A User Comment Draft of the Joint Committee on NTCIP

NTCIP 1201:200X v03.03

National Transportation Communications for ITS Protocol

Global Object (GO) Definitions – version v03

v03.03b November 2, 2007

A major revision of NTCIP 1201 v02.32

Published by

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

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

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

NOTICES

Copyright Notice

PDF File License Agreement

Data Dictionary and MIB Distribution Permission

(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 MIB is restricted in that the syntax 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;
(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 / 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 you.

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

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

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/or 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 document are the trademarks of their respective owners.

ACKNOWLEDGEMENTS

This publication was prepared by the NTCIP Base Standards and Profiles 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 this document was prepared, the following individuals were members of the NTCIP BSP2 WG:

- Bob De Roche (Chair) [CROSSCHECK AND UPDATE PER ROSTER for version 03]
- Bud Kent
- Joerg "Nu" Rosenbohm
- Alex Mousadi
- Peter Ragsdale
- Bob Rausch
- Ralph Boaz

Other individuals providing input to the document, include:

- Moon Chueng [EDIT PER ROSTER AND UC RECORDS for version 03
- Blake Christie
- Mike Forbis
- Craig Gardner
- Paul Hing
- Larry Lindley
- Robert Martindale
- Gary Meredith
- Jeris White

In addition to the many volunteer efforts, recognition is also given to those organizations which supported the efforts of the working group by providing comments and funding for the document, including:

- U.S. Department of Transportation, Research and Innovative Technology Administration
- [EDIT PER USER COMMENT RECORDS for version 03]
- Econolite Control Products, Inc.
- G. C. Herrick & Associates, Inc.
- Iteris, Inc.
- Ministry of Ontario
- Noblis (formerly Mitretek)
- Peek Transportation Systems
- Pillar Consulting
- Quixote Traffic Corp.
- Robert De Roche Consulting, Inc.
- Siemens ITS
- Telvent Farradyne Inc. (formerly PB Farradyne)
- Transcore
- Trevilon
- URS
- Washington State DOT

FOREWORD

This document uses only metric units.

The purpose of this publication is to identify and define the common object definitions that may be supported by devices that are NTCIP-compliant.

This document is an NTCIP Data Dictionary Standard. Data Dictionary Standards provide definitions of data elements for use within NTCIP systems; they are approved by AASHTO, ITE, and NEMA, after recommendation by the NTCIP Joint Committee.

For more information about NTCIP standards, visit the NTCIP Web Site at http://www.ntcip.org. For a hardcopy summary of NTCIP information, contact the NTCIP coordinator at the address below.

In preparation of this NTCIP document, input of users and other interested parties was sought and evaluated. Inquires, comments, and proposed or recommended revisions should be submitted to:

NTCIP Coordinator
National Electrical Manufacturers Association
1300 North 17th Street, Suite 1752
Rosslyn, VA 22209-3801
fax: (703) 841-3331
e-mail: ntcip@nema.org

Approvals

TBD This document 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 – Standard Specification; ***TBD***
ITE – Software Standard; ***TBD***
NEMA – Standard; ***TBD***

History

From 1996 to 1999, this document was referenced as NEMA TS 3.4. However, to provide an organized numbering scheme for the NTCIP documents, this document is now referenced as NTCIP 1201. The technical specifications of NTCIP 1201 are identical to the former reference, except as noted in the development history below:

NEMA TS 3.4-1996 v96.01.7, April 7, 1997. October 1996 – Version 1.5 approved by NEMA. April 1997 – Version 1.7 published by NEMA with editorial corrections. October 1996 – Accepted as a Recommended Standard by the Joint Committee on the NTCIP. Approved by AASHTO in 1997 and approved by ITE in December 1997.

NEMA TS 3.4 Amendment 1 v98.01.07. October 1998 – Version 98.01.05 accepted as a Recommended Amendment by the Joint Committee on the NTCIP, and edited v01.07 referred for balloting and approval by NTCIP Standards Bulletin B0032 in May 1999. Approved by AASHTO in October 1999, approved by ITE in January 2001, and approved by NEMA in December 1999. Amendment 1 clarified ambiguities discovered during real-world implementations of this standard.

