondary LU (SLU) as it is known in
           the SNA network.  This name is sent by the SSCP on
           the Activate Logical Unit (ACTLU) request."
      ::= { tn3270eSnaMapEntry 1 }

  tn3270eSnaMapLocalName OBJECT-TYPE
      SYNTAX      SnaResourceName
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The local name of the secondary LU (SLU)."
      ::= { tn3270eSnaMapEntry 2 }

  tn3270eSnaMapPrimaryLuName  OBJECT-TYPE
      SYNTAX   SnaResourceName
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "When there is a currently active LU-LU session for
          this connection, this object returns the primary LU
          (PLU) name from the BIND.  When there is no active
          LU-LU session, or when the PLU name is unavailable
          for some other reason, this object returns a
          zero-length octet string."
      ::= { tn3270eSnaMapEntry 3 }

  tn3270eClientResMapTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eClientResMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table defines resource-pool to client-group
          mappings.  Since both the resource pool name and client
          group name are included in the index clause of this
          table, multiple resource pools can be assigned to the
          same client group.  This enables use of multiple
          resource pools for use in client to resource mapping.
          Assigning multiple client groups to the same resource
          pool is also allowed, but is not the primary purpose
          for how the indexing is structured.

          Assignment of a resource pool to client group can be
          restricted based on TCP port.  An index value of 0 for
          tn3270eClientResMapClientPort disables restriction of
          resource assignment based on client target port
          selection.
```

No entry in this table shall exist without
a corresponding (same tn3270eSrvrConfIndex) entry in
the tn3270eSrvrConfTable existing.

An implementation SHOULD NOT retain SNMP-created
entries in this table across re-IPLs (Initial Program

```
          Loads) of the corresponding TN3270E server."
      ::= { tn3270eObjects 7 }

  tn3270eClientResMapEntry OBJECT-TYPE
      SYNTAX      Tn3270eClientResMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of a single resource pool to client group
          mapping."
      INDEX  { tn3270eSrvrConfIndex,
               tn3270eClientResMapPoolName,
               tn3270eClientResMapClientGroupName,
               tn3270eClientResMapClientPort }
      ::= { tn3270eClientResMapTable 1 }

  Tn3270eClientResMapEntry ::= SEQUENCE {
      tn3270eClientResMapPoolName           Utf8String,
      tn3270eClientResMapClientGroupName    Utf8String,
      tn3270eClientResMapClientPort         Unsigned32,
      tn3270eClientResMapRowStatus          RowStatus
  }

  tn3270eClientResMapPoolName OBJECT-TYPE
      SYNTAX      Utf8String (SIZE(1..24))
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of a resource pool."
      ::= { tn3270eClientResMapEntry 1 }

  tn3270eClientResMapClientGroupName OBJECT-TYPE
      SYNTAX      Utf8String (SIZE(1..24))
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of the client group that is mapped to a
           resource pool."
      ::= { tn3270eClientResMapEntry 2 }

  tn3270eClientResMapClientPort OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65535)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "A port number restricting the scope of a mapping
          from a resource pool to a client group.  The
          value 0 for this object indicates that the scope
          of the mapping is not restricted."
```

```
      ::= { tn3270eClientResMapEntry 3 }

  tn3270eClientResMapRowStatus OBJECT-TYPE
      SYNTAX      RowStatus
      MAX-ACCESS  read-create
      STATUS      current
```

```
      DESCRIPTION
          "This object allows entries to be created and deleted
          in the tn3270eClientResMapTable.  Entries may also be
          created and deleted as a result of implementation-
          dependent operations.

          An entry in this table is deleted by setting this object
          to destroy(6).  Removing an entry from this table doesn't
          affect any other table entry defined in this MIB."
      REFERENCE
          "RFC 1903, 'Textual Conventions for version 2 of the
          Simple Network Management Protocol (SNMPv2).'"
      ::= { tn3270eClientResMapEntry 4 }

  tn3270eResMapTable OBJECT-TYPE
      SYNTAX      SEQUENCE OF Tn3270eResMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "This table defines the actual mapping of a resource
          to a client address.

          No entry in this table shall exist without
          a corresponding (same tn3270eSrvrConfIndex) entry in
          the tn3270eSrvrConfTable existing."
      ::= { tn3270eObjects 8 }

  tn3270eResMapEntry OBJECT-TYPE
      SYNTAX      Tn3270eResMapEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Definition of the mapping of a Resource Element to
           a client address."
      INDEX  { tn3270eSrvrConfIndex,
               tn3270eResMapElementName }
      ::= { tn3270eResMapTable 1 }

  Tn3270eResMapEntry ::= SEQUENCE {
      tn3270eResMapElementName       SnaResourceName,
      tn3270eResMapAddrType          IANATn3270eAddrType,
      tn3270eResMapAddress           Tn3270eTAddress,
      tn3270eResMapPort              Unsigned32,
      tn3270eResMapElementType       Tn3270ResourceType,
      tn3270eResMapSscpSuppliedName  SnaResourceName
  }

  tn3270eResMapElementName OBJECT-TYPE
      SYNTAX      SnaResourceName
```

```
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The name of a resource element.  This is the name by
          which the server implementing this table knows the
          resource.  It may be different from the name by which
```

