A Joint Standard of AASHTO, ITE, and NEMA

NTCIP 8004 version v02

National Transportation Communications for ITS Protocol

Structure and Identification of Management Information (SMI)

published in June 2010

This Adobe® PDF copy of an NTCIP standard is available at no-cost for a limited time through support from the U.S. DOT / Research and Innovative Technology Administration, whose assistance is gratefully acknowledged.

Published by

American Association of State Highway and Transportation Officials (AASHTO)
444 North Capitol Street, N.W., Suite 249
Washington, D.C. 20001

Institute of Transportation Engineers (ITE)
1099 14th Street, N.W., Suite 300 West
Washington, D.C. 20005-3438

National Electrical Manufacturers Association (NEMA)
1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3801

file version 8004 v02.17r rita-2010a

NOTICES

Copyright Notice

PDF File License Agreement

Data Dictionary and MIB Distribution Permission

b) the copies or derivative works are not made part of the standards publications or works offered by other standards developing organizations or publishers or as works-for-hire not associated with commercial hardware or software products intended for field implementation;
c) use of the DD or MIB is restricted in that the syntax fields may be modified only to reflect a more restrictive subrange or enumerated values;
d) the description field may be modified but only to the extent that: 1) only those bit values or enumerated values that are supported are listed; and 2) 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 / 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 / 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 a given user.

Use of these materials does not constitute an endorsement or affiliation by or between AASHTO, ITE, or NEMA and the user, the user’s company, or the products and services of the user’s company.

If the user is unwilling to accept the foregoing restrictions, he or she should immediately return these materials.

Content and Liability Disclaimer

The information in this publication was considered technically sound by the consensus of persons engaged in the development and approval of the document at the time it was developed. Consensus does not necessarily mean that there is unanimous agreement among every person participating in the development of this document.

AASHTO, ITE, and NEMA standards and guideline publications, of which the document contained herein is one, are developed through a voluntary consensus standards development process. This process brings together volunteers and seeks out the views of persons who have an interest in the topic covered by this publication. While AASHTO, ITE, and NEMA administer the process and establish rules to promote fairness in the development of consensus, they do not write the document and they do not independently test, evaluate, or verify the accuracy or completeness of any information or the soundness of any judgments contained in their standards and guideline publications.

AASHTO, ITE, and NEMA disclaim liability for any personal injury, property, or other damages of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, application, or reliance on this document. AASHTO, ITE, and NEMA disclaim and make no guaranty or warranty, express or implied, as to the accuracy or completeness of any information published herein, and disclaims and makes no warranty that the information in this document will fulfill any of your particular purposes or needs. AASHTO, ITE, and NEMA do not undertake to guarantee the performance of any individual manufacturer or seller’s products or services by virtue of this standard or guide.

In publishing and making this document available, AASHTO, ITE, and NEMA are not undertaking to render professional or other services for or on behalf of any person or entity, nor are AASHTO, ITE, and NEMA undertaking to perform any duty owed by any person or entity to someone else. Anyone using this document should rely on his or her own independent judgment or, as appropriate, seek the advice of a

ACKNOWLEDGEMENTS

NTCIP 8004 v02 was prepared by the NTCIP Base Standards and Profiles 2 (BSP2) Working Group (WG), which is a subdivision of the Joint Committee on the NTCIP. The Joint Committee is organized under a Memorandum of Understanding among the American Association of State Highway and Transportation Officials (AASHTO), the Institute of Transportation Engineers (ITE), and the National Electrical Manufacturers Association (NEMA). The Joint Committee on the NTCIP consists of six representatives from each of the standards organizations, and provides guidance for NTCIP development.

At the time that NTCIP 8004 v02 was prepared, the following individuals were members of the NTCIP BSP2 WG:

- Ralph Boaz
- Bob De Roche (Chair)
- Bud Kent
- Jim Mahugh
- Greg Mizell
- Alex Mousadi
- Peter Ragsdale
- Robert Rausch
- Joerg “Nu” Rosenbohm

Other individuals providing input to NTCIP 8004 v02 include:

- Blake Christie
- Kenneth Vaughn

In addition to the many volunteer efforts, recognition is also given to those organizations that supported the efforts of the working group by providing funding for NTCIP 8004 v02, providing travel support, and submitting comments, including:

- U.S. Department of Transportation, Research and Innovative Technology Administration
- Econolite Control Products
- Noblis
- Pillar Consulting
- Robert De Roche Consulting
- Siemens ITS
- Telvent Farradyne
- Transcore
- Trevilon
- US Traffic
- URS Greiner
- Washington State Department of Transportation

FOREWORD

NTCIP 8004 v02 specifies a set of rules and protocols for organizing, describing, and defining transportation management information to be exchanged between transportation management applications and transportation equipment. NTCIP 8004 v02 defines requirements that are applicable to all NTCIP devices that exchange data in that transportation environment.

In addition, NTCIP 8004 v02 contains mandatory requirements that are applicable to all devices claiming conformance to NTCIP 8004 v02. NTCIP 8004 v02 also contains optional and conditional requirements, which may be applicable to a specific environment in which a device is used. NTCIP 8004 v02 has five annexes, but only Annex A is normative. NTCIP 8004 v02 uses only metric units.

NTCIP 8004 v02 is also an NTCIP Process, Control, and Information Management document that defines the practices and policies used by the NTCIP Joint Committee in developing and maintaining NTCIP standards publications. Specifically, NTCIP 8004 v02 is applicable to the NTCIP 1200 series and other NTCIP standards that deal with device data dictionaries.

The following keywords apply to NTCIP 8004 v02: AASHTO, Dynamic Objects, ITE, NEMA, NTCIP, SMI, SNMP, STMP, process and control standard.

User Comment Instructions

The term "User Comment" includes any type of written inquiry, comment, question, or proposed revision, from an individual person or organization, about any part of this standards publication's content. A "Request for Interpretation" of this standards publication is also classified as a User Comment. User Comments are solicited at any time. In preparation of this NTCIP standards publication, input of users and other interested parties was sought and evaluated.

All User Comments are referred to the committee responsible for developing and/or maintaining this standards publication. The committee chairperson, or their designee, may contact the submitter for clarification of the User Comment. When the committee chairperson or designee reports the committee's consensus opinion related to the User Comment, that opinion is forwarded to the submitter. The committee chairperson may report that action on the User Comment may be deferred to a future committee meeting and/or a future revision of the standards publication. Previous User Comments and their dispositions may be available for reference and information at www.ntcip.org.

A User Comment should be submitted to:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3801
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standards Publication number and version:
Page:
Section, Paragraph, or Clause:
Comment:
Editorial or Substantive?:
Suggested Alternative Language:

Please include your name, organization, and address in your correspondence.

Approvals

NTCIP 8004 v02 was separately balloted and approved by AASHTO, ITE, and NEMA after recommendation by the Joint Committee on the NTCIP. Each organization has approved NTCIP 8004 v02 as the following standard type, as of the date:

AASHTO—Standard Specification; May 2010
ITE—Software Standard; June 2010
NEMA—Standard; December 2009

History

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began development of the NTCIP. The Transportation Section's purpose was in response to user needs to include standardized systems communication in the NEMA TS 2 standard, *Traffic Controller Assemblies*. Under the guidance of the Federal Highway Administration's NTCIP Steering Group, the NEMA effort was expanded to include the development of communications standards for all transportation field devices that could be used in an Intelligent Transportation Systems (ITS) network.

In September 1996, an agreement was executed among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain the NTCIP standards. The Joint Committee on NTCIP has sponsored the development of a number of standards that define device data dictionaries. After the NTCIP Class B Profile was published, the Joint Committee on the NTCIP determined that the communications profiles should be modular to meet the varied needs of different communication environments. The Joint Committee on the NTCIP formed both the Base Standards and Protocols Working Group (BSP WG) and the Profiles WG. After reorganization, the two WGs merged to form the Base Standards and Profiles (BSP2) WG. The first meeting of the working group was in January 1999.

During early stages of development, portions of NTCIP 8004 v02 were part of NTCIP 1101:1996, which was also numbered and referenced as NEMA TS 3.2-1996. However, to provide a more systematic approach to an organized numbering scheme, and to reflect the joint copyright of AASHTO, ITE, and NEMA, the current designation is NTCIP 8004 v02.

NTCIP 8004 v02 was originally part of NTCIP 1101:1996, *Simple Transportation Management Framework*. In July 1999, the Joint Committee on the NTCIP approved a work item to separate a predecessor of NTCIP 8004 v02 into a stand-alone document. In January 2000, the BSP WG submitted a work plan and initiated development of NTCIP 8004 SMI.

NTCIP 8004 v02.04, May 2006—NTCIP Joint Committee accepted the User Comment Draft, which included changes harmonized to NTCIP 8005 v01.18. The references to the <DescriptiveName> and <DataType> subfields were deleted, and a new <Object identifier> subfield was added to reflect the full Object identifier value from the root node.

NTCIP 8004 v02.07, January 2007—NTCIP Joint Committee further accepted the addition of a new Section 2.6 in the User Comment Draft, in accordance with Technical Coordination Forum (TCF) recommendations on backward compatibility. In January 2007, Standards Bulletin B0118 issued v02.07 for review and comment.

NTCIP 8004 v02.12, October 2007—NTCIP Joint Committee accepted v02.12 as a Recommended Standard, with instructions for two revisions. January 2008—Revised definition of ‘deprecated’ per Joint Committee, and updated figures to reflect new nodes.

NTCIP 8004 v02.15, September 2009—NTCIP Standards Bulletin B0134 issued for ballot and approval, after v02.13 to v02.14 edited during March 2008 to July 2009 for object STATUS definitions and related content.

NTCIP 8004 v02.16, March 2010—Edited prior to publication.

Compatibility of Versions

To distinguish NTCIP 8004 v02 (as published) from previous drafts, NTCIP 8004 v02 also includes NTCIP 8004 v02.17 on each page header. All NTCIP standards publications have a major and minor version number for configuration management. The version number syntax is "v00.00a," with the major version number before the period, and the minor version number and edition letter (if any) after the period.

NTCIP 8004 v02 is designated, and should be cited as, NTCIP 8004 v02. Anyone using NTCIP 8004 v02 should seek information about the version number that is of interest to them in any given circumstance. The MIB, the PRL, and the PICS should all reference the version number of the standards publication that was the source of the excerpted material.

Compliant systems based on later, or higher, version numbers MAY NOT be compatible with compliant systems based on earlier, or lower, version numbers. Anyone using NTCIP 8004 v02 should also consult NTCIP 8004 v02 for specific guidelines on compatibility.

NOTE—The reference to NTCIP 8004 v02 is template information. The circular nature of the reference is recognized.

CONTENTS

Page

FIGURES

Section 1
GENERAL

1.1 SCOPE

NTCIP 8004 v02 specifies a set of rules and protocols for organizing, describing, and defining transportation management information to be exchanged between transportation management applications and transportation equipment. NTCIP 8004 v02 defines the Structure and Identification of Management Information (SMI) used in transportation-related devices. NTCIP 8004 v02 is applicable to the NTCIP 1200 series and other NTCIP standards that deal with device data dictionaries.

NOTE—NTCIP 8004 v02 relies on widely accepted conventions, generally designated as “SMIv1” and defined in IAB STD 16, as well as some elements of SMIv2, as defined in IAB STD 58.

1.2 REFERENCES

Normative references contain provisions that, through reference in this text, constitute provisions of NTCIP 8004 v02. Other references in NTCIP 8004 v02 might provide a complete understanding of the entire protocol and the relations between all parts of the protocol. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standard listed.

1.2.1 Normative References

IAB STD 16	(RFC 1155: 1990, *Structure and Identification of Management Information for TCP/IP-based Internets*; RFC 1212: 1991, *Concise MIB Definitions*)
IAB STD 58	(RFC 2578:1999, *Structure of Management Information Version 2 (SMIv2))*
ISO/IEC 8824-1:2002	*Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation*
ISO/IEC 8824-2:2002	*Information technology—Abstract Syntax Notation One (ASN.1): Information object specification*
ISO/IEC 8824-3:2002	*Information technology—Abstract Syntax Notation One (ASN.1): Constraint specification*
ISO/IEC 8824-4:2002	*Information technology—Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specifications*
ISO 1000:1992	*SI Units and recommendations for use of their multiples and of certain other units*

1.2.2 Other References

IAB STD 8	(*RFC 0854: 1983, Telnet Protocol Specification*, and RFC 0855: 1983, *Telnet Options Specifications*; J. Postel, J. Reynolds)
AASHTO / ITE / NEMA NTCIP 1103 v02	*Transportation Management Protocols (TMP)* publication July 2010
AASHTO / ITE / NEMA NTCIP 1201:2005	*Global Object (GO) Definitions—version 02* published October 2005
AASHTO / ITE / NEMA NTCIP 1202:2005	*Object Definitions for Actuated Traffic Signal Controller (ASC) Units—version 02* published November 2005
AASHTO / ITE / NEMA NTCIP 7001	*Assigned Numbers Summary—Part 1*
AASHTO / ITE / NEMA NTCIP 8005 v01	*Procedures for Creating Management Information Base (MIB) Files* published June 2010
NEMA TS_2-2003	*Traffic Controller Assemblies with NTCIP Requirements* 2003

Perkins, D; McGinnis, E., *Understanding SNMP MIBs*, Prentice-Hall, Inc., 1997, ISBN 0-13-437708-7

Stallings, William, *SNMP, SNMPv2, and CMIP: The Practical Guide to Network-Management Standards*, Massachusetts, Addison-Wesley Publishing Company, 1993, ISBN 0-201-63331-0

Larmouth, John, *ASN.1 Complete*, Academic Press, a Harcourt Science and Technology Company, May 1999, ISBN 0-12233-435-3, http://www.oss.com/asn1/booksintro.html (October 9, 2000)

Dubuisson, Olivier, ASN.1 Communication between Heterogeneous Systems, June 5, 2000, ISBN 0-12-6333361-0, http://www.oss.com/asn1/bookintro.html (October 9, 2000)

Booch, Grady, Rumbaugh, James, Jacobson, Ivar, *Unified Modeling Language User Guide*, September 30, 1998, ISBN 0-20157-168-4

UML basics: An introduction to the Unified Modeling Language, www-106.ibm.com/developerworks/rational/library/769.html#N10090.

