A Joint Information Management Policy of AASHTO, ITE, and NEMA

NTCIP 8004 version v01

National Transportation Communications for ITS Protocol

Structure and Identification of Management Information (SMI)

published in May 2008

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 22209-3806

file version 8004 v01.38r rita -2008c 0512

NOTICES

Copyright Notice

Visit www.ntcip.org for other copyright information, for instructions to request reprints of excerpts, and to request reproduction that is not granted below.

PDF File License Agreement

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of an Adobe® Portable Document Format (PDF) electronic data file (the “PDF File”), AASHTO / ITE / NEMA authorizes each registered PDF File user to view, download, copy, or print the PDF File available from the authorized Web site, subject to the terms and conditions of this license agreement:

(a) you may download one copy of each PDF File for personal, noncommercial, and intraorganizational use only;
(b) ownership of the PDF File is not transferred to you; you are licensed to use the PDF File;
(c) you may make one more electronic copy of the PDF File, such as to a second hard drive or burn to a CD;
(d) you agree not to copy, distribute, or transfer the PDF File from that media to any other electronic media or device;
(e) you may print one paper copy of the PDF File;
(f) you may make one paper reproduction of the printed copy;
(g) any permitted copies of the PDF File must retain the copyright notice, and any other proprietary notices contained in the file;
(h) the PDF File license does not include (1) resale of the PDF File or copies, (2) republishing the content in compendiums or anthologies, (3) publishing excerpts in commercial publications or works for hire, (4) editing or modification of the PDF File except those portions as permitted, (5) posting on network servers or distribution by electronic mail or from electronic storage devices, and (6) translation to other languages or conversion to other electronic formats;
(i) other use of the PDF File and printed copy requires express, prior written consent.

Data Dictionary and MIB Distribution Permission

To the extent that these materials are distributed by AASHTO / ITE / NEMA in the form of a Data Dictionary (“DD”) or Management Information Base (“MIB”), AASHTO / ITE / NEMA extend the following permission:

You may make or distribute unlimited copies, including derivative works, of the DD or MIB, including copies for commercial distribution, provided that:
(i) each copy you make or distribute contains the citation “Derived from NTCIP 0000 [insert the standard number]. Used by permission of AASHTO / ITE / NEMA.”;
(ii) 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;
(iii) 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;
(iv) 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 / 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 competent professional in determining the exercise of reasonable care in any given circumstances. Information and other standards on the topic covered by this publication may be available from other sources, which the user may wish to consult for additional views or information not covered by this publication.

AASHTO, ITE, and NEMA have no power, nor do they undertake to police or enforce compliance with the contents of this document. AASHTO, ITE, and NEMA do not certify, test, or inspect products, designs, or installations for safety or health purposes. Any certification or other statement of compliance with any health or safety-related information in this document shall not be attributable to AASHTO, ITE, or NEMA and is solely the responsibility of the certifier or maker of the statement.

Trademark Notice

NTCIP is a trademark of AASHTO / ITE / NEMA. All other marks mentioned in this standard are the trademarks of their respective owners.

ACKNOWLEDGEMENTS

NTCIP 8004 v01 was prepared by the NTCIP Base Standards and Profiles 2 (BSP2) Working Group, 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 v01 was prepared, the following individuals were members of the NTCIP BSP2 Working Group:

- Bob De Roche (Chair)
- Bud Kent
- Greg Mizell
- Alex Mousadi
- Joerg “Nu” Rosenbohm
- Kenneth Vaughn

Other individuals providing input to the document include:

- Jas Bhullar
- Doug Crawford
- Blake Christie
- Bob Force
- Mike Robinson
- Brian Paulsmeyer
- Mike Robinson
- Sonja Sun
- Hoi Wong
- Paul Lim-Hing
- Bob Martindale †
- Amit Misra
- Jeff Thorn

In addition to the many volunteers, recognition is also given to those organizations which supported the effort of the working group by providing funding for the standard, providing travel support, and submitting comments, including:

- U.S. Department of Transportation, Research and Innovative Technology Administration
- Caltrans
- Econolite Control Products
- Iteris
- Mastec
- Ministry of Transportation, Ontario
- Mitretek
- New York Department of Transportation
- PB Farradyne Inc
- Peek Traffic Systems, Inc.
- Robert De Roche Consulting, Inc.
- Siemens ITS
- Southwest Research Institute
- Texas Department of Transportation
- Trevilon
- URS

FOREWORD

NTCIP 8004 v01 defines a process and control standard for the rules and procedures for defining structure and identification of management information in transportation related communications. It defines requirements that are applicable to all NTCIP devices that exchange data in that environment. It also contains optional and conditional clauses that are applicable to specific environments for which they are intended.

NTCIP 8004 v01 uses only metric units, and NTCIP 8004 v01 includes five annexes; however, only Annex A is normative.

NTCIP 8004 v01 is designated, and should be cited as NTCIP 8004 v01. NTCIP 8004 v01 is an adopted and published NTCIP Policy, reflecting the resolution of comments received during the development and approval process, when drafts were designated using a major and minor version number (v01.01 through v01.37). To distinguish NTCIP 8004 v01 from previous drafts, NTCIP 8004 v01 also includes NTCIP 8004 v01.38 on each page header.

During early stages of development, portions of NTCIP 8004 v01 were part of NTCIP 1101:1997, which was also numbered and referenced (designated and cited) as NEMA TS 3.2-1996. However, in order to provide a more systematic approach to an organized numbering scheme, and to reflect the joint copyright held by AASHTO, ITE, and NEMA, NTCIP 8004 v01 is now the appropriate designation.

NTCIP 8004 v01 is an NTCIP Process, Control, and Information Management document. Process, Control, and Information Management documents define the practices and policies used by the NTCIP Joint Committee in developing and maintaining NTCIP standards and documents; these are authorized for publication by AASHTO, ITE, and NEMA following recommendation by the Joint Committee of the NTCIP. NTCIP Process, Control, and Information Management documents are not distributed for industry-wide ballot. An NTCIP Process, Control, and Information Management document equates to the following standard types:

AASHTO – Recommended Practice
ITE – Informational Report
NEMA – Authorized Engineering Information

For more information about NTCIP standards, visit the NTCIP Web Site at www.ntcip.org.

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 standard publication’s content. A “Request for Interpretation” of this standard 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 will be 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 will be 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 disposition may be available for reference and information at www.ntcip.org.

A User Comment should be submitted to this address:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, Virginia 22209-3806
e-mail: ntcip@nema.org

A User Comment should be submitted in the following form:

Standard 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 v01 was separately balloted and approved by AASHTO, ITE, and NEMA after recommendation by the Joint Committee on the NTCIP. Each organization has approved this standard as the following standard type, as of the date:

AASHTO – Recommended Practice; November 2005
ITE – Informational Report; October 2005
NEMA – Authorized Engineering Information; December 2005

History

NTCIP 8004 v01 was originally part of NTCIP 1101, Simple Transportation Management Framework. A work item to separate 8004 v01 into a stand-alone document was approved by the Joint Committee on July 29, 1999.

NTCIP 8004 v01.32, February 2004 – Accepted as User Comment Draft by the Joint Committee on the NTCIP.

NTCIP 8004 v01.33, June 2004 – Made editorial changes to front matter and corrected miscellaneous typographical and grammatical errors.