NTCIP 1201:1996 [assigned version 01.08]. August 1999 – Assigned NTCIP 1201 document number in NTCIP Standards Bulletin B0038. August 2000 – Joint NTCIP Standards Publication cover used over TS 3.4 contents.

NTCIP 1201:1996 v01.10, December 2001. January 2002 – Formatted for printing: incorporated Amendment 1 v07 into text; updated title page date and version number; modified and reorganized front matter to conform to NTCIP 8002. Most references to TS 3 standard designations were changed to equivalent NTCIP standard numbers.

NTCIP 1201 v02. December 2002. Developed to reflect additional lessons learned, to incorporate better documentation (in the Annex) of some of the logic required to implement the standards, and to add new features requested by the ITS community.

NTCIP 1201 v02.14. September 2001 – Accepted by the NTCIP Joint Committee as a User Comment Draft. February 2002 – NTCIP Standards Bulletin B0071 distributed NTCIP 1201 v02.16 for review and comment.

NTCIP 1201 v02.24. October 2002 – Accepted by the NTCIP Joint Committee as a Recommended Standard. April 2004 – NTCIP Standards Bulletin B0092 referred NTCIP 1201 v02.26 for balloting. Approved by AASHTO in October 2004, approved by ITE in March 2005, and approved by NEMA in November 2004.

NTCIP 1201 v02.31. February 2005 – Disposed of ballot period comments on backward compatibility, object deprecation, and others. In clause 1.3 Terms, added Deprecated and Obsolete definitions.

NTCIP 1201:2005 v02.32. October 2005 – Edited document for publication with modified and reorganized front matter.

[EDIT AMEND 2 NOTES TO REFLECT NOT ISSUED] NTCIP 1201 v02.41+ Amendment 2 v09, October-November 2006. Modified globalDaylightSaving with additional information regarding 'other' and added a set of begin and end day saving time (DST) objects to reflect changes enacted by U.S. Congress (Energy Policy Act of 2005) to take effect in 2007. Revised names of AuxIO objects that appeared in v02.31 and added the AuxIO object that appeared in NTCIP 1203 but are now listed as deprecated in 1203. Moved the object definitions previously defined under the globalReport node to NTCIP 1103.
Amendment 2 v09d. Word >> "Compare and Merge Documents" used to create redline version.
Amendment 2 v10. Included these additional changes:

- In NTCIP 1201v02.32, the ACCESS of auxIOPort Direction and auxIO2PortDirection should have been listed as read-only as this was incorrect in NTCIP 1203v01.15.
- In NTCIP 1201v02.32, it was agreed the SYNTAX of auxIO2PortDescription should be changed from SIZE (0..50) to (0..255).
- In NTCIP 1201v02.32, it was agreed the SYNTAX of auxIO2PortResolution should be changed from SIZE (1..255) to (1..32).
- In NTCIP 1201v02.32, the maxAuxIO2TableNumDigitalPorts and maxAuxIO2TableNumDigitalPorts should not have been changed for 0..255 to 1..255 as this was what was shown in NTCIP 1203v01.15.
- In NTCIP 1201v02.32, the node registration of the new AuxIO objects shown in the ISO figure should have been {global 7}.

NTCIP 1201 v03.03. November 2007 – The Joint Committee withdrew acceptance of Amendment 2 v10 due to ballot comments on the Daylight Saving Time mechanisms. The WG proposed major version v03 with updated DST mechanisms, and with additional changes, and for easier version tracking. Major version v03 includes:

- Revised the DST mechanism by replacing the previously proposed DST objects with new objects.
- Added the SNMP interface dialogs from NTCIP 1203v02 to become the generic standard for all NTCIP device standards.
- Deferred the addition of externally-developed test procedures to a future major version.

Compatibility of Versions

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 this document 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 this document should also consult NTCIP 8004 for specific guidelines on compatibility.

INTRODUCTION

This publication provides definitions of data elements for use with various transportation devices. The data is defined using the Simple Network Management Protocol (SNMP) object-type format as defined in RFC 1212 and the defined NTCIP format defined in NTCIP 8004. This data would typically be exchanged using one of the NTCIP 1103 recognized Application Layers (e.g., SNMP).

This standard defines requirements that are applicable to all NTCIP environments and it also contains optional and conditional clauses that are applicable to specific environments for which they are intended.

