

**A Recommended Amendment to TS 3.4 - 1996  
of the Joint Committee on the NTCIP  
for Joint Adoption by AASHTO, ITE, and NEMA**

**NEMA Standards Publication TS 3.4-1996  
– Amendment 1**

---

*National Transportation Communications for ITS Protocol  
(NTCIP)  
Global Object Definitions*

---

**Draft Version 98.01.07**

**November 2, 1998**

This is a draft document, which is distributed for review and comment purposes only. You may reproduce and distribute this document within your organization, but only for the purposes of and only to the extent necessary to facilitate review and comment to the **NTCIP Coordinator**. Please ensure that all copies reproduced or distributed bear this legend. This document contains information that is subject to change.

*Published by*

**American Association of State Highway and Transportation Officials (AASHTO)**

444 North Capitol St., N.W., Suite 249  
Washington, D.C. 20001

**Institute of Transportation Engineers (ITE)**

525 School St., S.W., Suite 410  
Washington, D.C. 20024-2797

**National Electrical Manufacturers Association (NEMA)**

1300 N. 17th Street, Suite 1847

© Copyright 1998 by the National Electrical Manufacturers Association, and by the American Association of State Highway and Transportation Officials, and by the Institute of Transportation Engineers. All rights, including translation into other languages, reserved under the Universal Copyright Convention, the Berne Convention or the Protection of Literary and Artistic Works, and the International and Pan American Copyright Conventions. Do not copy without written permission of either NEMA, AASHTO, or ITE.

Rosslyn, Virginia 22209-3801

## MIB DISCLAIMER

- Add the following paragraphs after the title page of the document:

Joint AASHTO, ITE, NEMA Copyright  
and  
NTCIP Management Information Base Distribution  
**NOTICE**

© Copyright 1998 by the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). All intellectual property rights, including, but not limited to, the rights of reproduction, translation and display are reserved under the laws of the United States of America, the Universal Copyright Convention, the Berne Convention, and the International and Pan American Copyright Conventions. Except as provided, you may not copy these materials without written permission from either AASHTO, ITE or NEMA. Use of these materials does not give you any rights of ownership or claim of copyright in or to these materials.

To the extent and in the limited event these materials are distributed by AASHTO/ITE/NEMA in the form of a Management Information Base ("MIB"), AASHTO/ITE/NEMA extends the following permissions:

- (i) you may make and/or distribute unlimited copies (including derivative works) of the MIB, including copies for commercial distribution, provided that (a) each copy you make and/or distribute contains this Notice and (b) each derivative work of the MIB uses the same module name followed by "-", followed by your Internet Assigned Number Authority (IANA)-assigned enterprise number;
- (ii) use of the MIB is restricted in that the syntax field may be modified only to reflect a more restrictive subrange or enumerated values;
- (iii) the description field may be modified but only to the extent that: (a) only those bit values or enumerated values that are supported are listed; and (b) the more restrictive subrange is expressed.

These materials are delivered "AS IS" without any warranties as to their use or performance.

AASHTO/ITE/NEMA AND THEIR SUPPLIERS DO NOT WARRANT THE PERFORMANCE OR RESULTS YOU MAY OBTAIN BY USING THESE MATERIALS. AASHTO/ITE/NEMA AND THEIR SUPPLIERS MAKE NO WARRANTIES, EXPRESS OR IMPLIED, AS TO NONINFRINGEMENT OF THIRD PARTY RIGHTS, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AASHTO, ITE OR NEMA OR THEIR SUPPLIERS BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY CLAIM OR FOR ANY CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING ANY LOST PROFITS OR LOST SAVINGS, ARISING FROM YOUR REPRODUCTION OR USE OF THESE MATERIALS, EVEN IF AN AASHTO, ITE, OR NEMA REPRESENTATIVE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Some states or jurisdictions do not allow the exclusion or limitation of incidental, consequential or special damages, or the exclusion of implied warranties, so the above limitations may not apply to you.

Use of these materials does not constitute an endorsement or affiliation by or between AASHTO, ITE, or NEMA and you, your company, or your products and services.

If you are unwilling to accept the foregoing restrictions, you should immediately return these materials.

NTCIP is a trademark of AASHTO/ITE/NEMA.