```
          the resource is known in the SNA network.  This latter
          name is returned in the tn3270eResMapSscpSuppliedName
          object."
      ::= { tn3270eResMapEntry 1 }

  tn3270eResMapAddrType OBJECT-TYPE
      SYNTAX      IANATn3270eAddrType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Indicates the type of the client address represented
          in tn3270eResMapAddress."
      ::= { tn3270eResMapEntry 2 }

  tn3270eResMapAddress OBJECT-TYPE
      SYNTAX      Tn3270eTAddress
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A client address."
      ::= { tn3270eResMapEntry 3 }

  tn3270eResMapPort OBJECT-TYPE
      SYNTAX      Unsigned32 (0..65535)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "A client port."
      ::= { tn3270eResMapEntry 4 }

  tn3270eResMapElementType OBJECT-TYPE
      SYNTAX      Tn3270ResourceType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The type of the associated resource element."
      ::= { tn3270eResMapEntry 5 }

  tn3270eResMapSscpSuppliedName OBJECT-TYPE
      SYNTAX      SnaResourceName
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The name of the secondary LU (SLU) as it is known
          in a SNA network.  This name is sent by the SSCP on
          the Activate Logical Unit (ACTLU) request.  If this
          name is not known, this object returns a zero-length
          octet string."
      ::= { tn3270eResMapEntry 6 }
```

```
  -- Define the set of objects to supplement the TCP Connection Table

  tn3270eTcpConnTable OBJECT-TYPE
      SYNTAX SEQUENCE OF Tn3270eTcpConnEntry
      MAX-ACCESS  not-accessible
```

```
      STATUS      current
      DESCRIPTION
           "This table has an entry for each TN3270(E) client
           connection that is currently active at a TN3270E server.
           An implementation MAY retain entries for connections
           that have been terminated, but which entries are
           retained, how many entries are retained, and how long
           they are retained is entirely implementation-dependent.

           The indexing for this table is designed to support the
           use of an SNMP GET-NEXT operation using only the remote
           address type, remote address, and remote port, as a way
           for a Management Station to retrieve the table entries
           related to a particular TN3270(E) client."
      ::= { tn3270eObjects 9 }

  tn3270eTcpConnEntry OBJECT-TYPE
      SYNTAX Tn3270eTcpConnEntry
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Provides information about a single TN3270/TN3270E
          session.  Note: a tn3270eSrvrConfIndex is not needed
          in this table, since the combination of remote and
          local addresses and ports is sufficient to
          guarantee uniqueness across the TN3270E servers
          serviced by an SNMP agent.  Because of this indexing
          structure, however, this table does not support
          view-based access control policies that provide
          access to table rows on a per-server basis."
      INDEX { tn3270eTcpConnRemAddrType,
              tn3270eTcpConnRemAddress,
              tn3270eTcpConnRemPort,
              tn3270eTcpConnLocalAddrType,
              tn3270eTcpConnLocalAddress,
              tn3270eTcpConnLocalPort
            }
      ::= { tn3270eTcpConnTable 1 }

  Tn3270eTcpConnEntry ::=
      SEQUENCE
      {
      tn3270eTcpConnRemAddrType          IANATn3270eAddrType,
      tn3270eTcpConnRemAddress           Tn3270eTAddress,
      tn3270eTcpConnRemPort              Unsigned32,
      tn3270eTcpConnLocalAddrType        IANATn3270eAddrType,
      tn3270eTcpConnLocalAddress         Tn3270eTAddress,
      tn3270eTcpConnLocalPort            Unsigned32,
      tn3270eTcpConnLastActivity         TimeTicks,
```

```
      tn3270eTcpConnBytesIn             Counter32,
      tn3270eTcpConnBytesOut            Counter32,
      tn3270eTcpConnResourceElement     SnaResourceName,
      tn3270eTcpConnResourceType        Tn3270ResourceType,
      tn3270eTcpConnDeviceType          Tn3270DeviceTypes,
      tn3270eTcpConnFunctions           IANATn3270Functions,
```

```
      tn3270eTcpConnId                    Unsigned32,
      tn3270eTcpConnClientIdFormat        IANATn3270eClientType,
      tn3270eTcpConnClientId              OCTET STRING,
      tn3270eTcpConnTraceData             Tn3270eTraceData,
      tn3270eTcpConnLogInfo               IANATn3270eLogData,
      tn3270eTcpConnLuLuBindImage         OCTET STRING,
      tn3270eTcpConnSnaState              INTEGER,
      tn3270eTcpConnStateLastDiscReason   INTEGER,
      tn3270eTcpConnSrvrConfIndex         Unsigned32,
      tn3270eTcpConnActivationTime        TimeStamp
      }

  tn3270eTcpConnRemAddrType OBJECT-TYPE
      SYNTAX   IANATn3270eAddrType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "Indicates the type of the value of the
          tn3270eTcpConnRemAddress object.  For example,
          ipv4(1) or ipv6(2)."
      ::= { tn3270eTcpConnEntry 1 }

  tn3270eTcpConnRemAddress OBJECT-TYPE
      SYNTAX      Tn3270eTAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The remote address associated with a TN3270E client.
           tn3270eTcpConnRemAddrType indicates the address type
           (IPv4 or IPv6, for example).