The following keywords apply to this document: AASHTO, ITE, NEMA, NTCIP, global, data, data dictionary, object.

In 1992, the NEMA 3-TS Transportation Management Systems and Associated Control Devices Section began the effort to develop the NTCIP. 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 ITS network.

In September 1996, an agreement was executed among AASHTO, ITE, and NEMA to jointly develop, approve, and maintain the NTCIP standards. In late 1998, the Global Object Working Group was tasked with updating the Global Object Definitions standard. The first meeting of the GO WG was held in January 1999. [INSERT HISTORY OF BS, PROFILES, BSP, and BSP2 WG MERGERS AND YEARS]

CONTENTS

Page

FOREWORD iii

Section 1 GENERAL 1

1.1 Scope 1

1.2 References 1

National Electrical Manufacturers Association 1

1.2.1 Normative References 1

1.2.2 Other References 2

1.3 Terms 3

1.4 Abbreviations 3

1.5 Object Tree 4

1.6 Controller Class Diagram 6

Section 2 OBJECT DEFINITIONS 7

2.1 NTCIP Objects 7

2.2 Global Configuration Node 9

2.2.1 Global Set ID Parameter 9

2.2.2 Maximum Modules Parameter 10

2.2.3 Module Table 10

2.2.3.1 Module Number Parameter 10

2.2.3.2 Module Device Node Parameter 11

2.2.3.3 Module Make Parameter 11

2.2.3.4 Module Model Parameter 11

2.2.3.5 Module Version Parameter 11

2.2.3.6 Module Type Parameter 12

2.2.4 Base Standards Parameter 12

2.3 Global Database Management Node 12

2.3.1 Database Creation Transaction 13

2.3.2 Database Error Type Parameter 16

2.3.3 Database Error ID Parameter 16

2.3.4 Database Transaction ID Parameter 16

2.3.5 Database Make ID Parameter 17

2.3.6 Database Verify Status Parameter 17

2.3.7 Database Verify Error Parameter 18

2.4 Global Time Management Node 18

2.4.1 Global Time Parameter 18

2.4.2 Global Daylight Savings Parameter 18

2.4.3 TimeBase Event Scheduler Node 20

2.4.3.1 Maximum Number of Time Base Schedule Entries Parameter 20

2.4.3.2 Time Base Schedule Table 20

2.4.4 Day Plan Parameters 23
2.4.4.1 Maximum Number of Day Plans- Parameter 23
2.4.4.2 Maximum Number of Day Plan Events - Parameter 23
2.4.4.3 Day Plan Table 23
2.4.4.4 Day Plan Status Parameter 26
2.4.4.5 Schedule Status Parameter 26
2.4.5 Global Local Time Differential Parameter 26
2.4.6 Standard Time Zone Parameter 27
2.4.7 Local Time Parameter 27
2.4.8 Daylight Saving Time Node 27
2.4.8.1 Maximum Daylight Saving Time Table Entries Parameter 27
2.4.8.2 Daylight Saving Time Table Parameter 28
2.5 Report Parameter Node 34
2.6 STMP Object Node 34
2.7 PMPP Object Node 34
2.7.1 Maximum HDLC Group Address Parameter 34
2.7.2 HDLC Group Address Table 35
2.7.2.1 HDLC Group Address Index Parameter 35
2.7.2.2 HDLC Group Address Parameter 35
2.7.2.3 HDLC Group Address Number Parameter 36
2.8 Security Node 36
2.9 New Auxiliary I/O Objects in this standard 36
2.9.1 Maximum Number of Digital Auxiliary IOs Parameter 38
2.9.2 Maximum Number of Analog Auxiliary IOs Parameter 38
2.9.3 Auxiliary IO Table Parameter 38
2.9.3.1 Auxiliary Port Type Parameter 39
2.9.3.2 Auxiliary Port Number Parameter 39
2.9.3.3 Auxiliary Description Parameter 40
2.9.3.4 Auxiliary Resolution Parameter 40
2.9.3.5 Auxiliary Value Parameter 40
2.9.3.6 Auxiliary Port Direction Parameter 41
2.9.3.7 Auxiliary Port Last Commanded State Parameter 41
2.10 Old Auxiliary I/O Objects from NTCIP 1203:1997 42
2.10.1 Maximum Number of Digital Auxiliary IOs Parameter 44
2.10.2 Maximum Number of Analog Auxiliary IOs Parameter 44
2.10.3 Auxiliary IO Table Parameter 44
2.10.3.1 Auxiliary Port Type Parameter 45
2.10.3.2 Auxiliary Port Number Parameter 45
2.10.3.3 Auxiliary Description Parameter 45
2.10.3.4 Auxiliary Resolution Parameter 45
2.10.3.5 Auxiliary Value Parameter 46