## 1.2 REFERENCES

- Add the following paragraphs to the top of this clause:

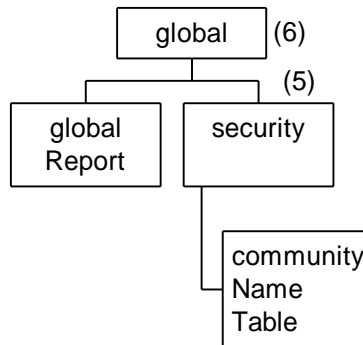
For approved revisions, contact:

NTCIP Coordinator  
National Electrical Manufacturers Association  
1300 North 17<sup>th</sup> Street, Suite 1847  
Rosslyn, VA 22209-3801

For draft revisions, which are under discussion by the relevant NTCIP Working Group, and recommended revisions of the NTCIP Joint Committee, visit the World Wide Web at <http://www.ntcip.org>.

## 1.5 OBJECT TREE

- Add a security branch off the global node as shown below:



### 2.2.1 Global Set ID Parameter

- The SYNTAX field shall be modified to read:

SYNTAX INTEGER (0..65535)

### 2.3.1 Database Creation Transaction

- Replace this clause with the following:

dbCreateTransaction OBJECT-TYPE  
SYNTAX INTEGER { normal (1),  
transaction (2),  
verify (3),  
done (6) }

ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"This object provides transaction control for device configuration. The transaction mode changes the behavior of the agent to force buffering of database objects until all related database objects have been modified. In the normal mode, SET operations to database objects may be stored in a device's database immediately with no regard to whether other changes will be made. In the transaction mode, SET operations to database objects are buffered until a verify state performs a consistency check. When the consistency check completes, the device automatically transitions to the done state where a normal or transaction command can be issued.

A database object is a user provided piece of setup information that is necessary for the proper operation of a device. It is static in nature in that the agent would never change it without direction from the management station. For example, a parameter that defines a default mode of operation would be a database object. A parameter that indicates the current state of the device would not be a database object.

The states and commands are defined as:

**NORMAL:** SET operations behave as normal SNMP SETs and can have an immediate effect on the value of any database objects used by the device. This is the default state of this object.

The only command that may be written to dbCreateTransaction while in this state is TRANSACTION. Any other values written to this object in this state shall result in an error response of 'badValue'.

**TRANSACTION:** SET operations of database objects are buffered by the agent device for later consistency checks. Standard SYNTAX checking takes place at the time of the SET operation. A transaction may consist of multiple SET operations over multiple frames.

The only commands that can be written to dbCreateTransaction while in this state are VERIFY and NORMAL. A VERIFY command will change the state to VERIFY. If a NORMAL command is received, all buffered data is discarded and the state is returned to NORMAL. Any other values written to this object when in this state shall result in an error response of 'badValue'.

**VERIFY:** Specific database objects are checked for consistency. When consistency checks are complete the device will automatically advance to the DONE state.

The state of dbCreateTransaction cannot be changed when in the VERIFY state. Any other values written to this object in this state shall result in an error response of 'badValue'.

The consistency check analyzes certain critical objects 'in context' and treats them as an interrelated whole rather than separate non-related data items. The consistency check rules are not defined in this standard. They are device and implementation specific. Where applicable, the consistency check rules are defined in application specific object definition standards. A specific implementation may add additional checks beyond those defined in the standards. As a simplified example of a consistency check, consider the following. Two objects are defined to specify the month and the day-of-month of an event. Valid values for day-of-month would normally be 1 to 31, but in the context of month 9 (September), only the values 1 to 30 are correct.

**DONE:** This state is entered automatically once consistency checks have completed in the VERIFY mode. The value of dbVerifyStatus and dbVerifyError indicate whether the consistency check found any errors.

Only two valid values can be written to dbCreateTransaction in this state; NORMAL and TRANSACTION. Any other values written to this object in this state shall result in an error response of 'badValue'.

If a NORMAL command is issued and dbVerifyStatus indicates doneWithNoError, the buffered data is transferred to the device memory and the state is returned to NORMAL. If a NORMAL command is

issued and dbVerifyStatus indicates something other than doneWithError then the buffered data is discarded and the state is returned to NORMAL.