1.2.3 Contact Information

1.2.3.1 NTCIP Standards

For revision information on NTCIP 8004 v02, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3801
e-mail: ntcip@nema.org

For draft revisions of NTCIP 8004 v02, recommended revisions of the NTCIP Joint Committee, and all other NTCIP standards publications, visit www.ntcip.org .

1.2.3.2 IAB/RFC Documents

Internet Architecture Board (IAB) documents, or Request for Comment (RFC) electronic documents are available from several repositories on the World Wide Web, or by “anonymous” File Transfer Protocol (FTP) with several hosts. Browse or FTP to:

Internet Architecture Board (IAB)
www.rfc-editor.org
www.rfc-editor.org/repositories.html

1.2.3.3 ISO/IEC Standards

Members of ISO maintain registers of currently valid ISO/IEC International Standards. For the U.S., the member of ISO is the American National Standards Institute (ANSI), which may be contacted as follows:

ANSI
11 West 42nd Street, 13th Floor
New York, NY 10036
(212) 642-4900
http://global.ihs.com

1.3 DEFINITIONS AND ACRONYMS

For the purposes of NTCIP 8004 v02, the following terms and definitions (with acronyms, where appropriate) apply. For terms not defined in this section, English words are used in accordance with their definitions in the latest edition of *Webster's New Collegiate Dictionary*. Electrical and electronic terms not defined in this section or in *Webster's New Collegiate Dictionary* are used in accordance with their definitions in IEEE Std 100-2000.

agent	The entity that receives commands and transmits responses to the received commands.
block object	A grouping of individual objects primarily for the purpose of referring to the group with a single name and identifier.
compatibility	The ability of two or more systems or components to exchange information. [from IEEE Standards Dictionary, Glossary of Terms and Definitions]
data	Information before it is interpreted.
datagram	A self-contained unit of data transmitted independently of other datagrams.
deprecated	In the context of a MIB, “deprecated” is an object STATUS value that indicates the object is valid in limited circumstances, but has been replaced by another. NOTE—This definition is modified from “Understanding SNMP MIBS.” To maintain multi-version interoperability (backward compatibility) for legacy implementations, objects with a STATUS value of “deprecated” may require support. When necessary to support legacy implementations, required support for objects with a STATUS value of “deprecated” is indicated using the PICS or Protocol Requirements List (PRL).

file	A grouping of individual or block database objects into a single sequence of bytes that can be referred to by file operations. NOTE—A file exists nominally in a directory, and can have an associated path.
Intelligent Transportation Systems (ITS)	The application of advanced information processing and communications, sensing, and control technologies to surface transportation with the objective of promoting more efficient use of the existing highway and transportation network, increasing safety and mobility, and decreasing environmental impacts.
International Organization for Standardization (ISO)	An international standards organization. NOTE—ANSI is the primary interface to ISO within the United States. Often thought to be International Standards Organization, because of its acronym ISO.
Internet	A large collection of connected networks, primarily in the United States, running the Internet suite of protocols. NOTE—Sometimes referred to as the *DARPA Internet, NSF/DARPA Internet*, or the *Federal Research Internet.*
Internet Protocol (IP)	The network protocol offering a connection-less mode network service in the Internet suite of protocols.
interoperability	The ability of two or more systems or components to exchange information and use the information that has been exchanged. NOTE--From IEEE *Standards Dictionary, Glossary of Terms and Definitions*.
IP address	A 32-bit quantity used to represent a point of attachment in an internet.
Management Information Base (MIB)	A structured collection or database of related managed objects defined using Abstract Syntax Notation One (ASN.1).
manager	The entity that sends commands to entities and processes their responses.
mandatory	In the context of a MIB, an object STATUS value that indicates the object is valid. NOTE—In SMIv1, “mandatory” also indicates implementation is required for conformance. This definition is modified from “Understanding SNMP MIBS.”
National Transportation Communications for ITS Protocol (NTCIP)	A family of protocols that provide common control and data collection services as well as accommodating various system topologies and data routing duties. NOTE—NTCIP is designed to support not only currently deployed systems, but also new systems and technologies as they become available.
network	A collection of subnetworks connected by intermediate systems and populated by end systems.
network layer	That portion of an OSI system responsible for data transfer across the network, independent of both the media comprising the underlying subnetworks and the topology of those subnetworks.

object	An instance of object type is a data structure that can be used to describe the attribute or properties of a single data element or a group of data elements, such as a table.
OBJECT IDENTIFIER	A unique name (identifier) that is associated with each type of object in a MIB that is a defined ASN.1 type.
object type	A data structure used to describe the attribute or properties of an object or a group of objects.
OBJECT-TYPE	A macro used to define the meta attributes of an object in an SNMP MIB.
obsolete	In the context of a MIB, an object STATUS value that indicates the definition is no longer valid, was found to be flawed, was redundant, or was not useful. NOTE—In the next (or some future) edition of a standard, the object or group with a STATUS value of “obsolete” may be removed. This definition is modified from “Understanding SNMP MIBS.”
protocol	A specific set of rules, procedures, and conventions defining the format and timing of data transmissions between devices that are accepted and used to understand each other.
Simple Network Management Protocol (SNMP)	A communications protocol developed by the Internet Engineering Task Force (IETF), used for configuration and monitoring of network devices.
Simple Transportation Management Framework (STMF)	Describes the organization of information within devices and methods of retrieving or modifying any information within the device. NOTE—STMF also explains how to generate and use computer-readable information organization descriptions.
subnet/ subnetwork	A physical network within a network on which all devices share the same physical media.
trap	An event within an SNMP agent that generates a message for a management station.

1.4 OTHER ABBREVIATIONS AND ACRONYMS

Other abbreviations and acronyms used in NTCIP 8004 v02 are defined as follows:

ASC	Actuated Signal Controller
ASCII	American National Standard Code for Information Interchange
ASN.1	Abstract Syntax Notation One
IAB STD	Internet Architecture Board Standard
IANA	Internet Assigned Numbers Authority
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force

IP	Internet Protocol
MVI	Multi-Version Interoperability (formerly Backward Compatibility)
NVT	Network Virtual Terminal
OER	Octet Encoding Rules
OSI	Open Systems Interconnection
PDU	Protocol Data Unit
PICS	Protocol (or Profile) Implementation Conformance Specification
PRL	Protocol Requirements List
RFC	Request for Comments
SMI	Structure and Identification of Management Information
STMP	Simple Transportation Management Protocol
TCP	Transport Control Protocol
TMIB	Transportation Management Information Base
UML	Unified Modeling Language

Section 2
CONFORMANCE

2.1 INTRODUCTION

NTCIP 8004 v02 specifies a set of rules and protocols for organizing, describing, and defining transportation management information to be exchanged between transportation management applications and/or transportation equipment such that they interoperate with each other. NTCIP 8004 v02 is based on the Internet Architecture Board (IAB) Standard (STD) 16 and ISO/IEC 8824. NTCIP 8004 v02 intends that a high level of compatibility, including interoperability, be maintained between traffic management systems and field ITS devices.

NTCIP 8004 v02 covers four major areas:

a) Organization and Object Identification
b) Object Specifications
c) Meta Attributes
d) Management Information Bases (MIBs)

Within NTCIP, each piece of information is described by a number of characteristics. These characteristics are also referred to as meta attributes. Most of the characteristics are defined by RFC 1212 (IAB STD 16) and are used to represent the logical concept of an “object” such that it can be used with NTCIP protocols. The specific set of characteristics used varies based on the type of object being defined. The general format for documenting these characteristics is known as the **base object specification** and is discussed in Section 2.3.1. This general definition is then further refined for three specific cases: a simple object, a block object, or a dynamic object.

The simple object specification is used to document how simple atomic pieces of information can be exchanged by Simple Network Management Protocol (SNMP) or Simple Fixed Management Protocol (SFMP). The details for defining simple objects are discussed in Section 2.3.2.

In some cases, it is desirable to predefine groups or structures of data that can be referenced by a single identifier. This NTCIP representation is specified within the **block object specification**. A block object is treated as a single object in terms of identification and exchange but the values within the block reflect more elemental objects (e.g., simple objects). The details for defining block objects are discussed in Section 2.3.3.

The third case is given by the **dynamic object**. A block object is always a predefined sequence of variables. In other words, the variables that make up the sequence, and the order in which they appear, are defined by a standard and do not change unless the standard is changed. Simple Transportation Management Protocol (STMP) extends this concept to allow the structure of a group of objects to be defined at run-time. These objects can be defined and redefined at any time during the operation of the system and do not require a change to the standard. The complete specification for defining dynamic objects and how they are used is covered in NTCIP 1103 v02 Section 5.

In each of the preceding cases, it is necessary for each processing entity involved in an exchange to have some predefined knowledge about what is or can be transferred. The means of sharing this information is through the use of a **MIB**. A MIB is a computer-readable file that contains the definition of individual objects. The details of defining a MIB are covered in Section 2.5. It should be emphasized that definitions of block object contents are external to a MIB.

Whereas a MIB is a computer-readable file, users may need additional information to understand, handle, and process objects. Users also need to know general configuration information about a MIB. The description of these NTCIP-defined meta attribute requirements are in Section 2.4.

2.1.1 Message Meta Model

The relationships among many of the different terms of a message can be depicted in the Unified Modeling Language (UML), as shown in Figure 1.

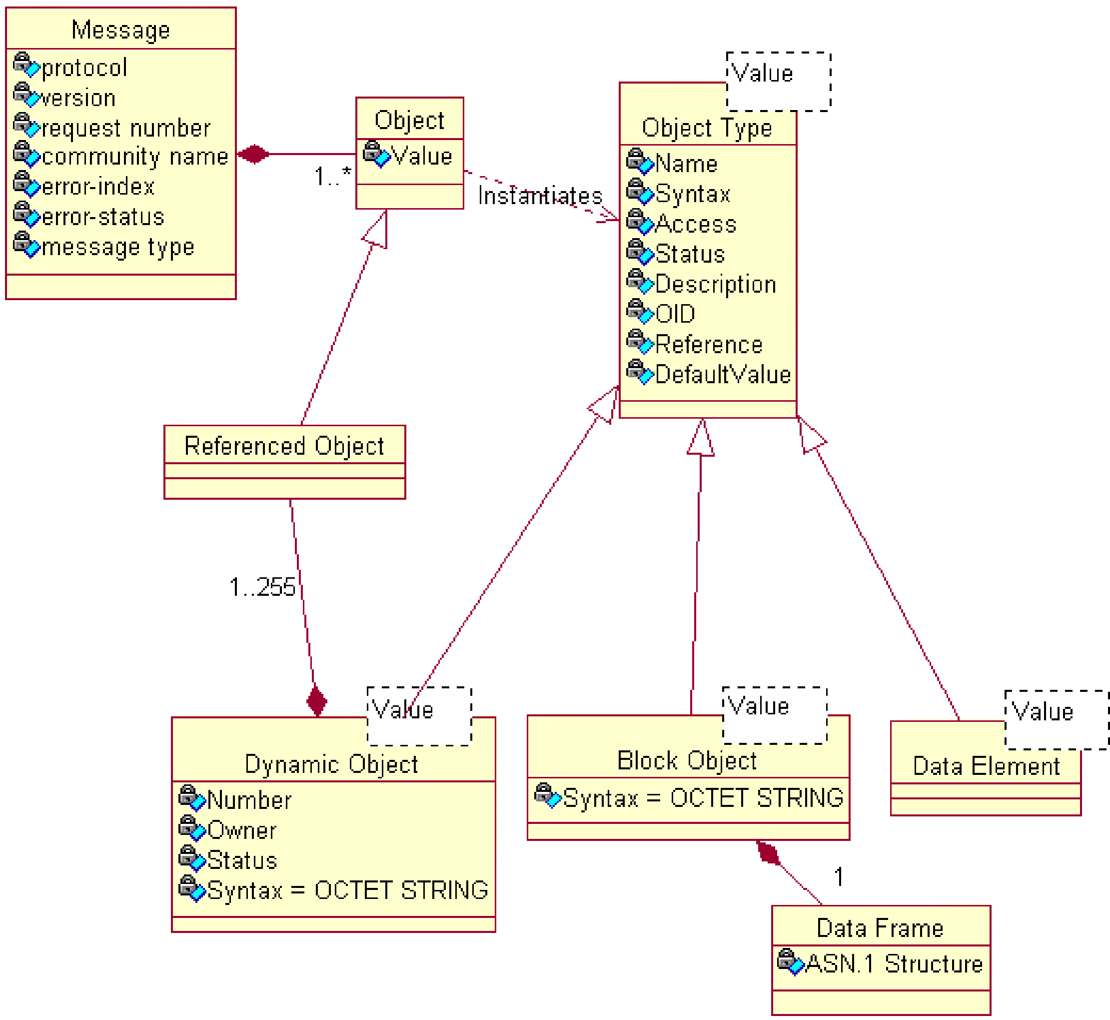

Figure 1 Relationships Among Terms

Starting in the upper-left corner of Figure 1, a message is the logical structure behind the Data Packet (i.e., byte-stream) that is exchanged by protocol entities. It consists of a variety of information, such as the request number, community name, and the type of message along with one or more Objects. (Although in the case of STMP and SFMP, the message is limited to containing a single Object.)