2.10.3.6 Auxiliary Port Direction Parameter 46
Section 3 Conformance 47
Annex A Concept of Operations (Normative) 48
A.1 Download Transaction Use Case 48
A.2 Set Time 50
A.3 Configure Scheduler 53
Annex B Test Procedures (Normative) 54
Annex C Class Diagrams (Informative) 55
C.1 Configuration Information 55
C.2 Transaction Information 55
C.3 Time Information 56
C.4 Auxiliary Input/Output Information 57
Annex D Summary of Changes (Informative) 59
D.1 Updated ISO Tree 59
D.2 Updates to Conform with NTCIP 8004 59
D.3 Updated Name of the MIB 59
D.4 Added Default Value Statements 60
D.6 Enhanced Module Version Definition 60
D.7 Added an Object to Identify Supported Standards 60
D.8 Corrected the Database Transaction Feature 60
D.9 Added Support for Additional Daylight Saving Modes 60
D.10 Added new Objects to Address US Daylight Saving Time Modifications 61
D.11 Added a Schedule Status Object 61
D.12 Clarified Definitions of Day Plan Objects 61
D.13 Corrected Problems with the Local Time Logic 61
D.14 Clarified Definitions Related to the Event Log 61
D.15 Reordered Clauses for the Event Log 62
D.16 Added Support for Another Mode to the Event Log 62
D.17 Added Error Value to the Event Configuration Status 62
D.18 Corrected Syntax of Event Log Size Object 62
D.19 Replaced the Group Address Object 62
D.20 Added Generic Auxiliary I/O Objects 63
D.21 Removed Conformance Statements 63
D.22 Added a Concept of Operations 63
D.23 Prepared Communication Objects to be Moved to 1103 63
D.24 Deleted Annex B to Document Deprecated Objects 63
D.25 Added Class Diagrams 64
D.26 Added Generic SNMP Interface Definitions 64
Annex E SNMP INTERFACE (Normative) 65
E.1 Generic SNMP Get Interface 65
E.2 Generic SNMP Get-Next Interface 66
E.3 Generic SNMP Set Interface 66

E.4 Variable Binding List Structure 67
E.5 Additional Requirements 67
E.5.1 Grouping of Objects in a Request 67
E.5.2 Support of Get 67
E.5.3 Support of GetNext 67
E.5.4 Support of Set 67
E.5.5 Performance 67

Section 1
GENERAL

1.1 SCOPE

The messaging between Transportation Management Center and field devices is accomplished by using the NTCIP Application Layer services to convey requests to access or modify values stored in a given device; these values are referred to as objects. The purpose of this publication is to identify and define these objects definitions that may be supported by multiple device types (e.g. actuated signal controllers and variable message signs). The grouping of objects for a given device type is performed in the device-type-specific object definition standard.

1.2 REFERENCES

For approved revisions, contact:

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

For draft revisions, which are under discussion by the relevant NTCIP Working Group, and recommended revisions of the NTCIP Joint Committee, visit the world wide web at http://www.ntcip.org.

The following standards (normative references) contain provisions which, through reference in this text, constitute provisions of this Standard. Other documents and standards (other references) are referenced in these documents, which 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 below.

1.2.1 Normative References

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

NTCIP 1103 v02.12 (v02.12 was a BSP2 WG internal draft as of October 2007; v02.10b was a UCD as of December 2006)	*Transportation Management Protocols – version 02*
NTCIP 8004 v2.07a (v02.10 was a proposed RS as of October 2007; v02.07a was a UCD as of January 2007)	*Structure and Identification of Management Information – version 02*