If a TRANSACTION command is issued, regardless of dbVerifyStatus, no action takes place (the buffered data is not changed) and the TRANSACTION state is re-entered.

		COMMANDED STATE			
		<i>transaction</i>	<i>verify</i>	<i>normal</i>	<i>done</i>
CURRENT STATE	normal	transaction (1)	normal (2)	normal (2)	normal (2)
	transaction	transaction (2)	verify (3)	normal (4)	transaction (2)
	verify (7)	verify (2)	verify(2)	verify (2)	verify (2)
	done (8)	transaction (5)	done(2)	normal (6)	done (2)

Operational procedures and error responses:

- (1) Once a copy of all database objects is placed in a buffer the state is changed to transaction and error response indicates noError. If the operation fails, the state remains the same and error response indicates genErr.
- (2) No action takes place, the state remains the same, but response indicates badValue.
- (3) The state is changed to verify, a consistency check is started, and response indicates noError.
- (4). The buffered copy of all database objects is discarded, the state is changed to normal, and response indicates noError.
- (5) The buffered copy of all database objects is not changed or reloaded, the state is changed to transaction, and response indicates noError.
- (6) If dbVerifyStatus indicates doneWithNoError, then the buffered copy of all database objects is transferred to memory, the state is changed to normal and response indicates noError. If dbVerifyStatus indicates doneWithError then the buffered data is discarded, the state is changed to NORMAL, and response indicates noError.
- (7) The state will automatically change to done when the consistency check completes.
- (8) dbVerifyStatus and dbVerifyError are only valid in this state."

::= { globalDBManagement 1 }

### 2.3.2 Database Error Type Parameter

- Change the STATUS clause as follows:

STATUS deprecated

### 2.3.3 Database Error ID Parameter

- Change the STATUS clause as follows:

STATUS deprecated

### 2.3.4 Database Transaction ID Parameter

- Change the STATUS clause as follows:

STATUS deprecated

### 2.3.5 Database Make ID Parameter

- Change the STATUS clause as follows:

STATUS deprecated

- After clause 2.3.5, add the following two object definitions.

### 2.3.6 Database Verify Status Parameter

dbVerifyStatus OBJECT-TYPE

SYNTAX INTEGER { notDone (0),  
doneWithError (1),  
doneWithNoError (2) }

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object indicates the current status of verify (consistency checking) processing. The value of this object is only valid when the dbCreateTransaction object is in the Verify or Done state. If read during any other state, the value of this object is not valid but no error will be indicated."

::= { globalDBManagement 6 }

### 2.3.7 Database Verify Error Parameter

dbVerifyError OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..255))

ACCESS read-only

STATUS mandatory

DESCRIPTION

"This object contains a textual description of or a reference to an error that was found by the verify (consistency checking) processing. The value of this object is only valid when the dbCreateTransaction object is in the Done state and the dbVerifyStatus object is in the doneWithError state. If read during any other state, the value of this object is not valid but no error will be indicated."

::= { globalDBManagement 7 }

### 2.4.1 Global Time Parameter

- The DESCRIPTION field shall be modified to read:

DESCRIPTION "The current time in seconds since the epoch of 00:00:00 (midnight) January 1, 1970 UTC (a.k.a. Zulu).

### 2.4.3.2 Time Base Schedule Table

- Replace the second to the last sentences of the DESCRIPTION with:

Select the more specific event based on their MONTH settings; if the same, select the most specific DOM; if that is still the same, select the most specific DOW; if that's still the same, the first occurrence within the time base event table shall be selected.

- Add the following after the last sentence:

All entries in Time Base Schedule Table are expressed in local time and date.

#### 2.4.5 Global Local Time Differential Parameter

- This clause shall be added with the following text:

globalLocalTimeDifferential OBJECT-TYPE  
SYNTAX INTEGER (-43200..43200)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION "Indicates the number of seconds offset between local time and GMT. Positive values indicate local times in the Eastern Hemisphere up to the International Date Line and negative values indicate local times in the Western Hemisphere back to the International Date Line. If one of the daylight savings times is activated, this value will change automatically at the referenced time. For example, Central Standard Time (CST) is -21600 and Central Daylight Time (CDT) is -18000."