NTCIP 8004 v01.36, March 2005 – Accepted as Recommended Standard by the Joint Committee on the NTCIP.

NTCIP 8004 v01, March 2005 – The Annex B DMS example was revised per instructions of the Joint Committee. In July 2005, Standards Bulletin B0106 distributed version v01.37for balloting and approval. NTCIP 8004 v01 was Jointly Approved in 2005. May 2008 – 8004 v01.38 was edited and formatted for publication.

Compatibility of Versions

To distinguish NTCIP 8004 v01 (as published) from previous drafts, NTCIP 8004 v01 also includes NTCIP 8004 v01.38 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.

Anyone using NTCIP 8004 v01 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 v01 should also consult NTCIP 8004 v01 for specific guidelines on compatibility.

NOTE—The above reference to NTCIP 8004 v01 is template information. The circular nature of the reference is recognized.

INTRODUCTION

NTCIP 8004 v01 is an NTCIP Process, Control, and Information Management document. Process, Control, and Information Management documents define the practices and policies used by the NTCIP Joint Committee in developing and maintaining NTCIP standards and document. NTCIP 8004 v01 is intended for use in association with certain Application Layer protocols in transportation-related communications such as SNMP, STMP, SFMP, FTP, and TFTP.

NTCIP 8004 v01 contains mandatory requirement statements that are applicable to all devices claiming conformance to this standard. It also contains options and conditional requirements, which may be applicable to a specific environment in which a device is used.

The following keywords apply to this document: AASHTO, Dynamic Objects, ITE, NEMA, NTCIP, SMI, SNMP, STMP, Process and Control Standard.

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. After the NTCIP Class B Profile was published, the Joint Committee 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 merged to form the Base Standards and Profiles 2 (BSP2) WG. The first BSP2 WG meeting was in January 1999.

The BSP2 WG submitted a work plan and initiated development of NTCIP 8004 v01 in January 2000.

< This page is intentionally left blank. >

CONTENTS

Section 1
GENERAL

1.1 SCOPE

NTCIP 8004 v01 is applicable to the NTCIP 1200 series and other NTCIP standards that deal with device data dictionaries. NTCIP 8004 v01 defines the structure and identification of information used in transportation related devices.

1.2 REFERENCES

For revision information on this NTCIP standard, contact:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3806
e-mail: ntcip@nema.org

For draft revisions to this NTCIP standard, and recommended revisions of the NTCIP Joint Committee, visit www.ntcip.org.

1.2.1 Normative References

The following standards (normative references) contain provisions that, through reference in this text, constitute provisions of this NTCIP standard. Other documents and standards (other references) which are referenced in NTCIP 8004 v01 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.

IAB STD 15	(RFC 1157) *Simple Network Management Protocol,* J. Case, M. Fedor, M. Schoffstall, J. Davin, May 1990
IAB STD 16	(RFC 1155) Structure and Identification of Management Information for TCP/IP based Internets, M. Rose, K. McCloghrie, May 1990, (RFC 1212) Concise MIB Definitions, M. Rose and K. McCloghrie, March 1991
ISO/IEC 8824-1:1998	Information technology -- Abstract Syntax Notation One (ASN.1): Specification of basic notation
ISO/IEC 8824-2:1998	Information technology -- Abstract Syntax Notation One (ASN.1): Information object specification
ISO/IEC 8824-3:1998	Information technology -- Abstract Syntax Notation One (ASN.1): Constraint specification
ISO/IEC 8824-4:1998	Information technology -- Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 specification

ISO 14817:2002 (E)	Transport information and control systems — Requirements for TICS central data registry and TICS data dictionaries
NTCIP 1102 v01.11	Octet Encoding Rules (OER) Base Protocol

1.2.2 Informative References

IEEE Std. 610.12-1990:	IEEE Standard Glossary of Software Engineering Terminology
NTCIP 2202:2001	Transport Profile for Internet (TCP/IP and UDP/IP)
NTCIP 2301:2001	Application Profile for Simple Transportation Management Framework
NTCIP 7001	Assigned Numbers - Part 1
NTCIP 8003:2001	Framework and Classification of Profiles
NTCIP 8005 v01.12	Creating Management Information Base Files and a Functional Area Data Dictionary
NTCIP 9001 v02.06	The NTCIP Guide
RFC 854	*Telnet Protocol specification*, J. Postel, J. Reynolds, May 1983

1.2.3 Contact Information

1.2.3.1 ISO/IEC Standards

Members of the ISO maintain registers of currently valid ISO/IEC International Standards. For the USA, 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, New York 10036
(212) 642-4900
http://global.ihs.com

1.2.3.2 RFC Documents

Obtain Request for Comment (RFC) electronic documents from several repositories on the World Wide Web, or by "anonymous" File Transfer Protocol (FTP) with several hosts. Browse or FTP to:

http://www.ietf.org/rfc.html
http://www.rfc-editor.org/
http://www.rfc-editor.org/repositories.html
for FTP sites, read ftp://ftp.isi.edu/in-notes/rfc-retrieval.txt

1.2.3.3 NTCIP Documents

NTCIP documents may be obtained by contacting NEMA at the following address:

National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3801

1.3 DEFINITIONS

For the purposes of NTCIP 8004 v01, the following terms and definitions apply. For terms not defined in Section 1.3, 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 Section 1.3 or in *Webster's New Collegiate Dictionary* are used in accordance with their definitions in IEEE Std 100-19962.

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. It is the NTCIP representation of an ISO 14817 data frame.

compatible The ability of two or more systems or components to exchange information. [Per IEEE Std. 610.12-1990.]

data Information before it is interpreted.

data element Some single unit of information of interest (such as a fact, proposition, observation, etc.) about some (entity) class of interest (e.g., a person, place, process, property, concept, association, state, event). A data element is considered to be indivisible in a particular context. [Per ISO 14817.]

NOTE—A data element is represented by an object class, a property of the represented object class and a value domain.

NOTE—For NTCIP 8004 v01 purposes, the SNMP Object Type Macro is used to define the entity type, property, and an explicit value domain term.

data frame A grouping of data elements primarily for the purpose of referring to the group with a single name, and thereby efficiently reusing groups of data elements that commonly appear together (as an ASN.1 SEQUENCE, SEQUENCE OF, SET, SET OF or CHOICE) in a message specification. [Per ISO 14817.]

NOTE—This data concept type may be used to specify groups of data elements for other purposes as well.

NOTE—In the context of SNMP, a data frame consists of only the information (variable bindings) to be exchanged. It does not include the application layer header.

datagram A self-contained unit of data transmitted independently of other datagrams.

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 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 the environmental impacts.

International Organization for Standardization (ISO)
An international standards organization.

NOTE—ANSI is the primary interface to ISO within the United States.

NOTE—Often thought to be International Standards Organization because of the usage ISO for short.

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
The network protocol offering a connection-less mode network service in the Internet suite of protocols.

Interoperable
The ability of two or more systems or components to exchange information and use the information that has been exchanged [Per IEEE Std. 610.12-1990.]

IP address
A 32-bit quantity used to represent a point of attachment in an internet.

management information base
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.

message
A message is a grouping of data elements and/or data frames, as well as associated message metadata that is used to convey a complete unit of information. [Per ISO 14817]

NOTE—For purposes of NTCIP 8004 v01, a message is an abstract description; it is not a specific instance.