An Object is an instance of an Object Type. An Object Type is the structure that is defined in a MIB. For example, a MIB might define a table that contains information. The MIB only defines the columns of the table and which columns are used as indices. Each column would be an Object Type, and each cell in the table would be an Object, i.e., an instance of the respective Object Type as defined for that column.

There are three specializations of Object Type: Data Element, Block Object, and Dynamic Object. The simplest form of an Object is the Data Element. A Data Element is a single piece of information that can be represented using normal SNMP syntax, for example, an integer.

However, at times, there is a need to deal with more complex structures than a single piece of information. The Block Object provides this. A Block Object is the byte-stream (i.e., serialization) representing a defined structure. From the perspective of the message, a Block Object is merely an embedded byte-stream, but the end application on either end of the communications link can translate this byte-stream into a more complex structure. However, the structure is required to be statically defined and implemented within each end application.

A Dynamic Object takes this concept one step further. It allows the content of the data structure to be defined at run-time rather than forcing a static definition at design-time. A management station may configure a Dynamic Object to reference any combination of other Objects stored within the device (with few exceptions). The referencing mechanism is termed a Referenced Object. While this mechanism only allows for simple sequences of data, it still provides the management station with a powerful tool in managing communications over a link. However, Dynamic Objects are only accessible via STMP.

Annex D provides additional material on UML.

2.2 ORGANIZATION AND OBJECT IDENTIFICATION

As defined in the SMI MIB in Annex A, NTCIP data is organized through the use of a global naming tree. The naming tree was created by the ISO-OSI community and is commonly referred to as the ISO Naming Tree. The tree consists of three root nodes, each connected to a number of sub-nodes falling underneath the root node. Each sub-node may, in turn, have sub-nodes of its own. Each node, whether a root node or a sub-node, is managed by some organization. For example, the three root nodes are managed by ISO, the International Telecommunication Union Telecommunication Standardization Sector (ITU-T), and jointly by ISO and ITU-T. The manager of any node may delegate the management responsibility for any sub-node underneath its branch. In this case, the delegated node is termed a subtree. Just as any sub-node may have its own set of sub-nodes, subtrees can have their own subtrees. This structure may continue to any level of depth to meet the needs of the managing organization. Each node on this tree is assigned both a label and an integral number.

The label assigned to any node is the OBJECT DESCRIPTOR for that node. An OBJECT DESCRIPTOR is a user-friendly textual name used to convey some meaning and semantic understanding of the object being described.

The sequence of integral numbers starting from a root node and traversing the tree to a subject node is termed to be the OBJECT IDENTIFIER for that node. Through the proper management of the sub-nodes at each level in the tree, one is able to obtain a globally unique identifier.

The unique identifier can be used for any purpose for which an identifier may be useful. For example, most standards organizations have an OBJECT IDENTIFIER assigned to it for the purposes of identification. The tree can also be used for managing groups of related data. For example, all objects related to an Actuated Signal Controller are organized under a node defined as ‘asc’. In short, these attributes are a means for identifying some object, regardless of the semantics associated with the object.

Central to the notion of the OBJECT IDENTIFIER is the understanding that administrative control of the meanings assigned to the nodes may be delegated as one traverses the tree.

2.2.1 Naming Tree Administrative Nodes

The first two root nodes of the naming tree are administered by ITU-T (itu-t) and ISO (iso). The third node is jointly administered by ISO and ITU-T (joint-iso-itu-t).

NOTE—Until 1991, the U.S. name-registration authority conducted its business under the {iso(1) member-body(2) us(840)} arc, registering names for ANSI standards, private organizations with U.S. national standing, and the names of U.S. states and "state equivalents." In 1991, changes in the registration authority procedures standard ISO/IEC 9834 (Rec. ITU-T X.660) invalidated this procedure, requiring private organization names with national standing to be registered under the {joint-iso-itu-t(2) country(16) us(840)} arc. The existing register of private organization names moved, in tact, from the {1 2 840} arc to the {2 16 840 1} arc, with ANSI serving as the registration authority. Therefore, two equivalent prefixes exist (in perpetuity) for currently registered organization names. Post-1991 registrations are made only under the {2 16 840 1} arc, and organizations with pre-1991 registrations are encouraged (but not required) to construct no new identifiers under the {1 2 840} arc. Various sources provide current OID descriptions, including www.oid-info.com/index.htm.

NOTE— The International Telecommunication Union Telecommunication Standardization Sector (ITU-T) is the part of ITU (an agency of the United Nations) that provides standards for telecommunication equipment and systems (formerly separate from CCITT). The {itu-t(0)} arc is also named ccitt to recall that CCITT was previously an organization independent from ITU-T. Similarly, the {joint-iso-itu-t(2) arc is also named joint-iso-ccitt.

Under the "iso" node, ISO has designated one subtree for use by other (inter)national organizations (org). Under that subtree, one of the U.S. National Institute of Standards and Technology nodes is assigned to the U.S. Department of Defense (dod). The initial development of the Internet was a Department of Defense project and, therefore, the Internet community was assigned a node in the dod subtree. The Internet Architecture Board (IAB) administers the "internet" node. The descriptive name "internet" is defined as:

internet   OBJECT IDENTIFIER ::= { iso org dod 1 }
(also known as 1.3.6.1)

Because of the ease of obtaining a node from IAB, NEMA requested and received a node that NEMA administers. This node is defined as:

nema   OBJECT IDENTIFIER ::= { iso org dod internet private enterprise 1206 }
(also known as 1.3.6.1.4.1.1206)

All data related to NTCIP Device Data Dictionaries or Protocols shall be defined under the NEMA branch of the tree. The organization of the naming tree down to the "nema" node is shown in Figure 2.

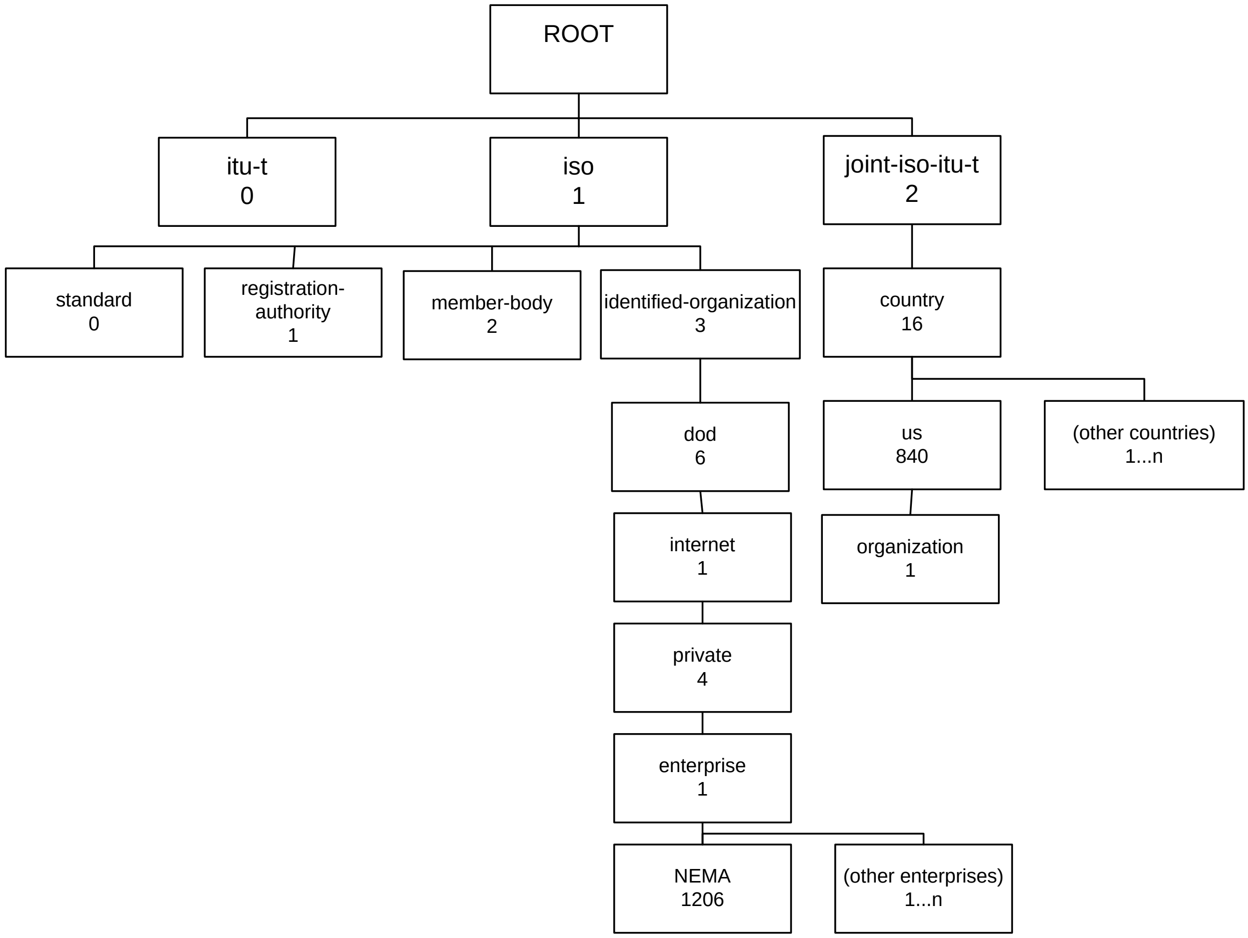

Figure 2 Portion of ISO Global Tree Showing Location of NEMA Node

2.2.2 NEMA Node

To organize information under the NEMA node, the subtree is further organized as shown in Figure 3.

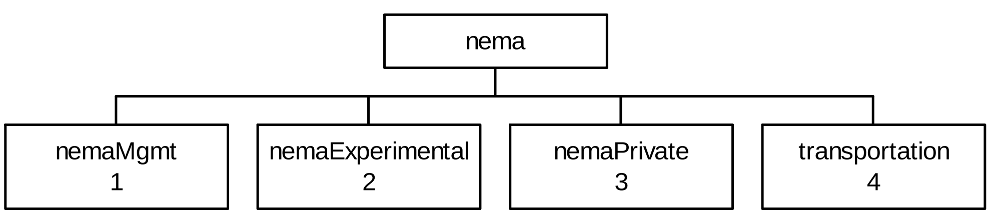

Figure 3 NEMA Node

This represents the organization under the nema node at the time NTCIP 8004 v02 was written. Current information can be found in NTCIP 7001.

2.2.2.1 NemaMgmt Node

The "nemaMgmt" node is used to register and identify objects that are defined in NTCIP standards that contain a MIB.

2.2.2.2 NemaExperimental Node

The "nemaExperimental" node is used to identify objects used only on an experimental basis. New MIBs, prior to being assigned a node number, can be put here. In the Internet community, multiple manufacturers use a MIB within the experimental area before it can move to a permanent location. As a part of the assignment process, NEMA may establish requirements as to how this subtree is used.
For example, an initial MIB proposed for objects related to loop-detectors could be defined and placed here for experimental usage. It might receive subnode number 17 and any objects would be identified with an OBJECT IDENTIFIER prefix of {nemaExperimental 17} or 1.3.6.1.4.1.1206.2.17.

2.2.2.3 NemaPrivate Node

The "nemaPrivate" node is used to identify objects defined unilaterally. Administration of the nemaPrivate subnode is by NEMA. NEMA assigns nodes to enterprises for the purpose of defining enterprise-specific MIBs. A request for a node assignment can be sent to the NTCIP Coordinator at ntcip@nema.org.

Upon receiving a node the enterprise may, for example, define new MIB objects under this node. In addition, it is strongly recommended that the enterprise also register its transportation devices under this subtree, to provide an unambiguous identification mechanism for use in management protocols. For example, if the "ABC, Inc." enterprise produced transportation devices, then it could request a node under the nemaPrivate node from NEMA. Such a node might be numbered:

1.3.6.1.4.1.1206.3.99

The "ABC, Inc." enterprise might then register their "Widget Controller" under the name of 1.3.6.1.4.1.1206.3.99.1, ensuring a unique identification. Thereafter, each enterprise is responsible for ensuring unique identification of information objects within their subtree. NEMA delegates the role of assigning numbers under each nemaPrivate node to those to which they are assigned, except of course for the initial enterprise number.

2.2.2.4 Transportation Node

The "transportation" node is used to register subnodes for protocol-related parameters, different classes of transportation equipment, and a subnode for the transit community. See Figure 4.

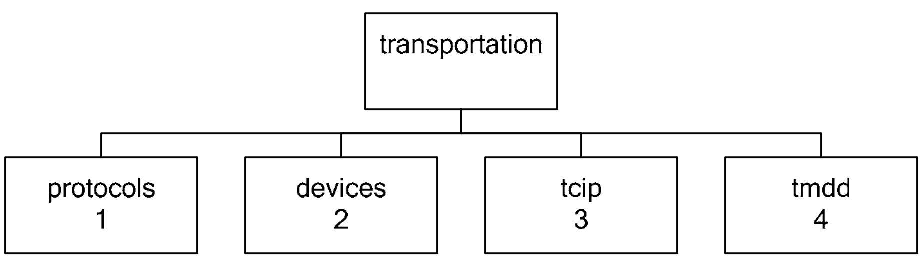

Figure 4 Transportation NODE

This represents the organization under the transportation node at the time NTCIP 8004 v02 was written. Current information can be found in NTCIP 7001.

2.2.2.4.1 Protocols Node

The “protocols” node shall be the beginning of a subtree that holds information related to specific protocols at various layers in a protocol stack, profiles that cover several layers, and one specific to Dynamic Object Management. The first four nodes in the subtree are assigned to:

a) Layers
b) Profiles

c) Dynamic Object Management
d) NTCIP Traps

Figure 5 represents the organization under the protocols node at the time that NTCIP 8004 v02 was written. Current information can be found in NTCIP 7001.