           If a TN3270(E) client is connected to its
           server via a proxy client the address represented by
           the value of this object shall be the remote client's
           address, not the proxy client's address."
      ::= { tn3270eTcpConnEntry 2 }

  tn3270eTcpConnRemPort OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The remote port associated with a TN3270E client.

           If a TN3270(E) client is connected to its
           server via a proxy client the port represented by
           the value of this object shall be the remote client's
           port, not the proxy client's port."
      ::= { tn3270eTcpConnEntry 3 }
```

```
  tn3270eTcpConnLocalAddrType OBJECT-TYPE
      SYNTAX   IANATn3270eAddrType
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
```

```
          "Indicates the type of the value of the
          tn3270eTcpConnLocalAddress object.  For example,
          ipv4(1) or ipv6(2)."
      ::= { tn3270eTcpConnEntry 4 }

  tn3270eTcpConnLocalAddress OBJECT-TYPE
      SYNTAX      Tn3270eTAddress
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The local address associated with a TN3270E client.
           tn3270eTcpConnRemAddrType indicates the address type
           (IPv4 or IPv6, for example)."
      ::= { tn3270eTcpConnEntry 5 }

  tn3270eTcpConnLocalPort OBJECT-TYPE
      SYNTAX      Unsigned32 (1..65535)
      MAX-ACCESS  not-accessible
      STATUS      current
      DESCRIPTION
          "The remote port associated with a TN3270E client."
      ::= { tn3270eTcpConnEntry 6 }

  tn3270eTcpConnLastActivity OBJECT-TYPE
      SYNTAX   TimeTicks
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The number of 100ths of seconds since any data was
           transferred for the associated TCP Connection."
      DEFVAL  { 0 }
      ::= { tn3270eTcpConnEntry 7 }

  tn3270eTcpConnBytesIn OBJECT-TYPE
      SYNTAX   Counter32
      UNITS "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The number of bytes received by the server from TCP
          for this connection.

          A Management Station can detect discontinuities in
          this counter by monitoring the
          tn3270eTcpConnActivationTime object."
      ::= { tn3270eTcpConnEntry 8 }

  tn3270eTcpConnBytesOut OBJECT-TYPE
      SYNTAX   Counter32
```

```
      UNITS "octets"
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The number of bytes sent to TCP for this connection.
```

```
          A Management Station can detect discontinuities in
          this counter by monitoring the
          tn3270eTcpConnActivationTime object."
      ::= { tn3270eTcpConnEntry 9 }

  tn3270eTcpConnResourceElement OBJECT-TYPE
      SYNTAX   SnaResourceName
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "LU/Print secondary name for connecting an client
           into an SNA network."
      ::= { tn3270eTcpConnEntry 10 }

  tn3270eTcpConnResourceType OBJECT-TYPE
      SYNTAX   Tn3270ResourceType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Indicates the type of resource identified by
           tn3270eTcpConnResourceElement."
      ::= { tn3270eTcpConnEntry 11 }

  tn3270eTcpConnDeviceType OBJECT-TYPE
      SYNTAX   Tn3270DeviceTypes
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Indicates the device type if negotiated with the
          client.  A value of unknown(0) should be used as the
          value of this object when a device type is not
          negotiated.  Refer to RFC 1647 for how device types
          can be negotiated."
      ::= { tn3270eTcpConnEntry 12 }

  tn3270eTcpConnFunctions OBJECT-TYPE
      SYNTAX   IANATn3270Functions
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "This object indicates which of the TN3270 and TN3270E
          functions were negotiated by the server and the client
          for this TCP connection.

          Refer to tn3270eSrvrFunctionsSupported for the list of
          these functions supported by the server."
      ::= { tn3270eTcpConnEntry 13 }

  tn3270eTcpConnId OBJECT-TYPE
```

```
      SYNTAX      Unsigned32
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The connection identifier associated with a TN3270 or
           a TN3270E session's TCP connection.  TCP implementations
```

```
           often assign a unique (with respect to itself) unsigned
           integer as an identifier for a TCP connection.

           The value 0 indicates that a connection does not have
           a valid connection identifier."
       ::= { tn3270eTcpConnEntry 14 }

  tn3270eTcpConnClientIdFormat OBJECT-TYPE
      SYNTAX      IANATn3270eClientType
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The format of a corresponding tn3270eTcpConnClientId
           object as defined by the IANSTn3270eClientType
           textual convention imported from the
           IANATn3270eTC-MIB."
      ::= { tn3270eTcpConnEntry 15 }

  tn3270eTcpConnClientId OBJECT-TYPE
      SYNTAX      OCTET STRING (SIZE (0..512))
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Additional client identification information.  The
          type of this information is indicated by the value of
          the corresponding tn3270eTcpConnClientIdFormat object.
          All values are returned in network-byte order.