::= { globalTimeManagement 4 }

#### 2.5.2.3 Event Log Configuration Mode Parameter

- Add the periodic mode to eventLogConfigMode as follows:

eventConfigMode OBJECT-TYPE  
SYNTAX INTEGER { other (1),  
onChange (2),  
greaterThanValue (3),  
smallerThanValue (4),  
hysteresisBound (5),  
periodic (6) }

ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"This object specifies the mode of operation for this event. All checks and entries to the table must occur within one second of the condition becoming true. The modes are defined as follows:

VALUE	DESCRIPTION
-------	-------------

onChange	create a log entry when value referenced by the eventTypeOID changes
greaterThanValue	create a log entry when the object value becomes greater than the value referenced to by the eventCompareValue object, if this value is exceeded for the amount of time specified in the eventConfigCompareValue2 object (in tenth of seconds) and this value is greater than zero (0). A value of zero (0) for eventConfigCompareValue2 indicates immediate logging.

**smallerThanValue** create a log entry when the object value becomes less than the value referenced to by the eventCompareValue object, if this value is exceeded for the amount of time specified in the eventConfigCompareValue2 object (in tenth of seconds) and this value is greater than zero (0). A value of zero (0) for eventConfigCompareValue2 indicates immediate logging.

**hysteresisBound** creates a log entry when the object value becomes either less than the lowerbound value or greater than the upperbound value. The lowerbound value is the lower value of the eventConfigCompareValue- and the eventConfigCompareValue2-objects, the upperbound is the other value.

**periodic** create a log entry every x seconds, where x is defined by the value stored in eventConfigCompareValue. The values stored in eventConfigCompareValue2 and eventConfigCompareOID are ignored in this mode. "

::= { eventLogConfigEntry 3 }

#### 2.5.2.5 Event Log Configuration Compare Value 2 Parameter

- The *DESCRIPTION* field shall be replaced with the following text:

"If the eventConfigMode is set to hysteresisBound, this object specifies the second comparison value for the hysteresis. If the eventConfigMode is set to greaterThanValue or smallerThanValue, this object specifies the time (in tenths of seconds) for which the comparison must be true prior to the event condition becoming true. If the eventConfigMode is set to onChange or periodic, the value of this object shall be ignored."

#### 2.5.2.6 Event Log Configuration Compare Object Identifier Parameter

- The *DESCRIPTION* shall be modified to read:

"This object contains the object identifier which references the value against which the comparison is made. If the eventConfigMode is set to periodic, the value of this object shall be ignored."

#### 2.5.4 Event Log Table

- Change the syntax of eventLogTime in the EventLogEntry definition as shown.

```
EventLogEntry ::= SEQUENCE {  
    eventLogClass    INTEGER,  
    eventLogNumber  INTEGER,  
    eventLogID      INTEGER,  
    eventLogTime    Counter,  
    eventLogValue   Opaque }
```

#### 2.5.4.3 Event Log ID Parameter

- The *ACCESS* shall be modified to read:

ACCESS read-only

#### 2.5.4.5 Event Log Value Parameter

- The *DESCRIPTION* shall be modified to read:

"The value of this object is set to the value referenced by the eventConfigLogOID of the associated eventLogID when the event was logged. Its length is variable."

#### 2.5.6 Event Log Table

- Change the syntax of eventClassClearTime in the EventClassEntry definition as shown.

```
EventClassEntry ::= SEQUENCE {  
    eventClassNumber      INTEGER,  
    eventClassLimit       INTEGER,  
    eventClassClearTime   Counter,  
    eventClassDescription OCTET STRING,  
    eventClassNumRowsInLog INTEGER }
```

### SECTION 2

- On page 25, add clause 2.8 just before the "END" statement:

#### 2.8 SECURITY NODE

security OBJECT IDENTIFIER ::= global 5  
-- This node is an identifier used to group all objects related to the  
-- assignment of community names and the access rights they provide.