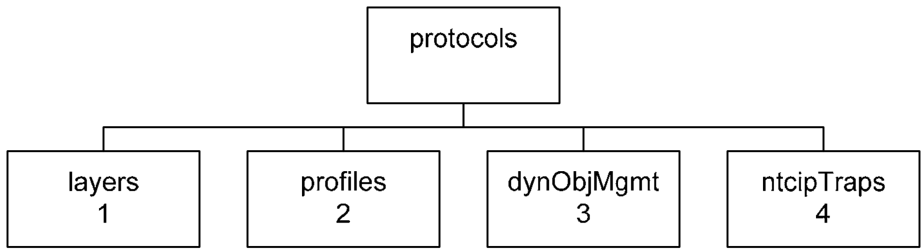

Figure 5 Protocols NODE

2.2.2.4.2 Devices Node

The devices node shall be the beginning of a subtree that holds information about various standard transportation device objects. See Figure 6. The first few nodes in the subtree are assigned to:

a) Actuated Signal Controllers
b) Ramp Meter Controllers
c) Dynamic Message Signs
d) Transportation System Sensors
e) Environmental Sensor Stations
f) Global

All approved standardized objects defined under each of these subtrees are standard objects specific to each class of transportation device.

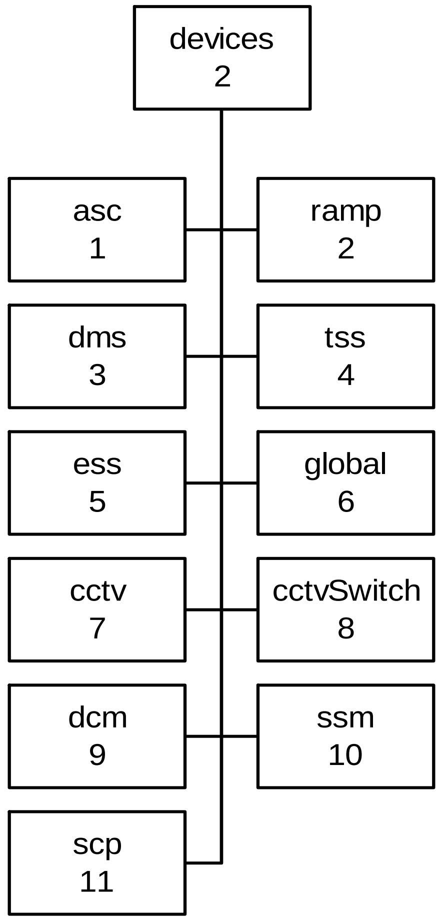

Figure 6 Devices NODE

This represents only part of the organization under the devices node. Current information can be found in NTCIP 7001.

2.2.2.4.3 TCIP Node

The “tcip” node has been assigned to the Transit Communications Interface Profiles Technical Working Group. Assignment of any nodes under this subtree is delegated to that group.

2.2.2.5 Other Nodes

NTCIP 7001 provides a listing of all node numbers that have been identified for use within the NTCIP suite of protocols. NTCIP 7001 is a living document and, as the need arises, new node numbers (and other types of identifiers) are added.

2.3 OBJECT SPECIFICATION

2.3.1 Base Object Specification

The base object specification shall be the OBJECT-TYPE Macro defined in IAB STD 16 (RFC 1212), but with the following restrictions and additions:

a) The ACCESS field shall never be assigned a value of "write-only".
b) The DESCRIPTION field shall be present.
c) If specified, the DEFVAL field value shall be used as the default value.

The OBJECT-TYPE Macro requires that each object definition consist of the following fields:

a) Object Name
b) SYNTAX
c) ACCESS
d) STATUS
e) DESCRIPTION
f) INDEX
g) REFERENCE
h) DEFVAL
i) Object Identifier

The INDEX field is only used in table entry constructs. The REFERENCE and DEFVAL fields are optional.

NOTE—Sections 2.3.1.1 through 2.3.1.9 are provided to put preceding and subsequent sections in context.

2.3.1.1 Base Object Name

This field shall consist of a textual name, termed the OBJECT DESCRIPTOR, for the object type being defined. An OBJECT DESCRIPTOR is an identifier consisting of one or more letters, digits, or hyphens. The initial character shall be a lower-case letter, and the final character shall not be a hyphen. Further, a hyphen shall not be immediately followed by another hyphen. The descriptor shall be unique and mnemonic, and shall not exceed 64 characters in length. The use of hyphen only applies to module names. Within the MIB, the hyphen is not allowed as a character in any descriptor.

This OBJECT DESCRIPTOR is equated with the value of the specified OBJECT IDENTIFIER field.

2.3.1.2 Base Object SYNTAX

IAB STD 16 (RFC 1155, Section 6) states that an object's SYNTAX fields shall resolve to an instance of one of the following primitive ASN.1 types:

a) SEQUENCE
b) INTEGER
c) OCTET STRING
d) OBJECT IDENTIFIER
e) NULL
f) Counter
g) Gauge
h) TimeTicks
i) Opaque
j) IpAddress

The term "resolve" is used because specification writers may use a textual convention or alias to create terminology that is more meaningful, This list of syntax types does not include all primitive ASN.1 types defined in ISO/IEC 8824.

Any object with a SYNTAX of OBJECT IDENTIFIER can be defined such that object represents an invalid/neutral state by setting its value to "null".

Unless specific normative text to the contrary is stated, any object that resolves to a SYNTAX of OCTET STRING (OPAQUE, for example) shall not be padded. For example, the string "test" shall be encoded as 0x74 65 73 74 and not 0x00 74 65 73 74 or 0x74 65 73 74 00.

2.3.1.3 Macro ACCESS

As defined in IAB STD 16, the ACCESS field is to be assigned one of the following values:

a) read-write
b) read-only
c) not-accessible
d) write-only

The values of read-write and read-only are straightforward. The value of not-accessible is used to define structural definitions primarily related to tables. The write-only access does not have a practical application.

For use within NTCIP, write-only access shall not be used.

2.3.1.4 Macro STATUS

As defined in IAB STD 16 (SMIv1) or IAB STD 58 (SMIv2), the STATUS field can take on one of the following values:

a) Mandatory—In SMIv1, this value indicates the definition is valid and implementation is required for conformance. In SMIv2, this value is not in use.
b) Optional—In SMIv1, this value indicates the definition is valid; however, implementation is not required for conformance. In SMIv2, this value is not in use.
c) Deprecated—The value is valid in specified implementations, for example to maintain backward compatibility, but the object may have been replaced.

   NOTE—To maintain backward compatibility for legacy implementations, objects with a STATUS value of “deprecated” may require support. When necessary to support legacy implementations, required support for objects with a STATUS value of “deprecated” is indicated using the Protocol Requirements List (PRL).

d) Obsolete—The definition is no longer valid, was found to be flawed, was redundant, or was not useful.

A status of mandatory is used to mean that if a group of objects expresses the functionality implemented in an entity, then the specific mandatory objects within that group are to be implemented. A status of optional is used to mean that this data element object may be optionally implemented.

Whether an object is required should be defined in a conformance statement, PICS, or PRL. Developers or specifiers of such statements should call for implementation if the functionality modeled by the definition is present or desired in the system.

2.3.1.5 Macro DESCRIPTION

The DESCRIPTION field is an ASCII-NVT string, which defines the semantics of the data element object type. The object description shall provide an unambiguous definition of what the object does. The description field shall be present in all OBJECT-TYPE definitions, and shall contain a meaningful definition of the purpose of the object.

ASCII-NVT is a 7-bit code representing ASCII characters 0 through 127. In ASCII-NVT, the following control characters have significance and support is required:

a) NULL (NUL)
b) Line Feed (LF)
c) Carriage Return (CR)

Other control characters are defined but, as stated in RFC 854 (IAB STD 8), support is not required. This includes the Tab (TAB) character. Its use is discouraged because specific character spacing cannot be guaranteed. Except for the CR and LF control characters, the ASCII characters 0 through 31 and 127 are interpreted as NUL.

According to IAB STD 16, the DESCRIPTION field need not be present. For NTCIP, the DESCRIPTION field shall always be present and used. Other information that shall be included in this field is defined in Section 2.4.

2.3.1.6 Macro INDEX

The INDEX field, which may be present only if that object type corresponds to a TableEntry, defines instance identification information for that object type.

2.3.1.7 Macro REFERENCE

The REFERENCE field is a textual cross-reference to a standard or document and specific section that provides information about specific range or values that quantify the object of interest.

2.3.1.8 Macro DEFVAL

As defined in IAB STD 16, the DEFVAL field is optional and is used to define an acceptable default value, which may be used when an object instance is created.

For NTCIP, if the DEFVAL field is specified, the value specified shall be used as the default value.

2.3.1.9 Macro Object Identifier

The Object Identifier shall consist of the subtree identifier's descriptive name of the node and the position of the object under that name.

2.3.2 Simple Object Specification

A simple object is an object type representation of a type of data element. The term simple object is used to differentiate it from those defined with the standard OBJECT-TYPE Macro, those defined with the block object specification, and those defined with the dynamic object specification.

The simple object specification shall be the base object specification as defined in Section 2.3.1, but with the following additions or modifications:

a) The DESCRIPTION field shall include the Data Dictionary meta attributes for:
    1) <Definition>—delimiting original OBJECT-TYPE Macro DESCRIPTION Field
    2) <Object Identifier>—full identifier from the root node
b) As appropriate, the DESCRIPTION field may include the Data Dictionary meta attribute for:
    1) <TableType>
    2) <Unit>
    3) <Format> (Bitmap)
c) The REFERENCE field shall provide a textual cross-reference to a standard or other document defining the Reference Value Domain

2.3.2.1 Simple Object DESCRIPTION

In addition to the base object specification requirements for the DESCRIPTION field, the meta attributes subfields <Definition> and <Object Identifier> shall be used and defined.

a) The original DESCRIPTION is delimited by "<Definition>".
b) The <Object Identifier> subfield shall be included and is the full Object Identifier from the root node.
c) The document editor of an NTCIP standard shall derive the Object Identifier by compiling the MIB. All object definitions defined in that NTCIP standard shall be contained in the MIB that is compiled, and use the resulting output file. The MIB compilation output file contains both the object name and the

full Object Identifier. The Object Identifier for each object shall be copied from that output file and be pasted into the Description field of the corresponding object definition. The Object Identifier subfield shall not include an instance identifier.

d) The optional <TableType> subfield shall be included if the base object specification defines a table construct.
e) The optional <Unit> subfield shall be included if the SYNTAX defines an object specification where a unit of measure is warranted to clearly define the syntax.
f) The optional Bitmap <Format> subfield shall be included if the base object specification defines a bitmapped object wherein each bit of the object has a specific meaning.

2.3.2.2 Simple Object REFERENCE

The REFERENCE field shall be a textual cross-reference to a standard or document and specific section that provides information about specific range or values that quantify the object of interest.

For example, NEMA TS 2-2003 Section 3.5.3.1 defines the values for the Yellow Change as 3.0 to 25.5 seconds.

2.3.3 Block Object Specification

A block object shall be defined as an object type representation of a data-frame. It is a merging of IAB STD 16 and meta attributes defined by NTCIP 8004 v02. The term block object refers to an object that has a syntax of OER Encoded String. Unlike the opaque syntax defined in IAB STD 16, the syntax shall be an "external" textual description of the objects to be encapsulated in the block. The external portion shall take the form of a non-compileable SEQUENCE or SEQUENCE OF construct.

The block object specification shall be the base object specification as defined in Section 2.3.1 but with these additions or modifications:

a) The Object Name shall end with the term "Block".
b) The SYNTAX shall be equated to an "OerString".
c) The ACCESS shall be either read-only or read-write.
d) The DESCRIPTION field shall include the Data Dictionary meta attributes for:
    1) <Definition>—delimiting original OBJECT-TYPE Macro DESCRIPTION Field
    2) <Object Identifier>—from the root node
e) An External Construct that shall list the Object Descriptors of the objects to be encoded in the OerString.

2.3.3.1 Block Object Name

The block name shall follow the same form as defined in Section 2.3.1.1 but shall end with the suffix "Block".

2.3.3.2 Block Object Syntax

The block SYNTAX shall be a textual convention that resolves to an "OerString" type. The "OerString" type is defined in Transportation Management Information Base (TMIB-III) as an unconstrained OCTET STRING. The actual size of the string shall be defined by the OER encoded value of the external construct.

The "OerString" type shall be imported into any module that uses it.

2.3.3.3 Block Access

The ACCESS value of block objects shall be restricted to one of the following:

a) read-only
b) read-write

Any block object containing a read-only object in the external construct shall have an ACCESS of read-only. For a set operation on a block object to succeed, all the objects with the block object shall have an access of read-write.

2.3.3.4 Block Description

The Block Description shall follow the same form as defined in Section 2.3.1.5. Block objects may encapsulate columnar objects and, therefore, all index values used in the identification of an object shall be clearly defined.

2.3.3.5 Block External Construct

An NTCIP message may need to include objects that are specific instances of objects and objects that are defined in multiple MIBs. Since this is outside the scope of a MIB, an external structure shall be used to define the objects contained in a message. This permits a variation that allows a clear and unambiguous reference to the objects defined in the MIB structure. Any Type or TypeReference field may be replaced with an NTCIPObjectReference defined as:

NTCIPObjectReference ::= NTCIPObjectName"."IndexSuffixes ExternalReference

The NTCIPObjectName is a name of an accessible object within a MIB.

The IndexSuffixes shall be as follows:

IndexSuffixes ::= IndexSuffix | IndexSuffixes "." IndexSuffix
IndexSuffix ::= DefinedValue

Thus, the IndexSuffixes field is a list of the associated indexes as required to explicitly identify the exact instance of a given object-type. Each index may be identified as either a specific value or a variable; if a variable is used, the variable shall be defined within the same module.