NOTE—In the context of SNMP, a message consists of the application layer header and the information (variable binding) to be exchanged.

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 data frame.

OBJECT IDENTIFIER
A unique name (identifier) that is associated with each type of object in a Management Information Base that is a defined ASN.1 type.

object type
A data structure used to describe the attribute or properties of a data element or group of data elements.

OBJECT-TYPE
A macro used to define the meta attributes of an object in an SNMP Management Information Bases.

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 A communications protocol developed by the Internet Engineering Task Force (IETF), used for configuration and monitoring of network devices.

Simple Transportation Management Framework The organization of the information within devices and the methods of retrieving or modifying any information within the device.

NOTE—STMF also explains how to generate and utilize 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 ABBREVIATIONS AND ACRONYMS

The acronyms used in NTCIP 1104 8004 v01 are defined as follows:

ANSI	American National Standards Institute
ASCII	American National Standard Code for Information Interchange.
ASN.1	Abstract Syntax Notation One
IAB STD	Internet Activities 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
MIB	Management Information Base
NTCIP	National Transportation Communications for Intelligent Transportation Systems (ITS) Protocol
NVT	Network Virtual Terminal
OER	Octet Encoding Rules
OSI	Open Systems Interconnection
PICS	Protocol (or Profile) Implementation Conformance Specification
RFC	Request for Comments
SMI	Structure and Identification of Management Information
SNMP	Simple Network Management Protocol
STMF	Simple Transportation Management Framework.
STMP	Simple Transportation Management Protocol
TCP	Transport Control Protocol
TMIB	Transportation Management Information Base
UML	Unified Modeling Language

< This page is intentionally left blank. >

Section 2
CONFORMANCE

2.1 INTRODUCTION

NTCIP 8004 v01 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. It is based on IAB STD 16 and ISO/IEC 8824. NTCIP 8004 v01 intends that a high level of compatibility, including interoperability, be maintained between traffic management systems and field devices.

There are four major areas covered by NTCIP 8004 v01:

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

Within NTCIP, each piece of information is described by a number of characteristics. Some of these characteristics, the meta-attributes, are defined by ISO 14817 and define the logical definition of the information; other characteristics are defined by RFC 1212 and are used to represent this logical concept as an “object,” such that it can be used with the 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 atomic pieces of information are exchanged within the NTCIP protocols. In ISO 14817 terminology, these atomic pieces of information (without the NTCIP representation) are known as data elements**.** The details for defining simple objects are discussed in Sesction 2.3.2.

In some cases, it is desirable to predefine structures of data that can be referenced by a single identifier. In ISO 14817 terminology, these structures are known as data frames; their NTCIP representation is then 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**. Typically, a data frame is a predefined sequence of variables. In other words, the variables that make up the sequence and the order in which they appear is defined by a standard and does not change unless the standard is changed. STMP has extended this concept to allow the structure of some data frames 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 Section 5.

In each of the above 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 Manage Information Base (**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 also need to be able to share more complete definitions about each piece of data so that all users have a common understanding of the data. The format used for this purpose is a data dictionary as defined in ISO 14817. ISO 14817 requires that each data element be defined through the use of several meta-attributes (i.e., fields that describe various aspects of the data such as the definition, name, data type, etc.). The description of how NTCIP addresses these requirements is covered in Section 2.4.

2.1.1 Message Meta-Model

The relationship among many of the different terms can be depicted in the Unified Modeling Language, as depicted in Figure 1.

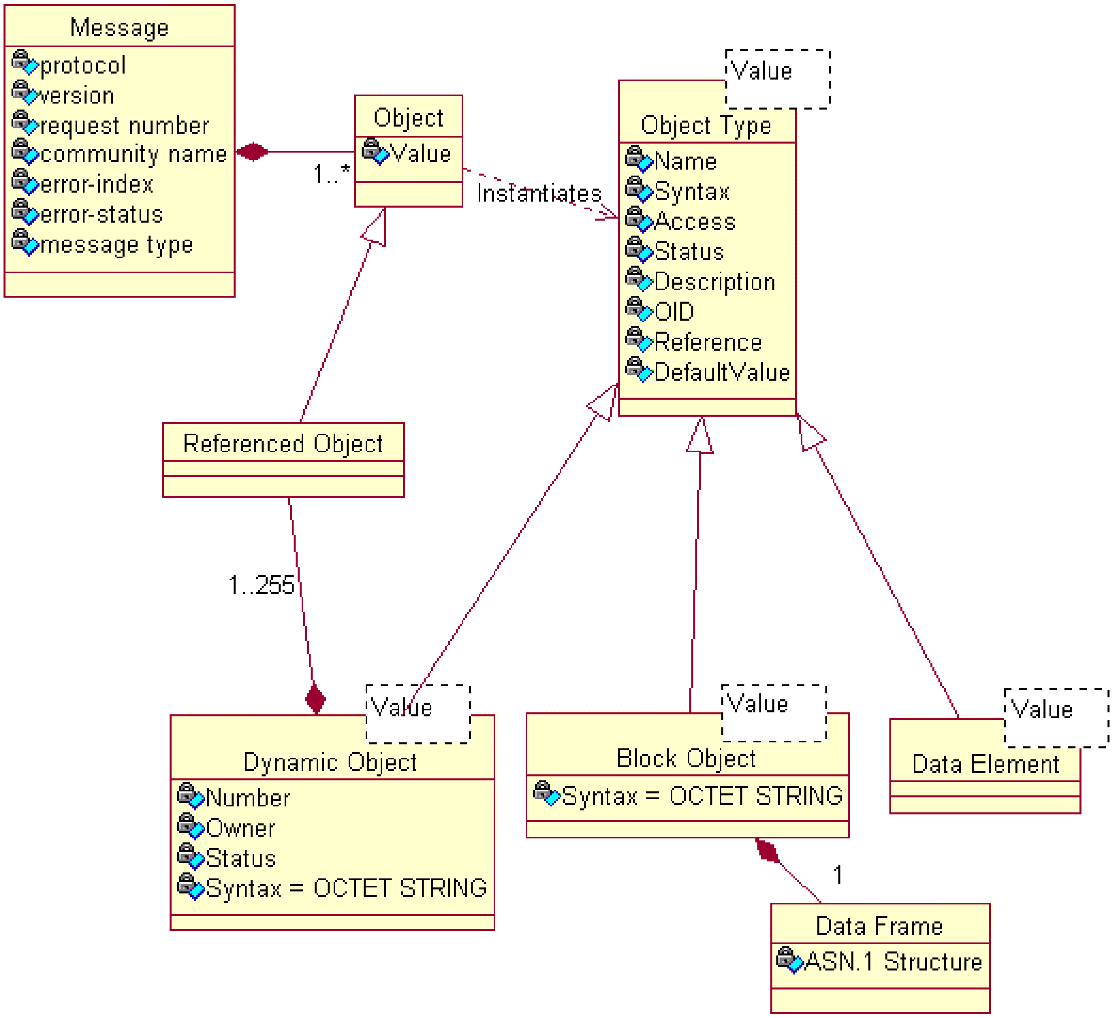


Figure 1
Relationships Among Terms

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 Data Frame (i.e., an ASN.1 data 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 must 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 some 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, CCITT, and jointly by the ISO and CCITT. The manager of any node may delegate the management responsibility for any sub-node underneath its branch. In this case, we may term the delegated node a sub-tree. Just as any sub-node may have its own set of sub-nodes, sub-trees can have their own sub-trees. This structure may continue to any level of depth in order 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 object 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 the International Telegraph and Telephone Consultative Committee (ccitt) and by the International Organization for Standardization (iso). The third node is jointly administered by the ISO and the CCITT (joint-iso-ccitt). Under the "iso" node, the ISO has designated one subtree for use by other (inter)national organizations (org). Under that subtree, one of the U.S. National Institutes 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 Activities 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 the IAB, NEMA has requested and received a node that it can administer. This node is defined as

nema OBJECT IDENTIFIER ::= { iso org dod internet private enterprises 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.