          The purpose of this object is to provide an alternate
          means of identifying a client, other than though the
          remote address returned in tn3270eTcpConnRemAddress."
      ::= { tn3270eTcpConnEntry 16 }

  tn3270eTcpConnTraceData OBJECT-TYPE
      SYNTAX   Tn3270eTraceData
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Trace data for this session."
      ::= { tn3270eTcpConnEntry 17 }

  tn3270eTcpConnLogInfo OBJECT-TYPE
      SYNTAX      IANATn3270eLogData
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "Log information, encoded as specified in the
          IANATn3270eLogData textual convention from the
          IANAtn3270eTC-MIB."
```

```
      ::= { tn3270eTcpConnEntry 18 }

  tn3270eTcpConnLuLuBindImage  OBJECT-TYPE
      SYNTAX   OCTET STRING (SIZE (0..256))
      MAX-ACCESS  read-only
      STATUS      current
```

```
      DESCRIPTION
          "When there is a currently active LU-LU session for
          this connection, this object returns the BIND Image
          (defined to be bytes 1-p of the complete BIND Request
          Unit -- see 'SNA Formats' for more information)
          that was received from the PLU during session
          activation.  When there is no active LU-LU session,
          or when a BIND image is unavailable for some other
          reason, this object returns a zero-length octet
          string."
      REFERENCE
          "'Systems Network Architecture Formats', IBM
          Publication GA27-3136."
      ::= { tn3270eTcpConnEntry 19 }

  tn3270eTcpConnSnaState  OBJECT-TYPE
      SYNTAX   INTEGER {
                    unknown(0),
                    noSluSession(1),
                    sscpLuSession(2),  -- but no LU-LU session
                    luLuSession(3),    -- but no SSCP-LU session
                    sscpLuSessionAndLuLuSession(4)
                       }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The current state of the SNA side of the end-to-end
          TN3270 connection.  The following states are defined:

              unknown(0)          - The true state is not known.
              noSluSession(1)     - The SLU has neither an SSCP-LU
                                    nor an LU-LU session active.
              sscpLuSession(2)    - The SSCP-LU session for the SLU
                                    is active, but the SLU is not
                                    currently in session with a PLU.
              luLuSession(3)      - The SLU is currently in session
                                    with a PLU, but the SSCP-LU
                                    session for the LU is not active.
              sscpLuSessionAndLuLuSession(4) - The SLU currently has
                                    an active session with a PLU,
                                    and the SSCP-LU session for the
                                    SLU is active."

      ::= { tn3270eTcpConnEntry 20 }

  tn3270eTcpConnStateLastDiscReason OBJECT-TYPE
      SYNTAX   INTEGER {
                   unknown(0),
                   hostSendsUnbind(1),
```

```
                    hostDontAcceptConnection(2),
                    outOfResource(3),
                    clientProtocolError(4),
                    invalidDeviceName(5),
                    deviceInUse(6),
                    inactivityTimeout(7),
```

```
                    hostNotResponding(8),
                    clientNotResponding(9),
                    serverClose(10),
                    sysreqLogoff(11),
                    serverSpecificHexCode(12)
                       }
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The last disconnect reason.  A session that has not
          experienced a disconnect shall use the value unknown(0)
          for this object.  Depending on when an implementation
          removes entries from this table, certain states may
          never be returned."
      ::= { tn3270eTcpConnEntry 21 }

  tn3270eTcpConnSrvrConfIndex OBJECT-TYPE
      SYNTAX   Unsigned32 (1..4294967295)
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "tn3270eSrvrConfIndex of the tn3270eSrvrConfEntry
          belonging to the TN3270E server to which this entry
          belongs."
      ::= { tn3270eTcpConnEntry 22 }

  tn3270eTcpConnActivationTime OBJECT-TYPE
      SYNTAX      TimeStamp
      MAX-ACCESS  read-only
      STATUS      current
      DESCRIPTION
          "The value of the sysUpTime object the last time
          this TCP connection became active."
      ::= { tn3270eTcpConnEntry 23 }

  tn3270eConfSpinLock OBJECT-TYPE
      SYNTAX      TestAndIncr
      MAX-ACCESS  read-write
      STATUS      current
      DESCRIPTION
        "An advisory lock used to allow cooperating
        TN3270E-MIB applications to coordinate their use
        of the tn3270eSrvrConfTable, the tn3270eSrvrPortTable,
        the tn3270eClientGroupTable, the tn3270eResPoolTable,
        and the tn3270eClientResMapTable.

        When creating a new entry or altering an existing entry
        in the any of the tables mentioned above, an application
        should make use of tn3270eRtSpinLock to serialize
```

```
        application changes or additions.