A leaf object shall always include a ".0" as a suffix.

The ExternalReference is included when any object in the block object is not defined in the current module. The ExternalReference is an ASN.1 comment indicating what module defines that object. The ExternalReference takes the form:

ExternalReference ::="-- @ " <module descriptor>

where <module descriptor> is the name of the external MIB Module where the subordinate object is defined. All block object definitions shall be mapped in a valid ASN.1 module structure per the rules above. Annex B provides several examples.

2.3.4 Dynamic Object Specification

A dynamic object shall be defined as an object type representation of a data-frame. A dynamic object is an implied, simple sequence of specific NTCIP objects, similar to a block object, but the component objects within a dynamic object are defined at run-time by the management station rather than being defined in a static definition.

The objects used in a DynObjectEntry shall be defined using the base object specification defined in Section 2.3.1, but with the following additions or modifications:

The DESCRIPTION field shall include the Data Dictionary meta attributes for:

a) <Definition>—delimiting original OBJECT-TYPE Macro DESCRIPTION Field
b) <Object Identifier>—from the root node

2.4 META ATTRIBUTES

To support additional understanding of objects, various meta attribute subfields are added to the DESCRIPTION Field of various object specifications. These include:

a) Definition
b) Object Identifier
c) Table Type
d) Unit of Measure
e) Format (Bitmap)

In addition to the description of the MIB that is defined in a standard, a compileable, text-only version of the MIB is also produced. The procedures for creating this file are defined in NTCIP 8005 v01. NTCIP 8005 v01 defines a set of "meta attributes" that appear in that version of the MIB. These meta attributes are the file header that is expressed in the form of comments. Technically, the header consists of the following "NTCIP-Defined" meta attributes that are added to each MIB Module:

a) Filename
b) Source
c) Description
d) MIB Revision History
e) Copyright Statement
f) MIB Distribution Notice

NOTE—Some of the DESCRIPTION subfields and elements of the Object Type Macro are also used to fill out meta attributes. For example, an entry for an ASN.1 Name is automatically derived from OBJECT DESCRIPTOR (see Section 2.3.1.1) as part of the process in entering the information in a data registry. NTCIP 8005 v01 contains further information and how some other meta attributes are derived.

2.4.1 Definition

To delineate the original intended content of the DESCRIPTION field of the OBJECT-TYPE Macro as defined in IAB STD 16 (RFC 1212), the text of that field shall be preceded by the delimiter <Definition> to separate it from other subfields that may now appear in the DESCRIPTION field.

2.4.2 Table Type

The DESCRIPTION field of an object specification that defines a table shall include a Table Type subfield. It is delimited by <TableType> and the value shall be either "static" or "dynamic". The term static shall apply to tables where all entries exist irrespective of whether they have been initialized or not. The term dynamic shall apply to tables where an entry does not exist until it is created by a management application.

2.4.3 Unit of Measure

The value of the Unit of Measure subfield is delimited by <Unit>. The units typically refer to a measure of time, distance, weight, or volume. General use values shall be defined in accordance with ISO 1000, which recommends selected decimal multiples and submultiples of units, such as seconds, deciseconds, and milliseconds. If ISO 1000 does not provide for a definition of <Unit>, then the <Unit> field can be left blank; however, the statement "<Unit>" shall be shown.

2.4.4 Bitmap Format

The content of the Bitmap Format subfield is delimited by <Format> and shall be defined according to the following:

a) If a Bitmap format is contained in a SYNTAX of INTEGER, then the following format shall be used when defining the DESCRIPTION field within an object:

Bit Number	Name		Description
Bit <x>:	<bitName>	=	<description>
	…		
Bit 1:	<bitName>	=	<description>
Bit 0:	<bitName>	=	<description> (Least Significant Bit of the Least Significant Byte)

b) If a SYNTAX of BITMAPx (i.e., BITMAP8, BITMAP16, BITMAP32, BITMAP64) is used for an object definition, than each bit should be explicitly defined as defined for a SYNTAX of INTEGER preceding.

c) If a Bitmap format is contained in a SYNTAX of OCTET STRING, then the definition within the DESCRIPTION field of an object shall explicitly state what each bit means and particularly where the first bit is and how to proceed within a matrix (first to right/left or first up/down).

Figure 7 shows the position of Most Significant Bit (MSB) versus Least Significant Bit (LSB) within each byte and the position of Leading Octet versus Trailing Octet. Figure 7 also shows which bits within an octet need to be padded, if fewer bits than the bits within a full octet are used.

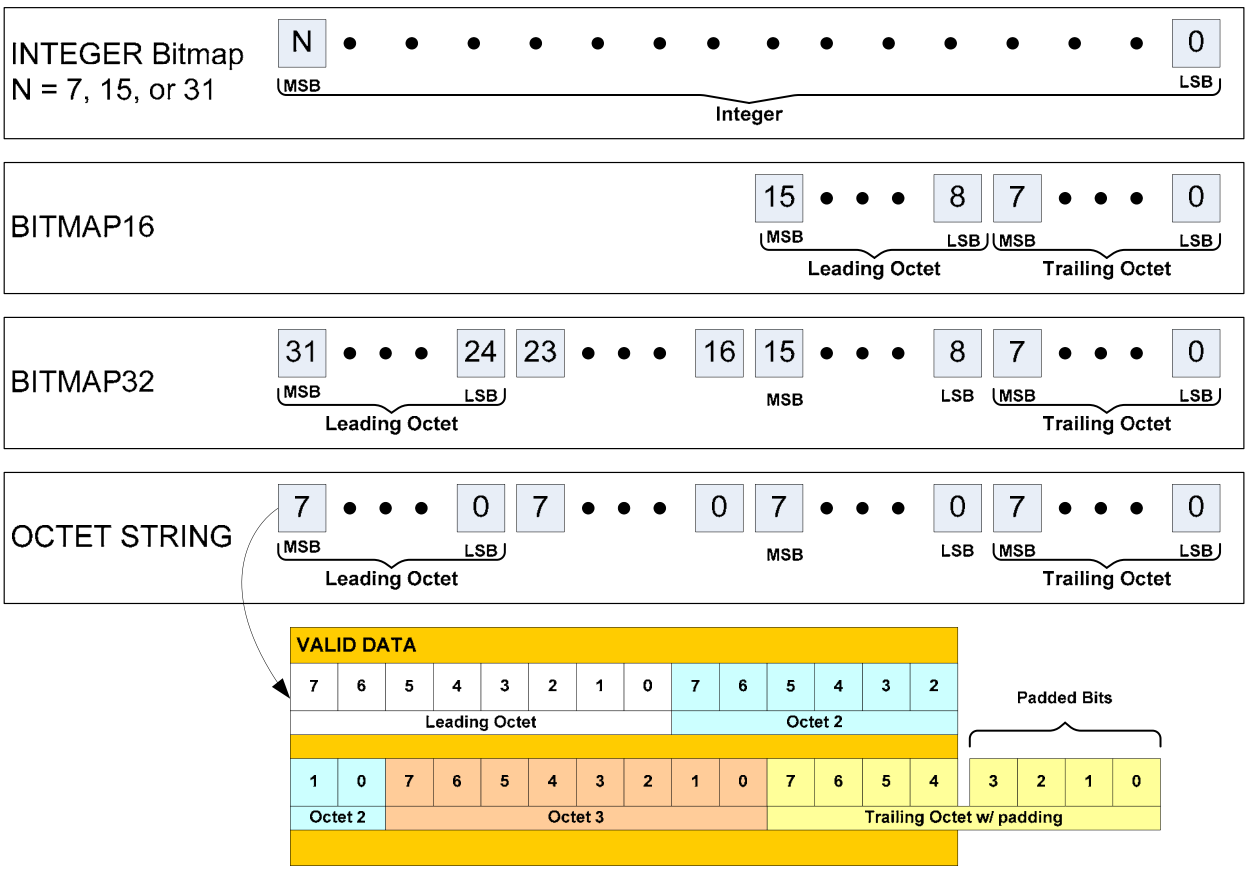

Figure 7 Most Significant Bit (MSB) versus Least Significant Bit (LSB)

2.4.5 MIB Module Meta Attributes

The definition of the "NTCIP-Defined" MIB Module meta attributes can be found in NTCIP 8005 v01. These meta attributes are the same for all of the objects in an NTCIP 1200 Series Document MIB and, therefore, are defined on a MIB Module basis. Furthermore, they may not necessarily be defined by the NTCIP Working Group but by the NTCIP Data Steward when the MIB Module is created or when the MIB Module is checked for proper syntax and construction. The exact definition of each MIB Module meta attribute can be found in the NTCIP standard and are only referenced here to show completeness.

2.4.6 Description Field Meta Attribute Encapsulation

Meta attribute subfields are added to the DESCRIPTION of various object definitions using the following format. In IAB 16 (RFC 1212), the description field is defined as:

DescrPart ::= "DESCRIPTION" value (description DisplayString) **|** empty

This is modified to include one or more of the meta attributes as shown:

DescrPart ::= "DESCRIPTION"	"<Definition>"	value (description DisplayString)
	"<Object Identifier>"	full Object Identifier value from the root node
	"<Unit >"	value (unit of measure DisplayString)

The meta attribute subfields shall appear after the last character of the "description" but before the closing quote. They shall start on a new line and appear on their own separate lines. The closing quote of the DESCRIPTION shall appear immediately after the last encapsulated meta attribute. See Annex A for examples.

2.5 MANAGEMENT INFORMATION BASE (MIB)

A Management Information Base (MIB) is a structured collection of objects. The objects represent the individual variables, parameters, and status, and other types of information that may be exchanged between two devices. Objects that are related to each other are defined in a MIB Module. All the groups (or MIB Modules) that represent the total information that may be exchanged between two systems are referred to as a MIB.

2.5.1 Logic Behind the Documentation Format

By defining information according to the rules defined in IAB STD 16 and those defined in ISO/IEC 8824, there is an unambiguous and coordinated method for sharing that information. Independent groups can be simultaneously developing object sets (data dictionaries) for different functional areas of a device. The rules and procedures also permit an organization or vendor to extend the basic set of standard information to cover implementation- or application-specific requirements.

Entire MIBs or individual Object Definitions can be combined and used for development of an integrated approach to the management of information. NTCIP Object Definitions, Internet MIBs, and Vendor-specific MIBs can be imported into a program that treats them as a single entity. This concept is illustrated in Figure 8. These modules also serve as the foundation for defining file structures. A program can interpret the data in a file by parsing it according to the description in the MIBs.

The result of creating and identifying managed objects using ASN.1 is a computer readable description of the information. These modules are ASCII-NVT text, not binary. Management applications running on central control hosts can read these modules to gain information about the capabilities of remote devices. Many SNMP management applications can dynamically load and unload modules (MIBs) describing the information within remote networking devices.

Manufacturers supplying transportation equipment should make available manufacturer-specific MIB modules for the equipment. All manufacturers need to obtain a vendor number from either NEMA (see Section 2.2.2.3) or IANA prior to creating MIB modules that have manufacturer-specific data.

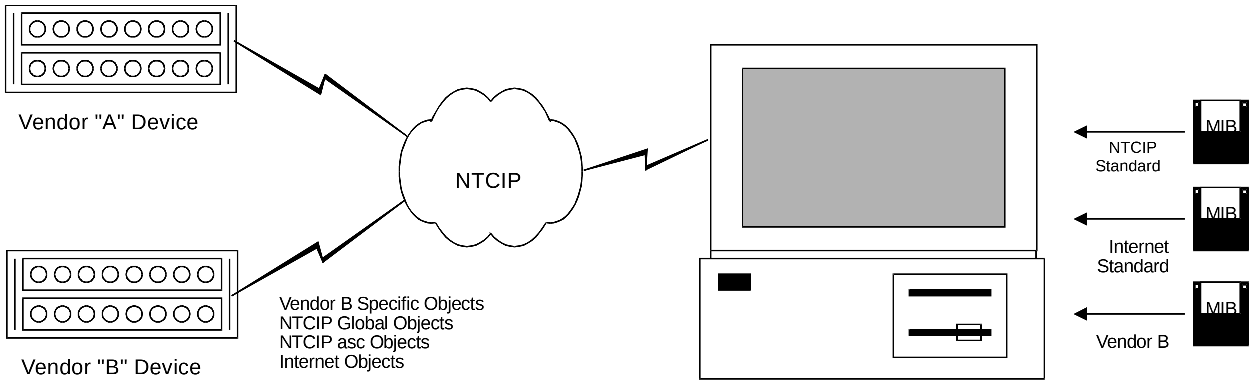

Figure 8 MIB Integration

2.5.2 Extensions to the MIB

Every new NTCIP Standard MIB version supersedes all previous versions. New versions shall:

a) identify object STATUS as “deprecated,” when appropriate, but shall not delete the object names;
b) identify object STATUS as “obsolete” for those objects with a prior STATUS value of “deprecated” (if necessary), but shall not delete their names;
c) augment the definition of an object type corresponding to a list by appending non-aggregate object types to the object types in the list, as appropriate;
d) define entirely new objects, as appropriate.

New versions shall not change the semantics of any previously defined object without re-defining that object, i.e., changing the name of the object and assigning a new OBJECT IDENTIFIER.

A new object definition that deprecates a previously defined object should note that fact in the new object’s DESCRIPTION Field.

2.6 MULTI-VERSION INTEROPERABILITY (MVI, BACKWARD COMPATIBILITY)

With the development and publication of newer versions of various NTCIP standards, the need to address and ensure multi-version interoperability (MVI, often referred to as “backward compatibility” with a previous version, or interoperability to a defined extent with future versions) arises. Provisions in Section 2.6 are designed to permit MVI.

