

Network Working Group
Request for Comments: 2561
Category: Standards Track

K. White
IBM Corp.
R. Moore
IBM Corp.
April 1999

Base Definitions of Managed Objects for TN3270E Using SMiv2

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

This memo defines a Management Information Base (MIB) for configuring and managing TN3270E servers. TN3270E, defined by RFC 2355 [19], refers to the enhancements made to the Telnet 3270 (TN3270) terminal emulation practices. Refer to RFC 1041 [18], STD 8, RFC 854 [16], and STD 31, RFC 860 [17] 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.

Table of Contents

1.0	Introduction	2
2.0	The SNMP Network Management Framework	3
3.0	Structure of the MIB	4
3.1	TN3270E Server Control	5
3.1.1	tn3270eSrvrConfTable	5
3.1.2	tn3270eSrvrPortTable	6
3.1.3	tn3270eSrvrStatsTable	7
3.2	TN3270E Server Resource Configuration	7
3.3	Resource Name / Client Address Mappings	8
3.3.1	tn3270eSnaMapTable	8
3.3.2	tn3270eResMapTable	9
3.3.3	tn3270eTcpConnTable	9
3.4	Advisory Spin Lock Usage	9
3.5	Row Persistence	10
3.6	IANA Considerations	10
4.0	Definitions	11
5.0	Security Considerations	51
6.0	Intellectual Property	52
7.0	Acknowledgments	53
8.0	References	53
9.0	Authors' Addresses	55
	Full Copyright Statement	56

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 [22].

2.0 The SNMP Network Management Framework

The SNMP Management Framework presently consists of five major components:

- o An overall architecture, described in RFC 2271 [1].
- o Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in RFC 1902 [5], RFC 1903 [6] and RFC 1904 [7].
- o Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2272 [11] and RFC 2274 [12].
- o Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- o A set of fundamental applications described in RFC 2273 [14] and the view-based access control mechanism described in RFC 2275 [15].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in SMIV1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

3.0 Structure of the MIB

The TN3270E-MIB is split into the following components:

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

There are three additional sections to address:

- o Advisory Spin Lock Usage
- o 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 [24], 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 [1], 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 Tn3270eAddress, are defined for this purpose. IANATn3270eAddress is essentially equivalent to the TAddress TC, defined by RFC 1903. The difference between the two is that IANATn3270eAddress allows a zero-length octet string, while TAddress doesn't. It is important that IANATn3270eAddress 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:

- o tn3270eSrvrConfTable
- o 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 tn3270eSrvrConfConnectivityChk
- o tn3270eSrvrConfTmNopInterval
- o tn3270eSrvrConfTmNopInactTime
- o tn3270eSrvrConfTmTimeout

define the parameters for performing the "Telnet Timing Mark Option" as defined by RFC 860 [17]. The object tn3270eSrvrConfConnectivityChk allows a Management Station to select either a NOP command or a TIMING-MARK command. Sending a NOP command results in less overhead than 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:

- o tn3270eSrvrConfIndex
- o tn3270eSrvrConfPort
- o tn3270eSrvrConfPortAddrType
- o 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:

- o maximum: maximum size of the shared pool
- o in-use: number currently in use as this type of LU
- o 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:

- o tn3270eClientGroupTable
- o 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:

- o tn3270eSnaMapTable
- o tn3270eResMapTable
- o 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 [20], TCP-MIB Definitions). It contains the following objects:

- o tcpConnState (INTEGER)
- o tcpConnLocalAddress (IpAddress)
- o tcpConnLocalPort (INTEGER)
- o tcpConnRemAddress (IpAddress)
- o 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, Tn3270eAddress 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:

- o tn3270eSrvrConfTable
- o tn3270eSrvrPortTable
- o tn3270eClientGroupTable
- o tn3270eResPoolTable
- o 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 [6]) objects to these tables.

3.6 IANA Considerations

The tn3270eSrvrFunctionsSupported, tn3270eTcpConnFunctions, tn3270eTcpConnClientIdFormat, and tn3270eTcpConnLogInfo objects, as well as a number of objects identifying various address types, resource types, and device types, 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 Numbers 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/>".

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, IANATn3270eAddress,
IANATn3270eClientType, IANATn3270Functions,
IANATn3270ResourceType, IANATn3270DeviceType,
IANATn3270eLogData
    FROM IANATn3270eTC-MIB;
```

tn3270eMIB MODULE-IDENTITY

```

LAST-UPDATED "9807270000Z" -- July 27, 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."
REVISION  "9807270000Z" -- July 27, 1998
DESCRIPTION
    "RFC nnnn (Proposed Standard)" -- RFC Editor to fill in
 ::= { 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))
```

```
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 implementation-dependent where the 'initial part of a PDU' starts. For SNA data, however, the starting point SHOULD be the first byte of the TH. For IP data the starting point SHOULD be the first byte of the IP header.

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,
tn3270eSrvrConfConnectivityChk	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 }

tn3270eSrvrConfConnectivityChk OBJECT-TYPE

SYNTAX INTEGER {
 timingMark(1),
 nop(2),
 noCheck(3)
 }

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   IANATn3270eAddress,
  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 IANATn3270eAddress

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,
    tn3270eSrvrStatsHCOctets    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 tn3270eSrvrStatsHCOctets 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   IANATn3270eAddress,
    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      IANATn3270eAddress
    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 2355.
```

```
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  IANATn3270ResourceType,
```

```
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 IANAtn3270ResourceType
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.