        Since this is an advisory lock, the use of this lock is
        not enforced."
      ::= { tn3270eObjects 10 }
```

```
  -- Conformance Definitions

  tn3270eGroups      OBJECT IDENTIFIER ::= { tn3270eConformance 1 }

  tn3270eCompliances OBJECT IDENTIFIER ::= { tn3270eConformance 2 }

  -- compliance statements

  tn3270eCompliance MODULE-COMPLIANCE
      STATUS      current
      DESCRIPTION
          "The compliance statement for agents that support the
          TN3270E-MIB."
      MODULE -- this module
          MANDATORY-GROUPS { tn3270eBasicGroup,
                             tn3270eSessionGroup
                           }
          GROUP       tn3270eResMapGroup
          DESCRIPTION
              "This group is optional and provides a method of
              performing tn3270eClientGroup to tn3270eResPool
              mapping."

          GROUP       tn3270eHiCapacityGroup
          DESCRIPTION
              "This group is optional and provides for support
              of high capacity counters."

          OBJECT tn3270eSrvrConfActivityCheck
             MIN-ACCESS  read-only
             DESCRIPTION
                "The agent is not required to support a set to this
                 object if the associated TN3270E server doesn't
                 support either TIMING-MARK or NOP processing.  In
                 this case an agent should return noCheck on
                 retrieval."

          OBJECT tn3270eSrvrConfTmNopInactTime
             MIN-ACCESS  read-only
             DESCRIPTION
                "The agent is not required to support a set to this
                 object if the functions enabled by
                 tn3270eSrvrConfActivityCheck are not supported.
                 An agent in this case should return a value of 0."

          OBJECT tn3270eSrvrConfTmNopInterval
             MIN-ACCESS  read-only
             DESCRIPTION
                "The agent is not required to support a set to this
```

```
                object if the functions enabled by
                tn3270eSrvrConfActivityCheck are not supported.
                An agent in this case should return a value of 0."

          OBJECT  tn3270eSrvrConfAdminStatus
              DESCRIPTION
```

```
                "A TN3270E server is not required to support a
                 stopImmediate state transition."

          OBJECT tn3270eSrvrConfRowStatus
             MIN-ACCESS read-only
             DESCRIPTION
                 "Write access is not required."

          OBJECT tn3270eSrvrConfTmTimeout
             MIN-ACCESS  read-only
             DESCRIPTION
                "The agent is not required to support a set to this
                 object if the functions enabled by
                 tn3270eSrvrConfActivityCheck are not supported.
                 An agent in this case should return a value of 0."

          OBJECT tn3270eSrvrPortRowStatus
             MIN-ACCESS read-only
             DESCRIPTION
                 "Write access is not required."

          OBJECT tn3270eClientGroupRowStatus
             MIN-ACCESS read-only
             DESCRIPTION
                 "Write access is not required."

          OBJECT tn3270eResPoolRowStatus
             MIN-ACCESS read-only
             DESCRIPTION
                 "Write access is not required."

          OBJECT tn3270eClientResMapRowStatus
             MIN-ACCESS read-only
             DESCRIPTION
                 "Write access is not required."

      ::= { tn3270eCompliances 1 }

  -- units of conformance

  tn3270eBasicGroup OBJECT-GROUP
      OBJECTS {
          tn3270eSrvrConfInactivityTimeout,
          tn3270eSrvrConfActivityCheck,
          tn3270eSrvrConfTmNopInactTime,
          tn3270eSrvrConfTmNopInterval,
          tn3270eSrvrFunctionsSupported,
          tn3270eSrvrConfAdminStatus,
          tn3270eSrvrConfOperStatus,
          tn3270eSrvrConfSessionTermState,
```

```
          tn3270eSrvrConfSrvrType,
          tn3270eSrvrConfContact,
          tn3270eSrvrConfRowStatus,
          tn3270eSrvrConfLastActTime,
          tn3270eSrvrConfTmTimeout,
```

```
            tn3270eSrvrPortRowStatus,
            tn3270eSrvrStatsUpTime,
            tn3270eSrvrStatsMaxTerms,
            tn3270eSrvrStatsInUseTerms,
            tn3270eSrvrStatsSpareTerms,
            tn3270eSrvrStatsMaxPtrs,
            tn3270eSrvrStatsInUsePtrs,
            tn3270eSrvrStatsSparePtrs,
            tn3270eSrvrStatsInConnects,
            tn3270eSrvrStatsConnResrceRejs,
            tn3270eSrvrStatsDisconnects,
            tn3270eSrvrStatsInOctets,
            tn3270eSrvrStatsOutOctets,
            tn3270eSrvrStatsConnErrorRejs,
            tn3270eClientGroupSubnetMask,
            tn3270eClientGroupPfxLength,
            tn3270eClientGroupRowStatus,
            tn3270eSnaMapLocalName,
            tn3270eSnaMapPrimaryLuName,
            tn3270eConfSpinLock
        }
        STATUS  current
        DESCRIPTION
            "This group is mandatory for all hosts supporting the
             TN3270E-MIB."
        ::= { tn3270eGroups 1 }