2.6.1 Rules For Existing Objects

The STATUS of an object may be changed to “deprecated,” “obsolete,” or “mandatory.” Often, a new object is created to replace one with a STATUS value of “deprecated.” Examples of reasons to change the STATUS value of an object to “deprecated” include:

a) The associated functionality is no longer relevant. For example, the object may be redundant, or might not have been implemented; or
b) The current design of the object is flawed or ambiguous in design, and cannot be corrected within the existing object given constraints on making changes to existing objects.

Provisions for changing the status of an object include:

a) Any new NTCIP object shall be assigned a textual name and OID that is unique among all NTCIP standards, even if it is replacing the capabilities of a deprecated object.
b) Deprecation of a particular instance (value) of an enumerated object is allowed. For example:

```
INTEGER { oldValue(0) -- deprecated, the value zero created problems
```


c) The DESCRIPTION of an object shall not be changed or added to except to make a minor editorial correction. It is acceptable to add an informative statement within the DESCRIPTION, by inserting the key word <Informative> prior to such statement. For example, an informative addition to the DESCRIPTION of a deprecated object might reference the replacement object, potential issues related thereto, and the reason for deprecation.
d) The standard shall make appropriate provisions in the PICS or PRL that may be used to require support of objects with a STATUS value of “deprecated” when needed to permit MVI.
e) Adding new enumerations is permissible.
f) Adding new columns to a table is permissible.
g) An object textual name can be changed to adhere to rule b) in Section 2.6.1 and rule a) in Section 2.6.2.
h) The meta attributes within the DESCRIPTION field of an object can be changed according to Section 2.4, except if the change creates an interoperability problem for center-to-field interfaces.
i) Extending the range of an Integer object is permissible.

2.6.2 Rules Allowed for New and Existing Objects

a) Strictly follow "Table" and "Entry" naming conventions.
b) Do not allow unused IMPORTS and textual conventions.
c) Do not allow DEFVAL on counters. An initial value can be specified in the DESCRIPTION if needed, or in an informative addition to the DESCRIPTION for an existing object.
d) Do not allow bad modeling practices in table design, which prohibits:

    1) Specifying the same item more than once in the same index
    2) Specifying an object that is not under the entry object in the OID tree
    3) Using the indices from another table in a different order than specified in the other table
    4) Using only a subset of the indices from another table
    5) Specifying additional index items before all the index items from another table
    6) Specifying the index items from more than one table

e) Strictly follow Access rules, which prohibit:

    1) Using write-only
    2) Using not-accessible except in table entries and table indexes

f) Require all OCTET STRINGs to have a size, and all INTEGERs to have a range. This requires standards developers to consider the impacts of unconstrained data types. The data type Counter has an implicit range of 0..4294967295 per RFC 1155. If a smaller counter is desired, use an integer range and describe the desired behavior. Also, the data type INTEGER has a default range of 1..2147483647.

2.6.3 Rules Application and Exceptions

Rules in Section 2.6.1 apply to existing objects only. Rules in Section 2.6.2 apply to both new and existing objects. The intent is that existing objects be modified one time only to bring them in compliance with Section 2.6.2. Any change made to an existing object per rules in Section 2.6.2 shall satisfy rules in 2.6.1. In some cases, an existing object may need to be deprecated in favor of a new object to achieve this.

Any exceptions to rules in Section 2.6.1 and Section 2.6.2, or their application per Section 2.6.3, shall be explained to and approved by the Joint Committee on the NTCIP.

Section 3
NTCIP PROCEDURES

3.1 NTCIP ELEMENTS OF PROCEDURES

IAB STD 16 does not elaborate on a number of functions related to different types of table constructs. The definition of traps was not introduced until later and does not deal with the definition of enterprise-specific traps. Section 3 defines additional requirements that shall be followed when defining tables.

3.2 TABLES, ROWS, AND BLOCK OBJECTS

SNMP tables are special types of SNMP objects that allow sequences of information to be supported. Tables are distinguished from scalar objects in that tables relate all items in a row to an index for that row. The developers of IAB STD 16 wanted a method to construct imaginary, tabular structures as part of the collection of objects that constitute the MIB. Each such conceptual table contains zero or more rows, and each row may contain one or more objects, termed columnar objects. This conceptualization is implemented by using the OBJECT-TYPE Macro to define both an object that corresponds to a table and an object that corresponds to a row in that table. However, at the protocol level, relationships among columnar objects in the same row are a matter of convention, not of protocol. The convention used within SNMP is that a table or row of a table is not-accessible and operations only apply to scalar objects and cells within a table.

The SNMP limitation of operating only on individual object instances can impose a severe penalty in cases where communications bandwidth has to be considered. To address this issue, NTCIP has added the concept of block objects. From the standpoint of SNMP or any similar protocol, a block object is treated as a single object.

When the number of rows in a table is defined, the minimum value of a maxObjects definition shall not be specified as 0. For example, NTCIP 1201:2005 defines the following:

```
globalMaxModules   OBJECT-TYPE
    SYNTAX         INTEGER (1..255)
    ACCESS         read-only
    STATUS         mandatory
    DESCRIPTION
            "<Definition>The number of rows that are listed in the globalModuleTable.
            <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.2
            <Unit>module"
::= { globalConfiguration 2}
```

The use of SYNTAX INTEGER (0..255) would be incorrect. Support of the table is indicated by a conformance statement and not by specifying that the number of rows is 0. When a table is not supported, accessing any associated maxObjects shall return a "noSuchName" Error.

3.3 TABLE OPERATIONS

Unless specifically prohibited by normative text added, extensions to the standard tables are permitted.

The values associated with the columnar objects in a table can lead to ambiguities unless specific steps are taken. This is especially true for dynamic tables. The row status definitions in Section 3.3.1 should be used to indicate that the objects in the row, when taken as a whole, satisfy any relationships (consistency checks) and/or can be used by the end application.

3.3.1 Row Status in Static Tables

In a static table, all rows exist irrespective of whether the columnar objects contain appropriate values. The value of a columnar object within a row may be inappropriate when the values of other columnar objects in the row are considered. An entire row itself may also be considered inappropriate under some circumstances. If this is the case, a static table shall include an additional columnar object that defines row status and has the SYNTAX of RowStatusStatic.

Where
```
RowStatusStatic  ::= INTEGER
                   { other (1),
                     standby (2),
                     available (3),
                     invalid (4),
                     activate (5),
                     deactivate (6) }
```

RowStatusStatic has four states and two commands associated with it. The four possible states are defined as:

a) Other—Status is actually controlled or defined by a user- or manufacturer-specific object.
b) Standby—All the columnar data in the row have passed any consistency but are not to be used by the end application.
c) Available—All the columnar data in the row have passed any consistency and are to be used by the end application.
d) Invalid—One or more columnar objects has a value that caused the row to fail the consistency check.

Setting RowStatusStatic equal to one of these values shall return a badValue Error.

The only two possible command values that may be set by a management application are:

a) Activate—Make the columnar data in the row available for use by the end application.
b) Deactivate—Make the columnar data in the row unavailable for use by the end application.

The UML state transition diagrams for RowStatusStatic are defined as indicated in Figure 9, Figure 10, or Figure 11.