Note: In certain pathological cases, it is possible that an SSCP will send on an ACTLU for a local LU an SLU name currently represented by an entry in this table that associates it with a different local LU. In these cases the association from the newer ACTLU SHOULD be the one represented in this table."

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	IANAtn3270eAddress,
tn3270eResMapPort	Unsigned32,
tn3270eResMapElementType	IANAtn3270ResourceType,
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 IANATn3270eAddress
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 IANATn3270ResourceType
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       IANATn3270eAddress,
    tn3270eTcpConnRemPort          Unsigned32,
    tn3270eTcpConnLocalAddrType    IANATn3270eAddrType,
    tn3270eTcpConnLocalAddress     IANATn3270eAddress,
    tn3270eTcpConnLocalPort        Unsigned32,
    tn3270eTcpConnLastActivity     TimeTicks,
    tn3270eTcpConnBytesIn          Counter32,
    tn3270eTcpConnBytesOut         Counter32,
    tn3270eTcpConnResourceElement  SnaResourceName,
    tn3270eTcpConnResourceType     IANATn3270eResourceType,
    tn3270eTcpConnDeviceType       IANATn3270eDeviceType,
    tn3270eTcpConnFunctions        IANATn3270eFunctions,
    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 IANATn3270eAddress

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 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The remote port associated with a TN3270E client. The value 0 is used if the tn3270eTcpConnRemAddrType identifies an address type that does not support ports.

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 IANATn3270eAddress

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 IANATn3270ResourceType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the type of resource identified by
tn3270eTcpConnResourceElement."

::= { tn3270eTcpConnEntry 11 }

tn3270eTcpConnDeviceType OBJECT-TYPE

SYNTAX IANATn3270DeviceType

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"Indicates the device type if negotiated with the
client. A value of unknown(100) should be used as
the value of this object when a device type is not
negotiated. Refer to RFC 2355 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(1),
                noSluSession(2),
                sscpLuSession(3),  -- but no LU-LU session
                luLuSession(4),    -- but no SSCP-LU session
                sscpLuSessionAndLuLuSession(5)
            }
```

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(1)	- The true state is not known.
noSluSession(2)	- The SLU has neither an SSCP-LU nor an LU-LU session active.
sscpLuSession(3)	- The SSCP-LU session for the SLU is active, but the SLU is not currently in session with a PLU.
luLuSession(4)	- The SLU is currently in session with a PLU, but the SSCP-LU session for the LU is not active.
sscpLuSessionAndLuLuSession(5)	- 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(1),
                hostSendsUnbind(2),
            }
```

```

        hostDontAcceptConnection(3),
        outOfResource(4),
        clientProtocolError(5),
        invalidDeviceName(6),
        deviceInUse(7),
        inactivityTimeout(8),
        hostNotResponding(9),
        clientNotResponding(10),
        serverClose(11),
        sysreqLogoff(12),
        serverSpecificHexCode(13)
    }
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The last disconnect reason. A session that has not
    experienced a disconnect shall use the value unknown(1)
    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 tn3270eSrvrConfConnectivityChk
```

```
        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 tn3270eSrvrConfConnectivityChk 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 tn3270eSrvrConfConnectivityChk 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 tn3270eSrvrConfConnectivityChk 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,
        tn3270eSrvrConfConnectivityChk,
        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,
    tn3270eSrvrStatsHCOctets
}
```

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 [12]. A protocol for access control is defined in [15].

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:

- o tn3270eSrvrConfInactivityTimeout
- o tn3270eSrvrConfConnectivityChk
- o tn3270eSrvrConfActivityTimeout
- o tn3270eSrvrConfActivityInterval
- o tn3270eSrvrConfAdminStatus
- o tn3270eSrvrConfSessionTermState
- o tn3270eSrvrConfContact
- o tn3270eClientGroupSubnetMask
- o tn3270eResPoolElementType
- o tn3270eSrvrConfRowStatus
- o tn3270eSrvrPortRowStatus
- o tn3270eClientGroupRowStatus
- o tn3270eResPoolRowStatus
- o 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 tn3270eTcpConnTable.

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] Harrington D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2271, January 1998.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, Performance Systems International, March 1991
- [5] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1902, January 1996.
- [6] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1903, January 1996.
- [7] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1904, January 1996.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2272, January 1998.

- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2274, January 1998.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMPv3 Applications", RFC 2273, January 1998.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2275, January 1998.
- [16] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, RFC 854, May 1983.
- [17] Postel, J. and J. Reynolds, "Telnet Timing Mark Option", STD 31, RFC 860, May 1983.
- [18] Rekhter, J., "Telnet 3270 Regime Option", RFC 1041, January 1988.
- [19] Kelly, B., "TN3270 Enhancements", RFC 2355, June 1998.
- [20] McCloghrie, K., "TCP-MIB Definitions", RFC 2012, November 1996.
- [21] Hovey, R. and S. Bradner, "The Organizations Involved in the IETF Standards Process", BCP 11, RFC 2028, October 1996.
- [22] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [23] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [24] Krupczak, C. and J. Saperia, "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

EMail: 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
EMail: remoore@us.ibm.com

Full Copyright Statement

Copyright (C) The Internet Society (1999). 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.