    tn3270eSessionGroup OBJECT-GROUP
        OBJECTS {
            tn3270eResMapAddrType,
            tn3270eResMapAddress,
            tn3270eResMapPort,
            tn3270eResMapElementType,
            tn3270eResMapSscpSuppliedName,
            tn3270eTcpConnLastActivity,
            tn3270eTcpConnBytesIn,
            tn3270eTcpConnBytesOut,
            tn3270eTcpConnResourceElement,
            tn3270eTcpConnResourceType,
            tn3270eTcpConnDeviceType,
            tn3270eTcpConnFunctions,
            tn3270eTcpConnSrvrConfIndex,
            tn3270eTcpConnActivationTime
          }
        STATUS  current
        DESCRIPTION
            "This group is mandatory for all hosts supporting the
             TN3270E-MIB."
        ::= { tn3270eGroups 2 }
```

```
  tn3270eResMapGroup OBJECT-GROUP
      OBJECTS {
          tn3270eResPoolElementType,
          tn3270eResPoolRowStatus,
          tn3270eClientResMapRowStatus,
```

```
          tn3270eTcpConnId,
          tn3270eTcpConnClientIdFormat,
          tn3270eTcpConnClientId,
          tn3270eTcpConnTraceData,
          tn3270eTcpConnLogInfo,
          tn3270eTcpConnLuLuBindImage,
          tn3270eTcpConnSnaState,
          tn3270eTcpConnStateLastDiscReason
        }
      STATUS  current
      DESCRIPTION
          "This group is optional for all hosts supporting the
           TN3270E-MIB."
      ::= { tn3270eGroups 3 }

  tn3270eHiCapacityGroup OBJECT-GROUP
      OBJECTS {
          tn3270eSrvrStatsHCInOctets,
          tn3270eSrvrStatsHCOutOctets
        }
      STATUS  current
      DESCRIPTION
           "Support of these objects is REQUIRED when the
           Counter32 versions can potentially wrap too
           frequently.  This group is optional for all other
           hosts supporting the TN3270E-MIB.

           The IF-MIB (RFC 2233) requires that the 64-bit
           versions of its counters be implemented when an
           interface can support rates of around 20 million
           bits per second or greater.  This implies a minimum
           wrap rate of just over 28 minutes.  It is recommended
           that this same guideline be used for determining
           whether an implementation implements these objects.

           This group contains two objects with the syntax
           Counter64.  An implementation that doesn't support
           these objects should return noSuchObject, since
           returning a zero is misleading."

      ::= { tn3270eGroups 4 }

  END
```

5.0 Security Considerations

Certain management information defined in this MIB may be considered sensitive in some network environments. Therefore, authentication of

received SNMP requests and controlled access to management information SHOULD be employed in such environments. An authentication protocol is defined in [10]. A protocol for access control is defined in [11].

Several objects in this MIB allow write access or provide for row creation. Allowing this support in a non-secure environment can have a

negative effect on network operations. It is RECOMMENDED that implementers seriously consider whether set operations or row creation should be allowed without providing, at a minimum, authentication of request origin. It is RECOMMENDED that without such support, the following objects be implemented as read-only:

- tn3270eSrvrConfInactivityTimout
- tn3270eSrvrConfActivityCheck
- tn3270eSrvrConfActivityTimeout
- tn3270eSrvrConfActivityInterval
- tn3270eSrvrConfAdminStatus
- tn3270eSrvrConfSessionTermState
- tn3270eSrvrConfContact
- tn3270eClientGroupSubnetMask
- tn3270eResPoolElementType
- tn3270eSrvrConfRowStatus
- tn3270eSrvrPortRowStatus
- tn3270eClientGroupRowStatus
- tn3270eResPoolRowStatus
- tn3270eResMapRowStatus

For all tables in the MIB except the tn3270eTcpConnTable, the first index identifies an individual TN3270E server. This makes it easy to implement an access control policy under which different principals have access to objects related to different servers. Implementation of such a policy is not possible for the entries in the tn3270eTcpConTable.

6.0 Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

7.0 Acknowledgments

This document is a product of the TN3270E Working Group. Thanks to Randy Presuhn of BMC Software for his valuable review comments on several versions of the document.

8.0 References

[1] Case, J., M. Fedor, M. Schoffstall, J. Davin, "Simple Network Management Protocol", RFC 1157, SNMP Research, Performance Systems International, MIT Laboratory for Computer Science, May 1990.

[2] McCloghrie, K., and M. Rose, Editors, "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.

[3] Case, J., McCloghrie, K., Rose, M., and Waldbusser S., "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.

[4] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.

[5] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.

[6] Case, J., McCloghrie, K., Rose, M., and Waldbusser, S., "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.

[7] Harrington D., Presuhn, R., Wijnen, B., "An Architecture for Describing SNMP Management Frameworks", RFC 2271, Cabletron Systems, BMC Software, Inc., IBM T.J. Watson Research, January 1998.

[8] Harrington D., Presuhn, R., Wijnen, B., "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, Cabletron Systems, BMC Software, Inc., IBM T.J. Watson Research, January 1998.

[9] Levi D., Meyer P., Stewart, B., "SNMPv3 Applications", RFC 2273, SNMP Research, Inc., Secure Computing Corporation, Cisco Systems, January 1998.

[10] Blumenthal, U., Wijnen, B., "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC

2274, IBM T. J. Watson Research, January 1998.

[11] Wijnen, B., Presuhn, R., McCloghrie, K., "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, IBM T.J. Watson Research, BMC Software, Inc., Cisco Systems, Inc., January 1998.

[12] Postel, J., and Reynolds, J., "Telnet Protocol Specification", RFC 854, May 1983.

[13] Postel, J., and Reynolds, J., "Telnet Timing Mark Option", RFC 860, May 1983.

[14] Rekhter, J., "Telnet 3270 Regime Option", RFC 1041, January 1988.

[15] Kelly, B., "TN3270 Enhancements", RFC 1647, July 1994.

[16] McCloghrie, K., "TCP-MIB Definitions", RFC 2012, November 1996.

[17] Hovey, R., and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.

[18] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[19] Hinden, R., Deering, S., "IP Version 6 Addressing Architecture", <draft-ietf-ipngwg-addr-arch-v2-02.txt>, July 16, 1997.

[20] Krupczak, Cheryl, Saperia, Jonathan, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.

9.0 Authors' Addresses

Kenneth D. White
Dept. BRQA/Bldg. 501/G114
IBM Corporation
P.O.Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709, USA
E-mail: kennethw@vnet.ibm.com

Robert Moore
Dept. BRQA/Bldg. 501/G114
IBM Corporation
P.O.Box 12195
3039 Cornwallis
Research Triangle Park, NC 27709, USA
Phone: +1-919-254-4436
E-mail: remoore@us.ibm.com

10.0 Full Copyright Statement

Copyright (C) The Internet Society (1997). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Appendix A. IANATn3270eTC-MIB

This appendix defines the current content of the IANATn3270eTC-MIB.
This section should be removed from this document prior to its approval,
at which time this MIB will be administered by IANA.