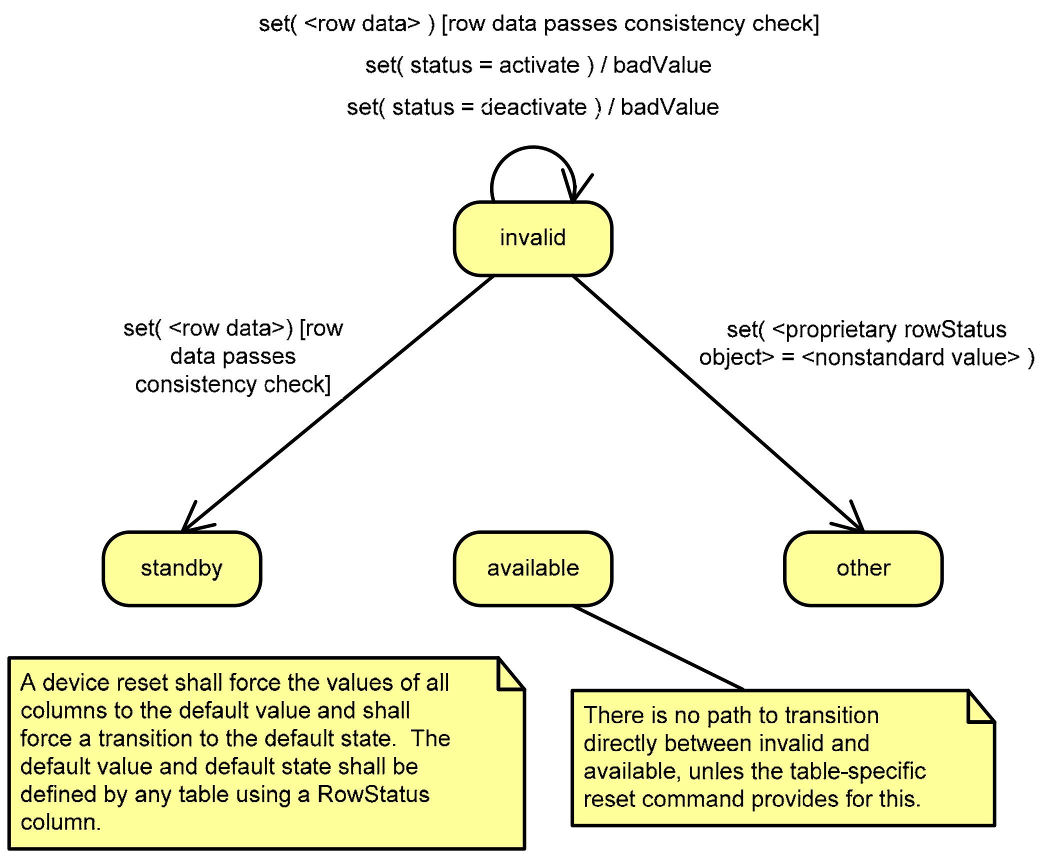

Figure 9 Row Status Static—Invalid

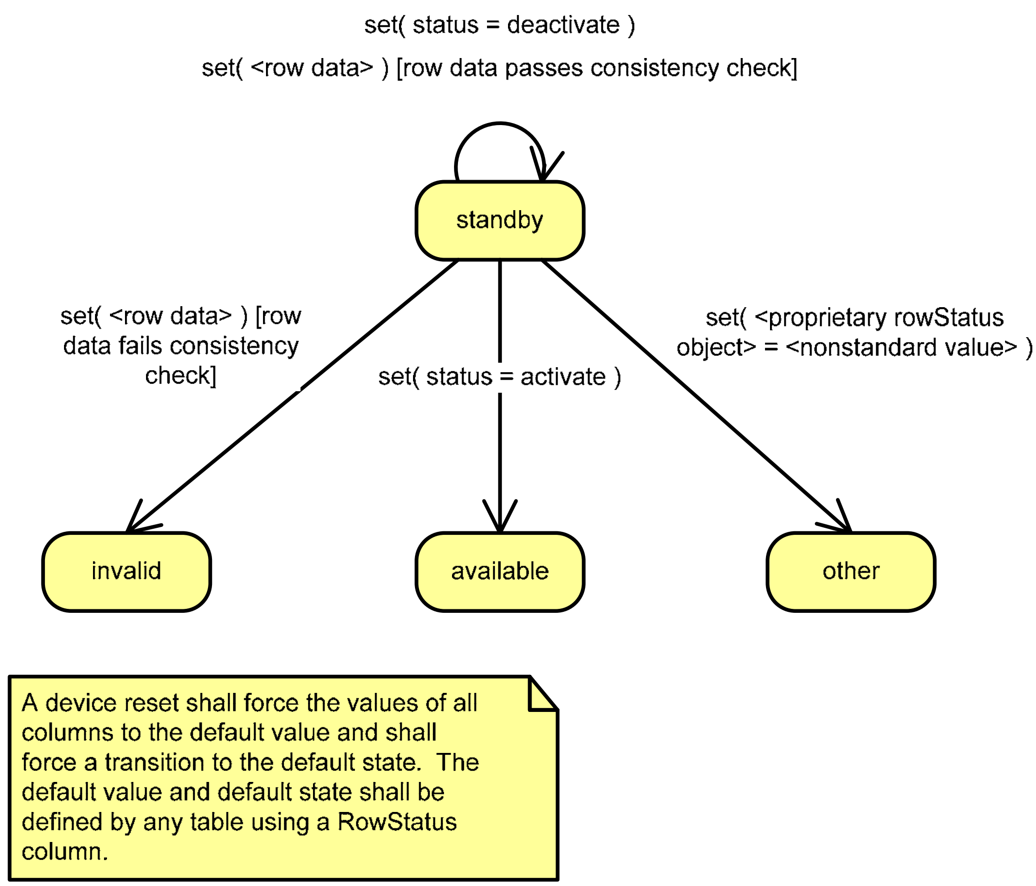

Figure 10 Row Status Static—Standby

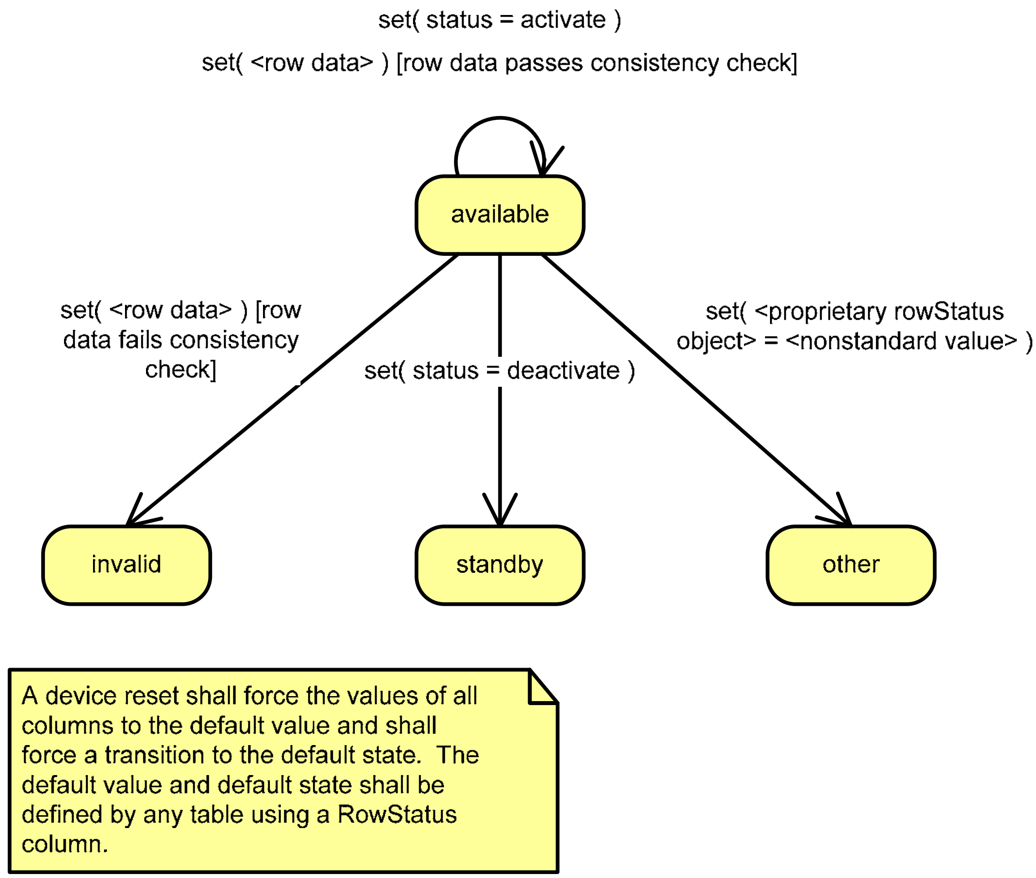

Figure 11 Row Status Static—Available

3.4 ENUMERATIONS DEFINED AS OTHER

All values of NTCIP objects that are not defined by the standards are reserved for committee use. A management application may not set an enumerated type to a value declared as “other”. An attempt to write a value of “other’ shall return a badValue error to a management application.

Unless normative text is added to specifically prohibit the use of the "other" state, a user- or manufacturer-specific object shall be permitted to define an object specification that extends the possible states, For example, NTCIP 1202:2005 includes the following object:

```
coordCorrectionMode   OBJECT-TYPE
   SYNTAX   INTEGER { other (1),
                      dwell (2),
                      shortway (3),
                      addOnly (4) }
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
      "<Definition> This object defines the Coord Correction
       Mode. The possible modes are:
         other: the coordinator establishes a new offset by
            a mechanism not defined in this standard.
```

```
         dwell: when changing offset, the coordinator shall
            establish a new offset by dwelling in the coord
            phase(s) until the desired offset is reached.
         shortway (Smooth): when changing offset, the
            coordinator shall establish a new offset by
            adding or subtracting to/from the timings in a
            manner that limits the cycle change. This
            operation is performed in a device specific
            manner.
         addOnly: when changing offset, the coordinator
            shall establish a new offset by adding to the
            timings in a manner that limits the cycle
            change. This operation is performed in a device
            specific manner.
       ...
```

To define a new correction mode, something like the following proprietary object could be used:

```
coordCorrectionModeExt   OBJECT-TYPE
   SYNTAX   INTEGER { other (1),
                      subOnly (2),
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
      "<Definition> This object defines an extension to the Coord Correction
       Mode as defined in NTCIP 1202. The possible modes are:
         other: the coordinator establishes a new offset by
            according to NTCIP-1202::coordCorrectionMode. Upon setting of
            this object to other, NTCIP-1202::coordCorrectionMode is
            automatically set to shortway.
         subOnly: when changing offset, the coordinator shall
            establish a new offset by subtracting from the timings in such a
            manner that limits the cycle change. This operation is performed
            in a device specific manner
         ...
```

In this case, setting coordCorrectionModeExt equal to "subOnly" forces coordCorrectionMode equal to "other". Setting coordCorrectionModeExt equal to "other" forces coordCorrectionMode equal to "shortway".

3.5 RESERVED BITS

Any object that has reserved bits shall return a bad value if a management application writes a one to them.

3.6 MULTIPLE MANAGEMENT ACCESS

NTCIP 8004 v02 does not preclude the potential for two or more management applications from simultaneously accessing the same object. The objects in the Global Database Management and Security groups do not prevent multiple management applications from accessing the data simultaneously. It is the responsibility of any agencies involved in inter-jurisdictional control to define procedures to ensure against this situation.

3.7 BITMAPPED OBJECTS

Objects that define an entity that gives specific meaning to the individual bits of the entity should use one of the BITMAP textual conventions (BITMAP8, BITMAP16, or BITMAP32) as its SYNTAX and include the Bitmap <Format> subfield in the DESCRIPTION field.

Annex A
NTCIP STRUCTURE OF MANAGEMENT INFORMATION MIB (MIB)
[Normative]

Annex A defines the overall structure of the NTCIP-defined management information and several textual conventions that are believed to be useful for a broad range of applications. The text provided from Annex A.1 through Annex A.3 (except the headings) constitutes the standard NTCIP 8004 v02 MIB.

Text preceded by a double hyphen in the MIB definitions represents normative text for NTCIP 8004 v02.

A.1 SMI HEADER

```
--
*****************************************************************************
-- Filename:    8004v0217MIB
-- Source:      NTCIP 8004 v02.17
-- Description: This MIB defines the overall structure of the
--              NTCIP-defined management information and several textual
--              conventions that are believed to be useful for a broad range
--              of applications
-- MIB Revision History:
-- 10/01/96     NEMA TS 3.2 approved
-- 01/01/98     Preliminary Release of TS 3.2 NEMA_SMI MIB formatted for
--              80 columns and no tabs
-- 07/08/98     Added Copyright Notice
-- 10/08/98     Amendment 1 approved
-- 03/09/00     Changed filename and updated copyright years
--              Updated the MIB to Amendment
-- 08/09/00     Modified header format and wording of copyright and MIB
--              Distribution Notice
-- 11/16/01     Incorporated into NTCIP 1103, defined tree down to and
--              including device nodes, and changed filename
-- 9/19/02      Incorporated into NTCIP 8004 and changed filename and module
--              name
-- 09/27/04     Updated header information and END remark
-- 01/26/05     Added chap and modem to list of Object Identifier nodes.
-- 02/12/05     Removed embedded comment indicator between EXPORTS and
--              EVERYTHING
-- 11/29/07     Added tmdd and ntcipTraps nodes
-- 03/16/10     Edited Jointly Approved NTCIP 8004 v02 for publication.
--
--Copyright 2010 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 in whole or in part in any form, translation
--into other languages and display are reserved by the copyright owners 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 for the MIB, Do not copy without written
--permission of either AASHTO,ITE, or NEMA.
--
```

```
--                      Joint NEMA, AASHTO, and ITE
--                   NTCIP Management Information Base
--                         DISTRIBUTION NOTICE
--
--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 sub-range 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.
--
--NTCIP is a trademark of AASHTO/ITE/NEMA.
--**************************************************************************
```

```
NTCIP8004v02
--DEFINITIONS ::= BEGIN

-- Initial definitions for Structure of Management Information.
-- This contains only structure information, no managed objects are defined

IMPORTS
   DisplayString
       FROM RFC1213-MIB
   enterprise
       FROM RFC1155-SMI;

-- EXPORTS EVERYTHING
```

A.2 STRUCTURE INFORMATION

```
nema                    OBJECT IDENTIFIER  ::=  { enterprise 1206 }
-- NEMA has received ID 1206 from IANA
-- NEMA starts at { iso org dod internet private enterprise 1206 } in the
-- global naming tree.

nemaMgmt                OBJECT IDENTIFIER  ::=  { nema 1 }
-- The mgmt subtree is used for standard NEMA object definitions that span
-- different NEMA sections.

nemaExperimental        OBJECT IDENTIFIER  ::=  { nema 2 }
-- The experimental subtree is used for technical committees developing
-- objects that may become standard in the future.

nemaPrivate             OBJECT IDENTIFIER  ::=  { nema 3 }
-- The private subtree is used for unilaterally defined objects. One common
-- use is for manufacturer-specific or customer-specific MIB definitions.

transportation          OBJECT IDENTIFIER  ::=  { nema 4 }
-- The transportation subtree is used by the NTCIP to define
-- standard objects specific for the transportation industry.

Protocols         OBJECT IDENTIFIER  ::=  { transportation 1 }
devices           OBJECT IDENTIFIER  ::=  { transportation 2 }
tcip              OBJECT IDENTIFIER  ::=  { transportation 3 }
tmdd              OBJECT IDENTIFIER  ::=  { transportation 4 }
layers            OBJECT IDENTIFIER  ::=  { protocols 1 }
application       OBJECT IDENTIFIER  ::=  { layers 7 }
profiles          OBJECT IDENTIFIER  ::=  { protocols 2 }
dynObjMgmt        OBJECT IDENTIFIER  ::=  { protocols 3 }
ntcipTraps        OBJECT IDENTIFIER  ::=  { protocols 4 }
```

```
asc               OBJECT IDENTIFIER  ::=  { devices 1 }
ramp              OBJECT IDENTIFIER  ::=  { devices 2 }
dms               OBJECT IDENTIFIER  ::=  { devices 3 }
tss               OBJECT IDENTIFIER  ::=  { devices 4 }
ess               OBJECT IDENTIFIER  ::=  { devices 5 }
global            OBJECT IDENTIFIER  ::=  { devices 6 }
cctv              OBJECT IDENTIFIER  ::=  { devices 7 }
cctvSwitch        OBJECT IDENTIFIER  ::=  { devices 8 }
dcm               OBJECT IDENTIFIER  ::=  { devices 9 }
ssm               OBJECT IDENTIFIER  ::=  { devices 10 }
scp               OBJECT IDENTIFIER  ::=  { devices 11 }
networkCamera     OBJECT IDENTIFIER  ::=  { devices 12 }
elms              OBJECT IDENTIFIER  ::=  { devices 13 }

chap              OBJECT IDENTIFIER  ::=  {layers 1}
modem             OBJECT IDENTIFIER  ::=  {layers 2}
```

A.3 COMMON TEXTUAL CONVENTIONS

```
OerString ::= OCTET STRING

Byte      ::= INTEGER (-128..127)
Ubyte     ::= INTEGER (0..255)
Short     ::= INTEGER (-32768..32767)
Ushort    ::= INTEGER (0..65535)
Long      ::= INTEGER (-2147483648..2147483647)

BITMAP8   ::= OCTET STRING (SIZE (1))
BITMAP16  ::= OCTET STRING (SIZE (2))
BITMAP32  ::= OCTET STRING (SIZE (4))

OwnerString ::= DisplayString (SIZE (0..127))
-- This data type is used to model an administratively assigned name of the
-- owner of a resource. This information is taken from the NVT ASCII
-- character set. It is suggested that this name contain one or more of the
-- following: management station name, manager’s name, location or phone
-- number.
--
-- SNMP access control is articulated entirely in terms of the contents of
-- MIB views; access to a particular SNMP object instance depends only upon
-- its presence or absence in a particular MIB view and never upon its value
-- or the value of related object instances. Thus, objects of this type
-- afford resolution of resource contention only among cooperating managers;
-- they realize no access control function with respect to uncooperative
-- parties.
--
-- Objects with this syntax are declared as having
--      SIZE ( 0..127 )
```

```
RowStatusStatic  ::= INTEGER
                   { other (1),
                     standby (2),
                     available (3),
                     invalid (4),
                     activate (5),
                     deactivate (6)}

END -- NTCIP8004v02 Definitions
```

Annex B
NTCIP OBJECT TYPE MACRO EXAMPLES
[Informative]