ANSI
11 West 42nd Street, 13th Floor
New York, New York 10036

ISO/IEC 8824-1:1998 *Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation*

1.2.2 Other References

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

NEMA TS 2-1998 *Traffic Controller Assemblies with NTCIP Requirements*

NTCIP 1102 *Octet Encoding Rules*

NTCIP 1104 *Naming Conventions*

NTCIP 9001 *NTCIP Guide*

ANSI
11 West 42nd Street, 13th Floor
New York, New York 10036
(212) 642-4900

ISO/IEC 8825-1:1998 *Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).*

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

IAB STD 15	RFC 1157	*A Simple Network Management Protocol (SNMP)*. M. Schoffstall; M. Feder; J. Davin; J. Case; 05/10/1990
IAB STD 16	RFC 1155	*Structure and Identification of Management Information for TCP/IP-based Internets*. K. McCloghrie; M. Rose; 05/10/1990
	RFC 1212	*Concise MIB Definitions*. K. McCloghrie; M. Rose; 03/26/1991
IAB STD 17	RFC 1213	*Management Information Base for Network Management of TCP/IP-based Internets:* MIB-II. K. McCloghrie; M. Rose; CP/IP-base

1.3 TERMS

For the purposes of this standard, the following terms, definitions, acronyms, and abbreviations apply. Terms not defined in this clause are in accordance with their definitions in the NTCIP 8004. Electrical and electronic terms not defined in this clause are used in accordance with their definitions in IEEE Std 100-2000. English words not defined in this clause or in IEEE Std 100-2000 are used in accordance with their definitions in *Webster's New Collegiate Dictionary*.

Class: An abstraction of any kind of object that may be described; equivalent to a ISO 14817 Object Class.

Component: A central system, field device, etc. that supports NTCIP.

Conformance level: Each of the defined Profiles have one or more layers specifying the protocols that must be implemented in a device to correspond to a particular level of NTCIP support.

Data Value: The value a data element

Database Object: Any object identified as a 'database object' by the relevant device-specific standard. For example, in NTCIP 1202 (version 02), objects that are identified as 'P' or 'P2' are database objects.

Deprecated: The 'deprecated' value in the STATUS field of an 'OBJECT-TYPE' macro (see Section 5) indicates that the subject object was included in a previous version of the standard but no longer represents the preferred design. An implementer implementing this version of the standard is not required to implement a deprecated object, but may wish to support it to foster interoperability with older implementations. The STATUS of a deprecated object will likely change to obsolete in some future version of the standard.

Feature: A capability of an component

Obsolete: The 'obsolete' value in the STATUS field of an 'OBJECT-TYPE' macro (see Section 5) indicates that the subject object was included in a previous version of the standard but no longer in significant use within the industry. While an implementer is allowed to implement an obsolete object, the benefits of doing so may be minimal due to the limited use of the object.

Profile: Refers to a set of protocols, each of which operates independently on one of the seven (7) OSI Layers, if this layer is utilized. Different protocols are utilized at the same layer within different profiles.

Static Database Object: A parameter that does not change other than by a user command.
For example, the controllerTimeZone object is a static database object since it only changes value through some sort of user command, however, globalTime is not a static database object since it is constantly incrementing.

1.4 ABBREVIATIONS

The abbreviations used in this Standard Publication are defined as follows:

ASC—Actuated Signal Controller

CRC—Cyclic Redundancy Check; polynominal algorithm performed on a specified range of data resulting in a 16 or 32 bit value.

MIB—Management Information Base; a collection of objects defined using Abstract Syntax Notation One (ASN.1) that can be accessed via a network management protocol.

NVT-ASCII—Network Virtual Terminal - American Standard Code for Information Interchange as defined in RFC 854

PMPP—Point-to-MultiPoint Protocol, a transporation specific subnetwork layer protocol that enables communication between multiple devices on the same communications line/channel.

STMP—Simple Transportation Management Protocol, part of the Transportation Management Protocols of the NTCIP effort (see NTCIP 1103). STMP provides a simple and bandwidth efficient mechanism to communicate with field devices.

1.5 OBJECT TREE

The following figure provides an overview of the organization of the data defined by this document.