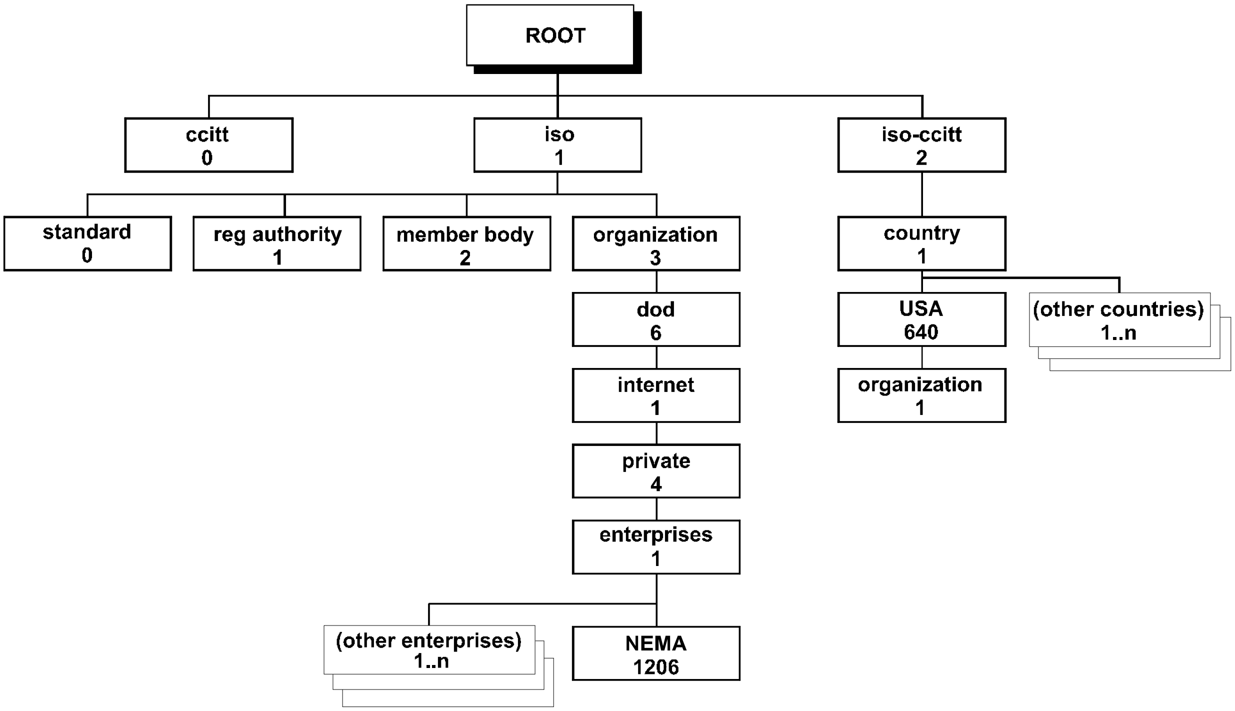


Figure 2
NEMA Authority Hierarchy

2.2.2 NEMA Node

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

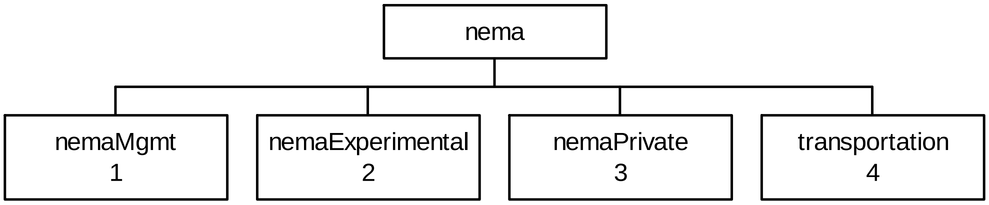


Figure 3
NEMA NODE

This represents the organization under the nema node at the time NTCIP 8004 v01 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 formal node number, can be put here. In the Internet community, multiple manufacturers must 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 in Section 1.2.

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, in order 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 its "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 enterprises number.

2.2.2.4 Transportation Node

The "nemaTransportation" 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.

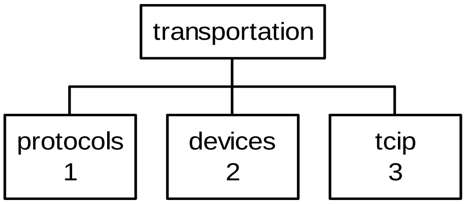


Figure 4
Transportation NODE

Figure 4 represents the organization under the transportation node at the time NTCIP 8004 v01 was written. See NTCIP 7001 for additional information.

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 the various layer in a protocol stack, profiles that cover several layers, and one specific to Dynamic Object Management. The first three nodes in the subtree are assigned to:

a) Layers
b) Profiles
c) Dynamic Object Management

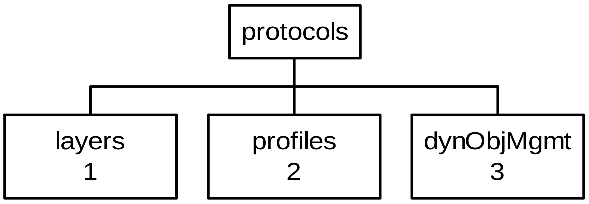


Figure 5
Protocols NODE

Figure 5 represents the organization under the protocols node at the time NTCIP 8004 v01 was written. See NTCIP 7001 for additional information.

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. 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.

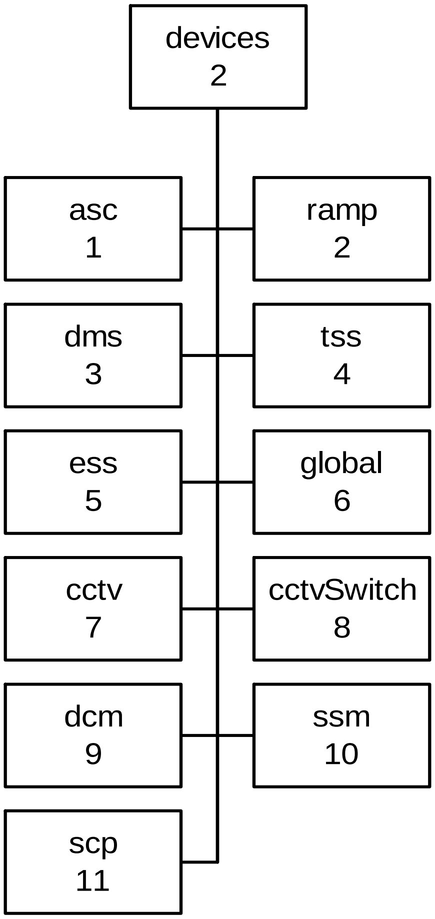


Figure 6
Devices NODE

Figure 6 represents only part of the organization under the devices node. See NTCIP 7001 for additional information.