The following examples show the form of the OBJECT TYPE Macro with possible subfields in the DESCRIPTION field.

```
globalSetIDParameter   OBJECT-TYPE
     SYNTAX   INTEGER (0..65535)
     ACCESS   read-only
     STATUS   optional
     DESCRIPTION "<Definition> Specifies a relatively unique ID for all user-
            changeable parameters of the particular device-type currently
            implemented in the device. Often this ID is calculated using a
            CRC algorithm.
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.1"
   ::= {globalConfiguration 1}

maxPhases   OBJECT-TYPE
     SYNTAX   INTEGER (0..255)
     ACCESS   read-only
     STATUS   mandatory
     DESCRIPTION "<Definition> The Maximum Number of Phases this Actuated
            Controller Unit supports. This object indicates the maximum rows
            which shall appear in the phaseTable object.
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.1.1.1"
     ::= {phase 1}

dmsSignType  OBJECT-TYPE
     SYNTAX  INTEGER{
                 other (1),
                 bos (2),
                 cms (3),
                 vmsChar (4),
                 vmsLine (5),
                 vmsFull (6),
                 portableOther (129),
                 portableBOS (130),
                 portableCMS (131),
                 portableVMSChar (132),
                 portableVMSLine (133),
                 portableVMSFull (134)}
     ACCESS   read-only
     STATUS   optional
     DESCRIPTION "<Definition> Indicates the type of sign. The descriptions
         are:
         other:   Device not specified through any other definition, refer
                  to device manual,
         bos:     Device is a Blank-Out Sign,
         cms:     Device is a Changeable Message Sign,
         vmsChar: Device is a Variable Message Sign with character matrix
                  setup,
         vmsLine: Device is a Variable Message Sign with line matrix setup,
         vmsFull: Device is a Variable Message Sign with full matrix setup.
```



```
        Same is true for all portable signs.
    <Object Identifier> 1.3.6.1.4.1.1206.4.2.3.1.2"
::= { dmsSignCfg 2 }

essNtcipNum   OBJECT-TYPE
     SYNTAX   IpAddress
     ACCESS   read-write
     STATUS   mandatory
     DESCRIPTION "<Definition> The unique IP Address of the station. This
              will make duplication of a BUFR identification number less
              likely to appear.
               <Object Identifier> 1.3.6.1.4.1.1206.4.2.5.2.1.1"
     ::= {essNtcipIdentification 1}

rangeMaximumPreset  OBJECT-TYPE
      SYNTAX    INTEGER (0..255)
      ACCESS    read-only
      STATUS    mandatory
      DESCRIPTION  "<Definition> A preset is the pre-specified position where
              a camera is pointed to a fixed point in space (includes
              positions for pan, tilt, and zoom). The maximumPreset is a
              number indicating the total number of possible preset positions
              supported by the device. A value of zero (0) identifies that
              the device does not support presets.
             <Object Identifier> 1.3.6.1.4.1.1206.4.2.7.1.1"
     ::= {cctvRange 1}

rmcCommRefreshThreshold OBJECT-TYPE
      SYNTAX   INTEGER (0..65535)
      ACCESS   read-write
      STATUS   mandatory
      DESCRIPTION  "<Definition> This is a global parameter that is being
               used for each metered lane. If no successful SET command
               communication has been issued to the rmcCommAction-object of
               a metered lane for the time indicated in this threshold, the
               value of the rmcCommActionStatus-object for that metered lane
               shall be set to 'noComm'. A suggested usable range is 0, 20 to
               14400 in 1-second increments. A value of zero (0) shall
               inhibit communications refresh timing.
              <Object Identifier> 1.3.6.1.4.1.1206.4.2.2.1.1
              <Unit> Second"
      REFERENCE   "see Annex A.3"
      ::={rmcCfg 1}

inputStatus   OBJECT-TYPE
SYNTAX   OCTET STRING (SIZE(1))  -- BITMAP8
ACCESS   read-only
STATUS   mandatory
DESCRIPTION   "<Definition> Input status denotes a bitmapped value that
indicates the current state of eight (8) user defined discrete
inputs, as outlined below:
<Format>
Bit Number  Name             Description
Bit7  Input 8    0 = OFF, 1 = ON for the active status of discrete Input 8,
Bit6  Input 7    0 = OFF, 1 = ON for the active status of discrete Input 7
Bit5  Input 6    0 = OFF, 1 = ON for the active status of discrete Input 6,
Bit4  Input 5    0 = OFF, 1 = ON for the active status of discrete Input 5,
```



```
Bit3  Input 4    0 = OFF, 1 = ON for the active status of discrete Input 4,
Bit2  Input 3    0 = OFF, 1 = ON for the active status of discrete Input 3,
Bit1  Input 2    0 = OFF, 1 = ON for the active status of discrete Input 2,
Bit0  Input 1    0 = OFF, 1 = ON for the active status of discrete Input 1.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.7.7.1”
::= {cctvSwitchInput 1}

sensorSystemReset  OBJECT-TYPE
SYNTAX   INTEGER { restart (1),
                   reinitialize (2),
                   shortDiagnostics (3),
                   fullDiagnostics (4),
                   commandComplete (255) }
ACCESS   read-write
STATUS   mandatory
DESCRIPTION  "<Definition> This object commands the entire sensor system to
reset. Reading a value other than 255 indicates the presence of a pending or
 executing command. A write to this object is not allowed unless the current
 value is 255. The reset commands are described as follows:
 Value    Meaning
 1        Restart command shall cause the unit to enter restart state and
          go through a complete shut-down and start-up process;
 2        Reinitialization commands hall cause the unit to flush all volatile
          memory and reinitialize all settings to their default values;
 3        Short diagnostics command shall cause the unit to go through an
          abbreviated vendor specific diagnostic routine which may be the
          same as full diagnostics for some vendors;
 4        Full diagnostics shall cause the unit to go through a complete
          vendor specific diagnostic routine which may be the same as
          short diagnostics for some vendors;
 5..128   Reserved for future use;
 255      Command complete shall be the default-state for which writing
          causes no action.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.4.1.1"
::= {tssSystemSetup 1 }

controlModeTable  OBJECT-TYPE
    SYNTAX   SEQUENCE OF ControlModeEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION  "
       <Definition> This object describes a static table of
       parameters that control and define the plan and special
       function values to put into effect.

       <TableType> static
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.10.10.1.0"
::= { controlMode 1 }

controlModeEntry  OBJECT-TYPE
    SYNTAX   ControlModeEntry
    ACCESS   not-accessible
    STATUS   mandatory
    DESCRIPTION  "
       <Definition> This object describes the parameters in the rows
       of the Mode Table.
```



```
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.10.10.1.1.0"
    INDEX  { sectionNumber}
::= { controlModeTable 1 }
```

Annex C
BLOCK OBJECT EXAMPLES
[Informative]

As defined in Section 2.3.3, the definition of a Block Object consists of two parts. The first part uses a slightly modified version of the ASN.1 Macro that is defined in RFC 1212 and is placed in the body of a MIB Module. The second part uses a textual convention that is placed outside the body of the MIB.

C.1 MAP DATA EXAMPLE

The following provides an example of how a block object for the data meant for a traffic signal control map display might be defined

MIB Definition:

```
EXAMPLE-MIB DEFINITIONS ::= BEGIN

  IMPORTS
        OBJECT-TYPE
        FROM RFC-1212
        OerString
        FROM TMIB-II;

  signal1-8StatusBlock    OBJECT-TYPE
        SYNTAX      OerString
        ACCESS      read-only
        STATUS      mandatory
                    DESCRIPTION "<Definition>  An OER encoded string
                    containing signal indications for phases 1-8 and a
                    time stamp. The indexes for the columnar objects can
                    take on the value 1 through 8.
                    <<Object Identifier>"

        ::= {phase 12}

  split1PhaseDataBlock    OBJECT-TYPE
        SYNTAX      OerString
        ACCESS      read-write
        STATUS      mandatory
                    DESCRIPTION "<Definition> An OER encoded string
                    containing a the split time, split mode and coord
                    phase programming for the phases associated with
                    split 1. The index can  take on the value of any
                    phase number.
                    <Object Identifier>"

        ::= {phase 13}
  END
```

The DESCRIPTION clearly states the indexes for any columnar objects and the values used.

The external, textual portions that are encoded as an OerString and would appear outside of any module definition might appear as follows:

External Definitions:

```
      SignalStatus1-8 :: = SEQUENCE {
            phaseStatusGroupGreens.x,
            phaseStatusGroupYellows.x,
            phaseStatusGroupWalks.x,
            phaseStatusGroupVehCalls.x,
            phaseStatusGroupPedCalls.x,
            overlapStatusGroupGreens.x,
            overlapStatusGroupYellows.x,
            globalTime.0                        -- @NTCIP1202 v02
            }

      Split1PhaseData :: = SEQUENCE {
            splitNumber.1.x,
            splitPhase.1.x,
            splitTime.1.x,
            splitMode.1.x,
            splitCoordPhase.1.x
            }
```

C.2 PHASE TIMING DATA EXAMPLE

The following is an example of how a block object can be defined as self-indexing and then used to retrieve all timing data associated with any enabled phase.

MIB Definition:

```
      databaseGroupPhaseBlock OBJECT-TYPE
            SYNTAX      OerString
            ACCESS      read-only
            STATUS      mandatory
DESCRIPTION " <Definition> An OER encoded string containing phase parameters
for phases that are enabled. By use of the extension marker (…), a phase's
data may include parameters that are not defined in this current definition.
This could apply to agency or vendor specific parameters.

This block object is self-indexing. The phaseNumber in each sequence defines
the index for subsequent data. The phase numbers included in the frame are
defined by the current state of bit 0 of the phaseOption object.

            <Object Identifier> "
      ::= {phase 12}
```

External Definition encoded as OerString:

```
      DatabaseGroupPhasesBlock ::= SEQUENCE OF PhaseTimingParameters
                              -- Indexes of SEQUENCE OF are defined by
                  enabled phases

      PhaseTimingPameters ::= SEQUENCE {
            phaseNumber.x,
            phaseWalk.x,
            phasePedestrianClear.x,
            phaseMinimumGreen.x,
            phasePassage.x,
```

```
            phaseMaximum1.x
            phaseMaximum2.x,
            phaseYellowChange.x,
            phaseRedClear.x,
            phaseRedRevert.x,
            phaseAddedInitial.x,
            phaseMaximumInitial.x,
            phaseTimeBeforeReduction.x,
            phaseCarsBeforeReduction.x OPTIONAL,
            phaseTimeToReduce.x,
            phaseReduceBy.x OPTIONAL,
            phaseMinimumGap.x,
            phaseDynamicMaxLimit.x OPTIONAL,
            phaseDynamicMaxStep.x OPTIONAL,
            phaseStartup.x,
            phaseOptions.x,
            ... }
    END
```


Annex D
UML INFORMATION
[Informative]

It is beyond the scope of NTCIP 8004 v02 to provide a rigorous or complete explanation of the UML Diagrams used within NTCIP 8004 v02. However, *UML basics: An introduction to the Unified Modeling Language*, and *The Unified Modeling Language User Guide* may provide a more complete understanding of UML. See Section 1.2.2.

Annex E
HISTORY OF CHANGES
[Informative]

Annex E summarizes the changes made between the prior version of NTCIP 8004 and NTCIP 8004 v02:

E.1 TITLE PAGE

The document status, version, and date were updated.

E.2 NOTICES

The new Notices section was added. However, the PRL and RTM copyright notice was deleted because NTCIP 8004 v02 does not contain such tables.

E.3 ACKNOWLEDGEMENTS

Individuals, agencies, companies, or universities contributing or commenting on NTCIP 8004 v02 were modified to reflect current status.

E.4 FOREWORD

History was updated.

E.5 SECTION 1.1 SCOPE

Added "and other NTCIP" to scope.

E.6 SECTION 1.2.1 NORMATIVE REFERENCES

a) Removed reference to ISO 14817. Aided by NTCIP Joint Committee decision in 2007, BSP2 WG removed all ISO 14817 references because data fields other than those defined in NTCIP 8004 v02 are not necessary for center-to-field communications, which NTCIP 8004 v02 addresses.
b) Added reference to ISO 1000:1992 as the reference basis for the data field <unit>.
c) Updated the versions of referenced standards to the now valid versions.

E.7 SECTION 1.2.2 OTHER REFERENCES

a) Updated the versions of referenced NTCIP standards to the current versions.
b) Added reference citations to other references contained in NTCIP 8004 v02.

E.8 SECTION 1.3 DEFINITIONS AND ACRONYMS

a) Added several definitions, particularly 'obsolete' and 'deprecated'.
b) Deleted some of the definitions no longer applicable.

E.9 SECTION 2.1 AND FOLLOWING

Removed ISO 14817 reference (see Annex E.6), and any text containing this reference was modified.

E.10 SECTION 2.2.1 NAMING TREE ADMINISTRATIVE NODES

a) Updated Figure 2 to reflect changes related to joint-iso-itu-t node, and other updates.
b) Added Notes to explain relationship between 1.16.840 and 2.16.840, and to reflect additional names.

E.11 SECTION 2.3.2 SIMPLE OBJECT SPECIFICATION

a) Removed reference to ISO 14817 and associated meta attributes no longer necessary.
b) Added the <Object Identifier> subfield definition.
c) Created separate items for the <table type> and <Unit> subfield definitions.

E.12 SECTION 2.3.3 BLOCK OBJECT SPECIFICATION

a) Removed reference to ISO 14817 and associated meta attributes no longer necessary.
b) Added the <Object Identifier> subfield definition.

E.13 SECTION 2.3.4 DYNAMIC OBJECT SPECIFICATION

a) Removed reference to ISO 14817 and associated meta attributes no longer necessary.
b) Added the <Object Identifier> subfield definition.

E.14 SECTION 2.4 META ATTRIBUTES

a) Removed reference to ISO 14817 and associated meta attributes no longer necessary.
b) Added text for the <Object Identifier> subfield definition.

E.15 SECTION 2.4.3 UNIT OF MEASURE

Expanded on definition of this subfield.

E.16 SECTION 2.4.4 BITMAP FORMAT

Expanded the definition and explanation of bitmap setup used within SYNTAXes of INTEGER versus OCTET STRING, and added diagram to further explain the difference.

E.17 SECTION 2.5 MANAGEMENT INFORMATION BASE (MIB)

Removed references to ISO 14817 meta attributes.

E.18 SECTION 2.6 MULTI-VERSION INTEROPERABILITY (MVI, BACKWARD COMPATIBILITY)

Added this entire Section defining how and under what circumstances object definitions might be deprecated to ensure MVI.

E.19 SECTION 3.2 TABLES, ROWS, AND BLOCK OBJECTS

a) Removed reference to ISO 14817 and associated meta attributes no longer necessary.
b) Added the <Object Identifier> subfield definition.

E.20 SECTION 3.3.1 ROW STATUS IN STATIC TABLES

Updated Figures 9, 10, and 11.

E.21 ANNEX A.1 SMI HEADER

Revised version number and history, and added tmdd node under the transportation node.

E.22 ANNEX B NTCIP OBJECT TYPE MACRO EXAMPLES

Updated all examples to comply with preceding modifications. Removed ISO 14817 reference and associated meta attributes no longer necessary, and added <Object Identifier> subfield definition.

E.23 ANNEX C BLOCK OBJECT EXAMPLES

Updated examples to comply with preceding modifications. Removed reference to ISO 14817 and associated meta attributes no longer necessary, and added the <Object Identifier> subfield definition.

E.24 ANNEX E HISTORY OF CHANGES

Added section showing changes between the prior version of NTCIP 8004, and NTCIP 8004 v02.

§