```
  IANATn3270eTC-MIB DEFINITIONS ::= BEGIN

  IMPORTS
      MODULE-IDENTITY, experimental
          FROM SNMPv2-SMI
      TEXTUAL-CONVENTION
          FROM SNMPv2-TC;

    ianaTn3270eTcMib MODULE-IDENTITY
        LAST-UPDATED "9802020000Z" -- February 2, 1998
        ORGANIZATION "IANA"
        CONTACT-INFO
            "Internet Assigned Numbers Authority

             Postal: USC/Information Sciences Institute
                     4676 Admiralty Way, Marina del Rey, CA 90292

             Tel:    +1  310 822 1511
             E-Mail: iana@isi.edu"
        DESCRIPTION
            "This module defines a set of textual conventions
            for use by the TN3270E-MIB and the TN3270E-RT-MIB.

            Any additions or changes to the contents of this
            MIB module must first be discussed on the tn3270e
            working group list at: tn3270e@list.nih.gov
            and approved by one of the following TN3270E
            working group contacts:

                Ed Bailey (co-chair) - elbailey@us.ibm.com
                Michael Boe (co-chair) - mboe@cisco.com
                Ken White - kennethw@vnet.ibm.com
                Robert Moore - remoore@us.ibm.com

            The above list of contacts can be altered with
            the approval of the two co-chairs.

            The Textual Conventions defined within this MIB have
            no security issues associated with them unless
            explicitly stated in their corresponding
            DESCRIPTION clause."
        -- IANA will assign a real OID when they take over this module.
        ::= { experimental 2001 }

  -- Textual Conventions
```

```
  IANATn3270eAddrType ::= TEXTUAL-CONVENTION
      STATUS      current
```

```
    DESCRIPTION
        "The textual convention for defining the type of a
        client address.  The enumeration value unknown(0) is
        usually used to indicate that no actual address is
        present."
    SYNTAX      INTEGER {
                          unknown(0),
                          ipv4(1),
                          ipv6(2)
                        }

IANATn3270Functions ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "This textual convention reflects the current set of
        TN3270 and TN3270E functions that can be negotiated
        between a server and its client:

        RFC856
        transmitBinary  The sender of this command REQUESTS
                        permission to begin transmitting, or
                        confirms that it will now begin
                        transmitting characters which are to
                        be interpreted as 8 bits of binary
                        data by the receiver of the data.
        RFC860
        timingMark      The sender of this command REQUESTS
                        that the receiver of this command
                        return a WILL TIMING-MARK in the data
                        stream at the 'appropriate place'.
        RFC885
        endOfRecord     The sender of this command requests
                        permission to begin transmission of
                        the Telnet END-OF-RECORD (EOR) code
                        when transmitting data characters, or
                        the sender of this command confirms it
                        will now begin transmission of EORs
                        with transmitted data characters.
        RFC1091
        terminalType    Sender is willing to send terminal
                        type information in a subsequent
                        sub-negotiation.
        RFC1041
        tn3270Regime    Sender is willing to send list of
                        supported 3270 Regimes in a
                        subsequent sub-negotiation.
        RFC1647
        scsCtlCodes     (Printer sessions only).  Allows the
                        use of the SNA Character Stream (SCS)
                        and SCS control codes on the session.
                        SCS is used with LU type 1 SNA sessions.
```

dataStreamCtl (Printer sessions only). Allows the use of the standard 3270 data stream. This corresponds to LU type 3 SNA sessions.