2.2.2.4.3 TCIP Node

This 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 must 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 must be a lower-case letter, and the final character may not be a hyphen. Further, a hyphen may 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 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 straight forward. 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 IAB STD 16, the STATUS field can take on one of the following values:

a) mandatory
b) optional
c) deprecated
d) obsolete

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. A status of deprecated indicates that the subject object no longer represents the current design of the standard. A deprecated object is not required to be implemented. The status of a deprecated object will change to obsolete in some future version of the standard. A status of obsolete means that this object is no longer used and is no longer required to be implemented.

The decision whether an object is required should be defined in a conformance statement or PICS. . 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, 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 purposes, the DESCRIPTION field shall always be present and used. Other information that must be included in this field is defined in Section 2.4.

2.3.1.6 Macro INDEX

The INDEX clause, 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 textural cross-reference to a standard or document and specific clause 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 purposese, 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 data element as defined in ISO 14817 Section 4.7. 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. <DescriptiveName>
    3. <DataConceptType>
b) As appropriate, the DESCRIPTION field may include the Data Dictionary meta attribute for:
    1. <TableType>
    2. <Unit>
    3. <Format> (Bit Map)
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>, <DescriptiveName>, and <DataConceptType> shall be used and defined. The original DESCRIPTION is delimited by "<Definition>". The optional <TableType> subfield shall be included if the base object specification defines a table construct. 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. The optional Bit Map < 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 clause that provides information about specific range or values that quantify the object of interest.

For example, NEMA TS 2-1998 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 ISO 14817. 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 the following 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. <DescriptiveName>
    3. <DataConceptType> defined as a "Data Frame"
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 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. In order for a set operation on a block object to succeed, all the objects with the block object must 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 shall be defined by using the general structure of an ASN.1 TypeAssignment. However, a variation has been made in order to allow a clear and unambiguous reference to the objects defined in the MIB structure. Any Type or TypeReference field may be replaced with an NTCIPObjectReference as defined:

```
NTCIPObjectReference ::= NTCIPObjectName"."IndexSuffixes  ExternalReference
```

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

The IndexSuffixes shall be:

```
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 must be defined within the same module.

NOTE—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 preceding rules. 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 as 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) <DescriptiveName>
c) <DataConceptType> defined as "Data Element"

2.4 META ATTRIBUTES

For support of various data dictionaries, various meta-attribute subfields are added to the DESCRIPTION Field of various object specifications. These include:

a) Definition
b) Descriptive Name
c) Data Concept Type
d) ASN.1 Name
e) Data Type
f) Table Type
g) Unit of Measure
h) Format (Bit Map)

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. It 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

For support of various data registries, the following ISO 14817-defined meta-attribute subfields are added to each MIB Module:

a) Class Name
b) Classification Scheme Name
c) Classification Scheme Name Version
d) Descriptive Name Context
e) Source Standard
f) Last Change Date
g) Submitter Organization Name
h) Submitter Phone Number

NOTE—The meta attributes are also used to fill out other meta attributes that may or may not be discussed in NTCIP 8004 v01. For example, ASN.1 Name is automatically derived OBJECT DESCRIPTOR (see Section 2.3.1.1) as part of the process in entering the information in a data registry. NTCIP 8005 contains further information and how some other meta attributes are derived.

2.4.1 Definition

For conformance to ISO 14817 Subclause B.3.1, the DESCRIPTION field of the OBJECT-TYPE Macro as defined in IAB STD 16 (RFC 1212), shall be preceded by the delimiter <Definition> in order to separate it from other subfields that may now appear in the DESCRIPTION field.

2.4.2 Descriptive Name

The value of the DESCRIPTIVE NAME subfield is delimited by <DescriptiveName> and the values shall be as defined by ISO 14817 Section 9 and Subclause D.1.6 of that standard. The format is ObjectClassTerm.propertyTerm:value-domain-term. In the context of NTCIP, this format equates to Entity Type.Data Element: value-domain-term.

The appropriate value-domain-terms (and their abbreviations) for NTCIP are:

a) amount (amt)
b) code (cd)
c) identifier (id)
d) number (nbr)
e) percent (pct)
f) quantity (qty)
g) rate (rt)
h) text (txt)
i) frame (frm) (See Section 2.4.3.3.)

For fully-qualified descriptive names, an OrganizationIdentifier-DocumentIdentifier followed by "::" is prepended. For NTCIP 1201 through NTCIP 1211, the OrganizationIdentifier-DocumentIdentifiers are:

GLO	= NTCIP-1201::
ASC	= NTCIP-1202::
DMS	= NTCIP-1203::
ESS	= NTCIP-1204::
CCTV	= NTCIP-1205::
DCM	= NTCIP-1206::
RMC	= NTCIP-1207::
CCTVSWITCH	= NTCIP-1208::
TSS	= NTCIP-1209::
SSM	= NTCIP-1210::
SCP	= NTCIP-1211::

NOTE—The Descriptive Name and other meta attributes may be empty if the object is not part of a 1200 Series Standard.

2.4.3 Data Concept Type

The value of the DATA CONCEPT TYPE subfield is delimited by <DataConceptType> and the value shall be as defined in ISO 14817 Subclause B.3.8. The valid values for DATA CONCEPT TYPE, relevant to NTCIP are "object-class", "data-element" and "data-frame"; however, the terms "Entity Type", "Data Element", and "Data Frame" are substituted, respectively.

2.4.3.1 Entity Type

As defined in ISO 14817 Subclause 4.25, an Entity Type (Object Class) is a construct used to represent any kind of object (also referred to as an entity) within an ITS/TICS information environment.

ISO 14817 Subclause 7.5 further states, "An object class shall be a description of a set of objects that share the same properties, relationships and semantics within a given domain of discourse about which there is a need to represent some information. Modifiers that qualify or further specialize the object class may be used. An object class is one of three data concepts used to characterize a data element."

For NTCIP purposes, all table and table entry object specifications shall be declared as "Entity Type".

2.4.3.2 Data Element

As defined in ISO 14817 Subclause 4.7, a Data Element is some single unit of information of interest (such as a fact, proposition, observation, etc.) about some (entity) class of interest (e.g., a person, place, process, property, concept, association, state, event). A data element is considered to be indivisible in a particular context.

ISO 14817 Subclause 7.10 further states, " A data element shall be a formalized representation of some information (i.e., a property; e.g., a fact, proposition, or an observation) about an object class (e.g. a person, place, process, concept, association, state, or event), with an explicit value domain. A data element (a data concept) shall be characterized by three data concepts [...]): object class, property, and an explicit value domain descriptive name (and value domain reference, where applicable, describing the physical form of the information).

For NTCIP purposes, all object specifications other than table, table entry, or a data frame shall be declared as "Data Element".

2.4.3.3 Data Frame

As defined in ISO 14817 Subclause 4.9, a Data Frame is a grouping of data elements primarily for the purpose of referring to the group with a single name, and thereby efficiently reusing groups of data elements that commonly appear together (as an ASN.1 SEQUENCE, SEQUENCE OF, SET, SET OF or CHOICE) in a message specification

ISO 14817 Subclause 7.4 further states, "A data frame shall be a structured grouping of data elements primarily for the purpose of referring to a group with a single name to efficiently reuse such groups of data elements that commonly appear together in a usage specification.

For NTCIP purposes, all block object specifications shall be declared as "Data Frame".

ISO 14817 does not define an appropriate value-domain-term for "Data Frame". For NTCIP purposes, the value "frame" or "frm" shall be used.

2.4.4 ASN.1 Name

The value of the ASN.1 NAME subfield is delimited by <ASN.1Name> and the value shall be as defined in ISO 14817 Subclause B.2.6 and ISO 14817 Annex A. The ASN.1 Name shall be the name of a data concept expressed as a valid “typereference”.

NOTE—At this time, there is no requirement to add an <ASN.1 Name> subfield to the DESCRIPTION field. Additional information about the ASN.1 Name may be found in NTCIP 8005.

2.4.5 Data Type

The value of the DATA TYPE subfield is delimited by <DataType> and the value shall be as defined in ISO 14817 Subclause B.5.1. The Data Type must be a valid instance of a specific subset of ASN.1 types, but constraints and indirect references to these types are allowed.

NOTE—At this time, there is no requirement to add an <DataType> subfield to the DESCRIPTION field. Additional information about Data Type may be found in NTCIP 8005.

2.4.6 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.

2.4.7 Unit of Measure

The value of the Unit of Measure subfield is delimited by <Unit> and the value shall be as defined in ISO 14817 Subclause B.5.3.

2.4.8 Bit Map Format

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

Bit Number	Name		Description
Bit <x>:	<bitName>	=	<description>
	…		
Bit 1:	<bitName>	=	<description>
Bit 0:	<bitName>	=	<description>

2.4.9 Related Data Concept

Support for this meta-attribute subfield is not required at this time.

2.4.10 Relationship Type

Support for this meta-attribute subfield is not required at this time.

2.4.11 MIB Module Meta-Attributes

The definition of the "NTCIP Defined" MIB Module meta-attributes can be found in NTCIP 8005. The definition of "ISO Defined" MIB Module meta-attributes can be found in ISO 14817. These meta-attributes are the same for each data concept in an NTCIP 1200 Series Document MIB and, therefore, are defined on a MIB Module basis. Furthermore, these are not defined by the NTCIP Working Group but by the NTCIP Data Steward when the MIB Module is created or when the MIB Module is imported into a data registry. The exact definition of each MIB Module meta-attribute can be found in the appropriate standard and are only referenced here to show completeness.

2.4.12 Description Field Meta Attribute Encapsulation

Meta attribute subfields are added to the DESCRIPTION clause 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 below:

```
DescrPart ::= "DESCRIPTION"   "<Definition>"         value (description DisplayString)
                              "<DescriptiveName>"    value (descriptive name) | empty
                              "<DataConceptType>"    "Entity Type" | "Data Element" | "Data Frame"
                              "<Unit >"              value (unit of measure DisplayString)
```

The meta attribute subfields shall appear after the last character of the "description" but before the closing quote. These 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

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.

The following MIB meta attributes are needed for simple, block and dynamic objects:

a) Class Name
b) Classification Scheme Name
c) Classification Scheme Name Version
d) Descriptive Name Context
e) Source Standard
f) Last Change Date
g) Submitter Organization Name
h) Submitter Phone Number

2.5.1 Logic Behind the Documentation Format

By defining information according to the rules defined in RFC 1155 and RFC 1212 and those defined in ISO 8824, we have 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 2-6. 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 in order 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. See Figure 7.

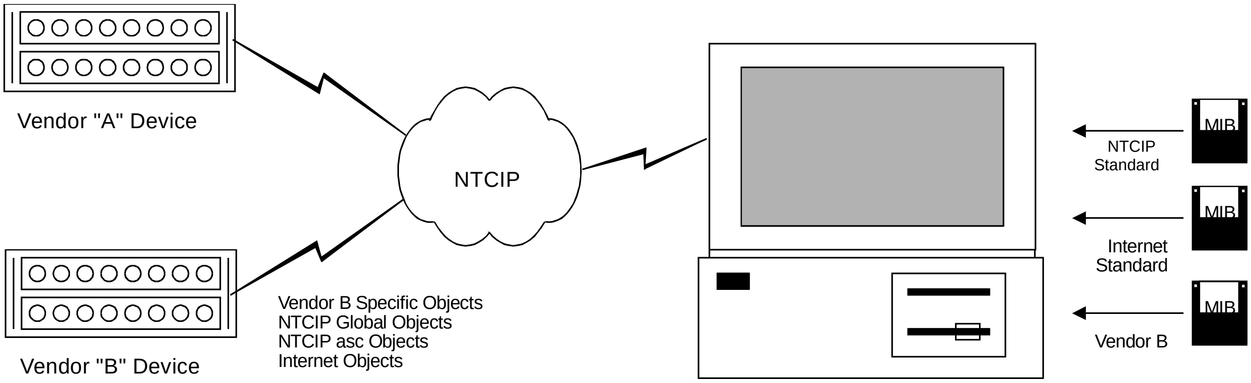


Figure 7
MIB Integration

2.5.2 Extensions to the MIB

Every NTCIP Standard MIB Revision obsoletes all previous revisions. New versions shall:

a) declare non-current object types deprecated, but shall not delete their names;
b) declare previously deprecated object types obsolete (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 object types, 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.

Section 3
NTCIP PROCEDURES

3.1 NTCIP ELEMENTS OF PROCEDURES

The IAB STD 16 – Structure and Identification of Management Information and the Definition Concise MIB Definitions does not elaborate on a number of functions related to different types of table constructs. The definition of traps was not introduced until sometime later and does not deal with the definition of enterprise-specific traps. This section defines additional requirements that must be followed when defining tables.

3.2 TABLES, ROWS, AND BLOCK OBJECTS

SNMP tables are special types of SNMP objects, which 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 has been formalized 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, it must be emphasized that, at the protocol level, relationships among columnar objects in the same row is 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 limitations of operating only on individual object instances can impose a severe penalty in cases where communications bandwidth has to be considered. In order to address this issue, the NTCIP effort 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 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.