##### 2.8.1 Community Name Administrator Parameter

communityNameAdmin OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE(8..16))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"This object is the community name that must be used to specifically gain access to information under the security node. A message with this value in the community name field of an SNMP message has user read-write access to the security node objects and all other objects implemented in the device. The syntax is defined as an OCTET STRING and therefore any character can have a value of 0..255."  
DEFVAL { "administrator" }  
::= { security 1 }

##### 2.8.2 Maximum Community Names Parameter

communityNamesMax OBJECT-TYPE  
SYNTAX INTEGER (1..255)  
ACCESS read-only  
STATUS mandatory  
DESCRIPTION  
"This object specifies the maximum number of rows that are implemented in the community name table."  
::= { security 2 }

##### 2.8.3 Community Names Table

communityNameTable OBJECT-TYPE

SYNTAX SEQUENCE OF CommunityNameTableEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"This table defines the community names that can appear in the community name field of the SNMP message and access privileges associated with that community name."  
::= { security 3 }

communityNameTableEntry OBJECT-TYPE  
SYNTAX CommunityNameTableEntry  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"This is the row index of information in the community name table."  
INDEX { communityNameIndex }  
::= { communityNameTable 1 }

CommunityNameTableEntry ::= SEQUENCE  
    { communityNameIndex INTEGER,  
      communityNameUser OCTET STRING,  
      communityNameAccessMask GAUGE  
    }

### 2.8.3.1 Community Name Index Parameter

communityNameIndex OBJECT-TYPE  
SYNTAX INTEGER (1..255)  
ACCESS not-accessible  
STATUS mandatory  
DESCRIPTION  
"This object defines the row index into the communityNameTable. This value shall not exceed the communityNamesMax object value."  
::= { communityNameTableEntry 1 }

### 2.8.3.2 User Community Name Parameter

communityNameUser OBJECT-TYPE  
SYNTAX OCTET STRING (SIZE(6..16))  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION  
"This object defines a community name value that a security administrator can assign user read-write access to information (other than security) in a device. A message with this value in the community name field of an SNMP message has user access rights as defined in the communityNameAccessMask. The syntax is defined as an OCTET STRING and therefore any character can have a value of 0..255."  
DEFVAL { "public" }  
::= { communityNameTableEntry 2 }

### 2.8.3.3 User Community Name Mask Parameter

communityNameAccessMask OBJECT-TYPE  
SYNTAX GAUGE (0..4294967295)  
ACCESS read-write  
STATUS mandatory  
DESCRIPTION

"This object defines a 32 bit mask that can be used to associate 'write access' with a community name. A value of 0x00000000 grants the community name user read-only access and overrides any individual object's read-write access clause. A value of 0xFFFFFFFF grants the community name user read-write access and an individual object's read-write access clause applies. Values other 0x00000000 and 0xFFFFFFFF are implementation specific and may limit viewing and/or accessing the information in a device."

DEFVAL { 4294967295 }  
::= { communityNameTableEntry 3 }

### Section 3

#### 3.2 DATABASE MANAGEMENT CONFORMANCE GROUP

- Modify the Database Management Conformance Group table as follows:

Object or Group Name	Reference
dbCreationTransaction	TS 3.4
dbVerifyStatus	TS 3.4-1996 Amendment 1
dbVerifyError	TS 3.4-1996 Amendment 1

#### 3.3 TIME MANAGEMENT CONFORMANCE GROUP

- Modify the Time Management Conformance Group table as follows:

Object or Group Name	Reference
globalTime	TS 3.4
globalDaylightSaving	TS 3.4
globalLocalTimeDifferential	TS 3.4-1996 Amendment 1

- Insert the following after clause 3.7:

#### 3.8 SECURITY CONFORMANCE GROUP

The Security Group consists of those mandatory global objects related to community name parameters. The Security Group shall consist of the following objects:

Object or Group Name	Reference
communityNameAdmin	TS 3.4
communityNamesMax	TS 3.4
communityNameTable	TS 3.4
communityNameIndex	TS 3.4
communityNameUser	TS 3.4
communityNameMask	TS 3.4

### Section 4

- Revise the Conformance Table on page 31 as shown.

Table 4-1  
CONFORMANCE TABLE

<b>Conformance Group</b>	<b>Reference</b>	<b>Conformance Requirement</b>
Configuration	TS 3.4	mandatory
Security	TS 3.4	mandatory
Database Management	TS 3.4	optional
Time Management	TS 3.4	optional
Time Base Event Schedule	TS 3.4	optional
Report	TS 3.4	optional
STMP	TS 3.4	optional
PMPP	TS 3.4	optional