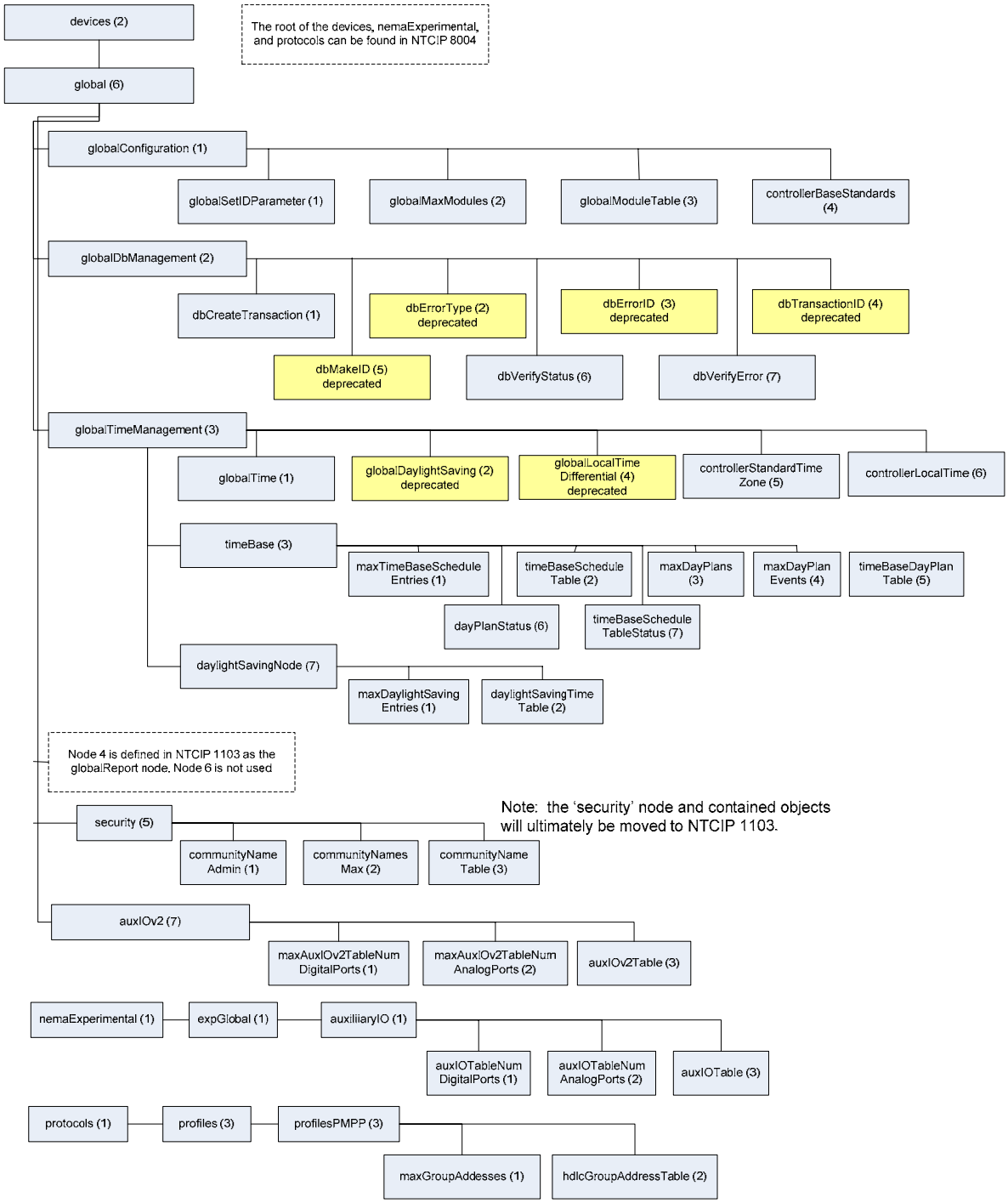

Figure 1: ISO Tree Structure

1.6 CONTROLLER CLASS DIAGRAM

The following figure depicts the components of data stored within a controller in Unified Modeling Language (UML) notation.