            <DescriptiveName>ModuleTable.maxModules:quantity
            <DataConceptType>Data Element
            <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 value 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 follows:

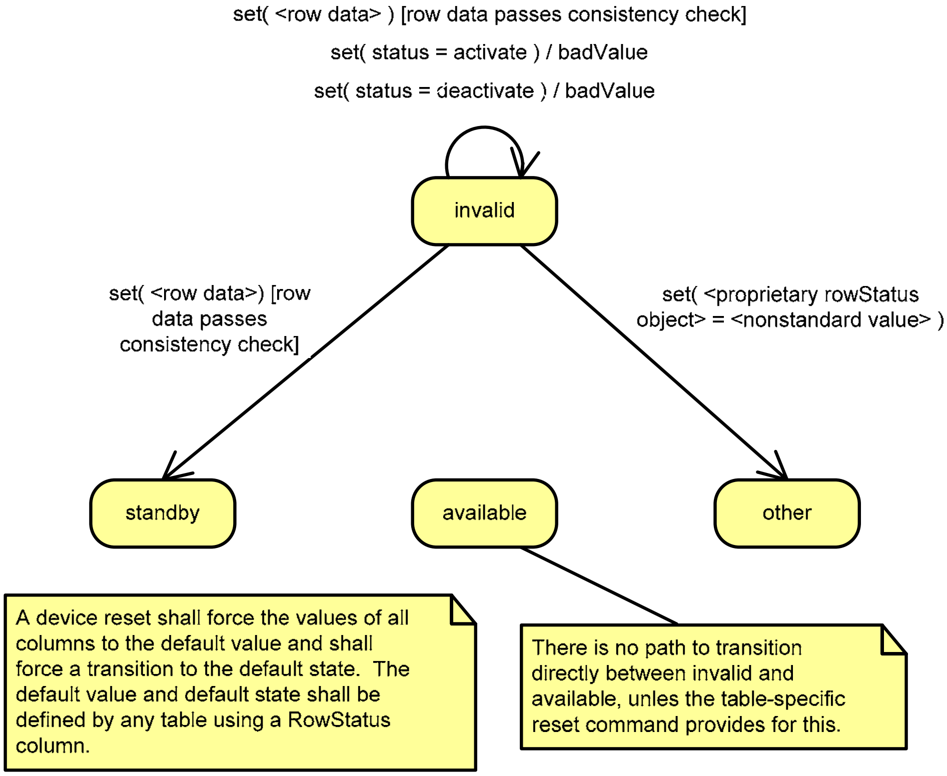
other — Status is actually controlled or defined by a user or manufacturer specific object.
standby — All the columnar data in the row have passed any consistency but is not to used by the end-application.
available — All the columnar data in the row have passed any consistency and is to be used by the end-application
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:

activate — Make the columnar data in the row available for use by the end-application.
deactivate — Make the columnar data in the row un-available for use by the end-application.

The UML state transition diagrams for RowStatusStatic are defined in Figure 8, Figure 9, and Figure 10.


set(<row data>) [row data passes consistency check]
set(status = activate) / badValue
set(status = deactivate) / badValue
invalid
set(<row data>) [row data passes consistency check]
set(<proprietary rowStatus object> = <nonstandard value>)
standby
available
other

A device reset shall force the values of all columns to the default value and shall force a transition to the default state. The default value and default state shall be defined by any table using a RowStatus column.

There is no path to transition directly between invalid and available, unles the table-specific reset command provides for this.

Figure 8
Row Status Static – Invalid

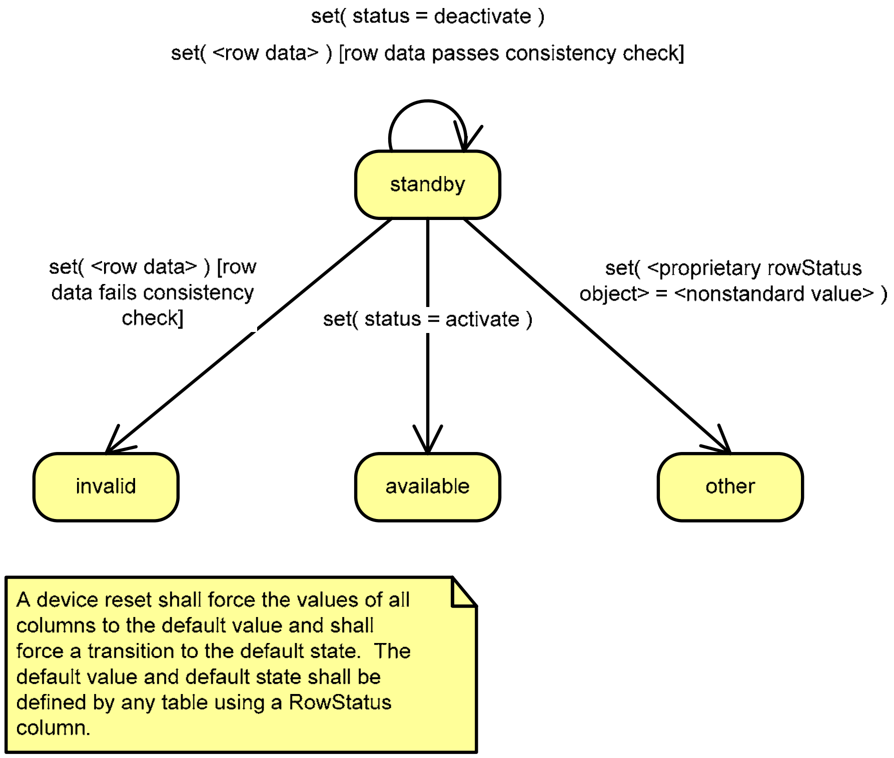
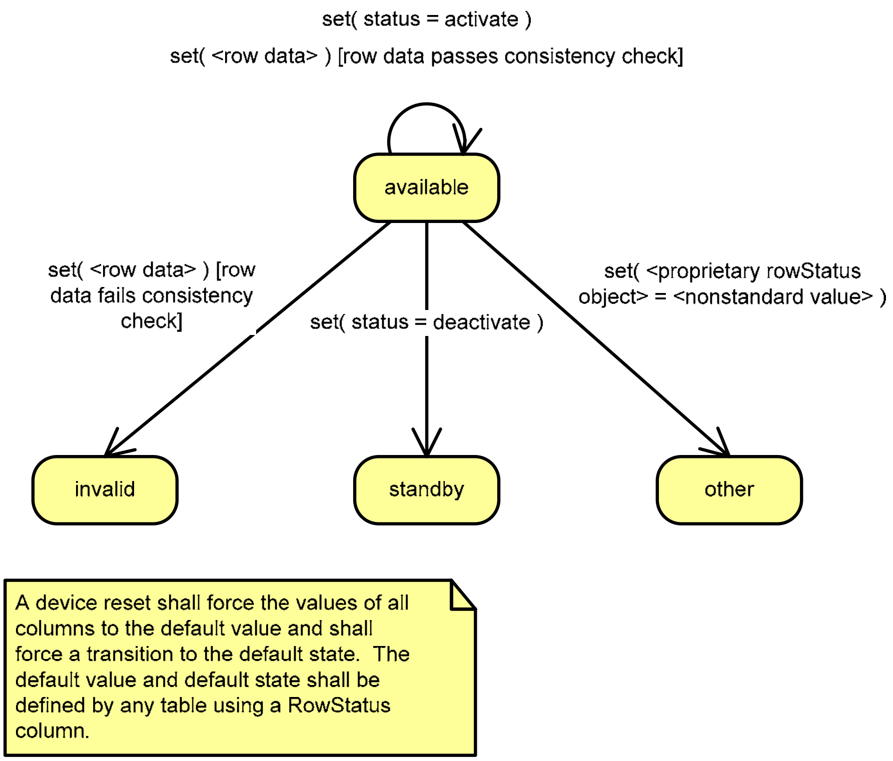

set(status = deactivate)
set(<row data>) [row data passes consistency check]
standby
set(<row data>) [row data fails consistency check]
set(status = activate)
set(<proprietary rowStatus object> = <nonstandard value>)
invalid
available
other
A device reset shall force the values of all columns to the default value and shall force a transition to the default state. The default value and default state shall be defined by any table using a RowStatus column.

Figure 9
Row Status Static – Standby


set(status = activate)
set(<row data>) [row data passes consistency check]
available
set(<row data>) [row data fails consistency check]
set(status = deactivate)
set(<proprietary rowStatus object> = <nonstandard value>)
invalid
standby
other

A device reset shall force the values of all columns to the default value and shall force a transition to the default state. The default value and default state shall be defined by any table using a RowStatus column.

Figure 10
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 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.
       ...
```