```
        responses       Provides support for positive and
                        negative response handling.  Allows the
                        server to reflect to the client any and
                        all definite, exception, and no response
                        requests sent by the host application.
        bindImage       Allows the server to send the SNA Bind
                        image and Unbind notification to the
                        client.
        sysreq          Allows the client and server to emulate
                        some (or all, depending on the server) of
                        the functions of the SYSREQ key in an SNA
                        environment."
    SYNTAX      BITS {
                      transmitBinary(0),-- rfc856
                      timemark(1),      -- rfc860
                      endOfRecord(2),   -- rfc885
                      terminalType(3),  -- rfc1091
                      tn3270Regime(4),  -- rfc1041
                      scsCtlCodes(5),   -- rfc1647
                      dataStreamCtl(6), -- rfc1647
                      responses(7),     -- rfc1647
                      bindImage(8),     -- rfc1647
                      sysreq(9)         -- rfc1647
                    }

IANATn3270eClientType ::= TEXTUAL-CONVENTION
    STATUS      current
    DESCRIPTION
        "The textual convention for defining the set of
         enumerations used by tn3270eTcpConnClientIdFormat
         in the TN3270E-MIB:

         ENUMERATION        OCTETs  DESCRIPTION

         none(0)              0     Not specified
         other(1)           1..512  Implementation specific
         ipv4(2)              6     4-octet IP Address plus
                                    2-octet TCP Port
         ipv6(3)              18    16-octet IPv6 Address
                                    plus 2-octet TCP Port
         domainName(4)      1..512  The DNS name of a
                                    client.
         truncDomainName(5) 1..512  The (truncated) DNS name
                                    of a client.
         string(6)          1..512  Unknown Utf8String
         certificate(7)     1..512  certificate
         userId(8)          1..8    Client's userid
         x509dn(9)          1..512  X.509 Distinguished Name

         Representation of a certificate(7) may be lead to
         a security exposure and is NOT RECOMMENDED without
```

```
            adequate security."
        SYNTAX     INTEGER {
                               none(0),
```

```
                          other(1),
                          ipv4(2),
                          ipv6(3),
                          domainName(4),
                          truncDomainName(5),
                          string(6),
                          certificate(7),
                          userId(8),
                          x509dn(9)
                        }

IANATn3270eLogData ::= TEXTUAL-CONVENTION
   STATUS current
   DESCRIPTION
      "An octet string representing log data as pertaining to
      either a TN3270 or TN3270E Session as reported from a
      TN3270E Server. Log data is stored in an octet string
      in time order (from earliest to latest).

      Each log element has the following form:

      +------+----+---------+------------+
      !length!type!TimeStamp! data       !
      +------+----+---------+------------+

      where

      length    = one-octet length of the data portion of the
                  trace element, not including the length,
                  type, and TimeStamp fields
      type      = one-octet code point characterizing the data.
      TimeStamp = A 4-octet field representing the number of
                  TimeTicks since the TN3270E server was last
                  activated.  The server's last activation time
                  is available in the tn3270eSrvrConfLastActTime
                  object in the TN3270E MIB, which has the
                  syntax DateAndTime.
      data      = initial part of a PDU.

      length   type

        0-255  x'00' - unknown
        0      x'01' - inactivity timer expired
        0      x'02' - dynamic timer expired
        0      x'03' - actlu req
        0      x'04' - bind req
        0      x'05' - clear req
        0      x'06' - dactlu req
        0      x'07' - warm actpu req
        0      x'08' - sdt req
        0      x'09' - unbind req
```

```
          0      x'0A' - notify resp
          0      x'0B' - reply PSID neg rsp
          0      x'0C' - reply PSID pos rsp
```

```
        0      x'0D' - unbind rsp
        0      x'0E' - hierarchical reset
        0      x'0F' - client connect req
        0      x'10' - client disconnect req
        0      x'11' - timingmark received
        0      x'12' - flowControl timer expired
        0      x'13' - neg rsp to host
        0      x'14' - neg rsp from host
        0      x'15' - data contention
        0      x'16' - no buffer to send SNA data
        0      x'17' - receive response while inbound
        0      x'18' - client protocol error
        0      x'19' - badClientSequenceReceived
        1-255  x'1A' - utf8String
        2      x'1B' - hexCode, implementation dependent

        log event entries have a minimum length of 6 octets.
        The zero-length string indicates that no log data is
        available."
    SYNTAX OCTET STRING (SIZE (0 | 6..2048))

END
```