If a user wanted to define a new correction mode, this could be accomplished by something like the following proprietary object:

```
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 v01 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 the 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 Bit Map <Format> subfield in the DESCRIPTION field.

Annex A
NEMA STRUCTURE OF MANAGEMENT INFORMATION 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 the end of the section (except the annex headings) constitutes the standard NTCIP 8004 MIB.

Text preceded by a double hyphen in the MIB definitions represents normative text for NTCIP 8004 v01.

A.1 SMI HEADER

```
--****************************************************************************
-- Filename:    8004v0137.MIB
-- Source:      NTCIP 8004 v01.38
-- Description: This MIB defines 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
--
--Copyright 1996-2005 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.
--****************************************************************************

NTCIP8004 v01-2008 DEFINITIONS ::= BEGIN

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

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

-- EXPORTS EVERYTHING
```

A.2 STRUCTURE INFORMATION

```
nema                    OBJECT IDENTIFIER  ::=  { enterprises 1206 }
```

```
-- NEMA has received ID 1206 from IANA
-- NEMA starts at { iso org dod internet private enterprises 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 }
layers            OBJECT IDENTIFIER  ::=  { protocols 1 }
application       OBJECT IDENTIFIER  ::=  { layers 7 }
profiles          OBJECT IDENTIFIER  ::=  { protocols 2 }
dynObjMgmt        OBJECT IDENTIFIER  ::=  { protocols 3 }

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 -- NTCIP8004 Definitions
```

Annex B
NTCIP OBJECT TYPE MACRO EXAMPLES
(Informative)

The following examples show the form of the OBJECT TYPE Macro with the descriptive name 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.
                          <DescriptiveName> Controller.databaseID:number
                          <DataConceptType> Data Element"
::= {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.
                          <DescriptiveName> PhaseTable.maxPhases:quantity
                          <DataConceptType> Data Element"
::= {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.
          <DescriptiveName>DMS.signType:code

          <DataConceptType>Data Element"
::= { 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.
                      <DescriptiveName> Ess.ntcipNum:identifier
                      <DataConceptType> Data Element"
::= {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.
          <DescriptiveName> cctvRange.mximumPreset:quantity
          <DataConceptType> Data Element "
::= {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.
          <DescriptiveName> Rmc.commRefreshThreshold:quantity
          <DataConceptType> Data Element
          <Unit> Second"
          REFERENCE   "see Clause A.3"
::={rmcCfg 1}
```

```
inputStatus OBJECT-TYPE
        SYNTAX          OCTET STRING (SIZE(1))  -- BITMAP8
        ACCESS          read-only
        STATUS          mandatory
        DESCRIPTION     "<Definition> Input status denotes a bit mapped 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.

                        <DescriptiveName> SwitchInput.status:text
                        <DataConceptType> Data Element"
::= {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.
                        <DescriptiveName> Tss.sensorSystemReset:code
                        <DataConceptType> Data Element"
::= {tssSystemSetup 1 }
```

```
timebaseOperationalModeTable   OBJECT-TYPE
         SYNTAX   SEQUENCE OF TimebaseOperationalModeEntry
         ACCESS   not-accessible
         STATUS   optional
         DESCRIPTION “<Definition> A table containing Signal System Master Timebase
              action parameters. The number of rows in this table is equal to the
              maxTimebaseAscActions object.
             <DescriptiveName> TimebaseOperationalModeTable
             <DataConceptType> Entity Type
             <Table Type> static"
::= { plan 1 }

timebaseOperationalModeEntry OBJECT-TYPE
         SYNTAX TimebaseOperationalModeEntry
         ACCESS   not-accessible
         STATUS   mandatory
         DESCRIPTION “<Definition> Mode parameters for timebase control of the operational
             mode.
             <DescriptiveName> TimebaseOperationalModeEntry
             <DataConceptType> Entity Type"
          INDEX   { timebaseAscActionNumber }
::= { timebaseAscActionTable 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 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 are
                          can take on the value 1 through 8.
                          <DescriptiveName> Signal1-8Status8:frame
                          <DataConceptType> Data Fame"
          ::= {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.
                          <DescriptiveName> Split1PhaseData:frame
                          <DataConceptType> Data Fame"
          ::= {phase 13}
     END
```

The DESCRIPTION clause must clearly state 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                    -- @NTCIP1201-2003
        }

    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 data frame 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.
        <DescriptiveName> DatabaseGroupPhase:frame
        <DataConceptType> Data Frame"
    ::= {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
```

< This page is intentionally left blank. >

Annex D
UML INFORMATION (INFORMATIVE)

It is beyond the scope of NTCIP 8004 v01 to provide a rigorous or complete explanation of the UML Diagrams used within this standard. However, IBM provides the article “**UML basics: An introduction to the Unified Modeling Language**” at

http://www-106.ibm.com/developerworks/rational/library/769.html#N10090.

There are additional resources listed on that web page. “The Unified Modeling Language User Guide” cited in Section 1.2 may also serve to provide a more complete understanding of UML.

< This page is intentionally left blank. >

Annex E
ISO 14817 NAMING CONVENTIONS
(Informative)

E.1 NAME CONVERSIONS

Prior to the introduction of ISO 14817, the only "name" appearing in an NTCIP MIB was the object name. For the purposes of a data registry, ISO 14817 adds Descriptive Name, Fully Descriptive Name, and ASN.1 Name. The following illustrates the differences:

OrganizationIdentifier-DocumentIdentifier::ObjectClassTerm.propertyTerm:value-domain-term

Object Name:	phaseMinGreen
Descriptive Name:	Phase.minGreen:qty
Fully Qualified Descriptive Name:	NTCIP-1202::Phase.minGreen:qty
ASN.1 Name:	Phase-minGreen

The following are the general rules used to convert an Object Name to a Descriptive Name:

a) The ObjectClassTerm is usually a person, place, or thing of interest and is something that you can apply a verb to. For columnar objects, the ObjectClassTerm is usually name of the table less "table" For example the ObjectClassTerm for entries in the TimeBaseScheduleTable would be "TimeBaseSchedule". For scaler or non-columnar objects, the ObjectClassTerm is usually the name of entity to which the propertyTerm applies. For example, globalSetIDParameter can be thought of as applying to a "Controller".
b) The propertyTerm is usually the objectName. However, if the objectName includes the ObjectClassTerm, the ObjectClassTerm is typically dropped. For example, in the TimeBaseScheduleTable, timeBaseScheduleNumber would become "number".
c) The value-domain-term is one of terms defined in Section 2.4.2.

To convert a Descriptive Name to a Fully Qualified Descriptive Name, add the NTCIP-XXXX (Standard Number) and a "::".

To convert a Descriptive Name to an ASN.1 Name, replace the "." with a "-" and drop the property term.

§