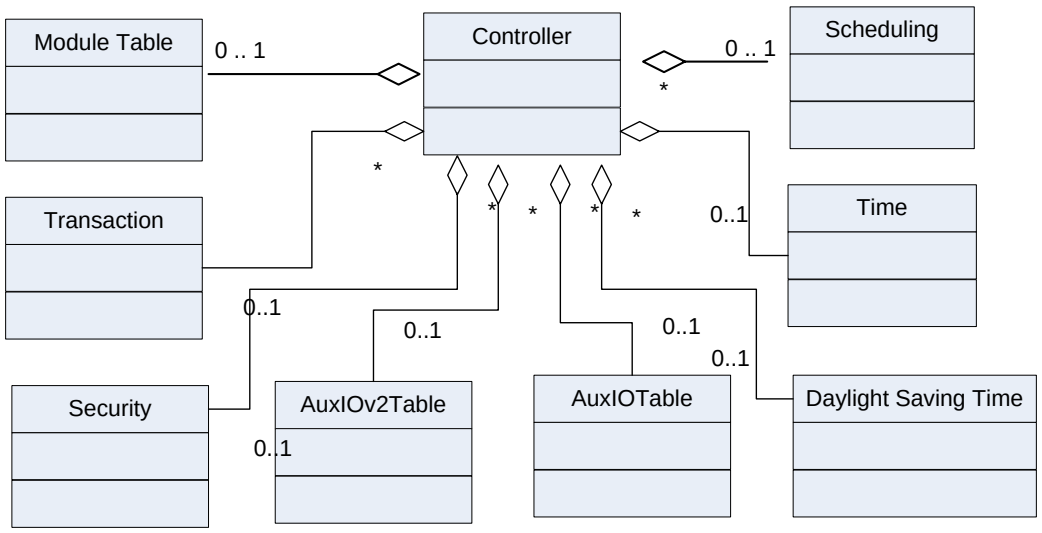

Figure 2: Controller Class Diagram

The diagram indicates that a Controller may or may not contain the following major components:

- A module table
- A transaction service
- A definition of time
- A definition of daylight saving time
- A scheduling service
- A definition of security objects
- An auxiliary input/output tables (version 01 and version 02 tables)

The details of each class are defined through the Management Information Base provided in Section 2. More detailed class diagrams for each feature are provided in Annex C.

Note that there is another group that is currently defined in this standard. This group contains the object definitions defined under the PMPP Object Node. However, this node and all its object definitions will ultimately be moved to NTCIP 2101. Therefore, this information is not included in the diagram above.

Section 2
OBJECT DEFINITIONS

This section defines those objects which are expected to be used by different device types such as actuated signal controllers, variable message signs, ramp meter controllers. The objects are defined in OJBECT-TYPE macro defined in RFC 1212 per the rules defined in NTCIP 8004. The text provided from Clause 2.1 through the end of the section (except the clause headings) constitutes the standard NTCIP1201-2004 MIB.

In order to convert these object definitions into data concepts, e.g. for the exchange in center-to-center communications, the rules defined in NTCIP 8005 shall apply.

All of the objects defined in this document reside under the "global" node of the global naming tree. To aid in object management, the "global" node has been subdivided into logical categories, each defined by a node under the "global" node. The individual objects are then located under the appropriate node.

Nodes should not be confused with conformance requirements, which are defined in device specific object standards. Conformance requirements are based on logical groupings of objects that provide specific features that may be desired in a device. While the conformance requirements will frequently correspond to the nodal structure, a conformance group may contain objects that are not lexigraphically ordered. For example, a schedule conformance group may contain both "global" and "asc" specific objects.

NOTE – This version of the standard uses the new NTCIP 8004 conventions. It specifies all (non-deprecated/non-obsolete) objects to have a STATUS of "mandatory" according to the conventions stated in NTCIP 8004; it is the responsibility of any document referring to this standard to specify exactly which objects should be supported under what conditions through a Protocol Requirements List (PRL). Documents referring to the 1997 version of this standard shall use the STATUS settings as defined in the published (and amended) 1997 version.

Text preceded by a double hyphen in the MIB definitions represent normative text for this standard. The class diagrams contained within this standard are supplemental normative requirements to the text.

[EDIT NOTE – REVISE FOOTER TO Copy Per MIB Distribution Notice]

2.1 NTCIP OBJECTS

```
-- ************************************************************************
-- Filename:    1201v0302 Main.mib
-- Description: This MIB defines various globally applicable objects
-- Source:      NTCIP 1201v03.02
-- MIB Revision History:
-- 09/27/04   Created file form Word .Doc file.
--            Changed name of file
--            Removed global report, logicalNameTranslation and communityName
--            nodes per NTCIP 1103v0124
--            Removed state transition diagram form 2.3.1
--            Changed "FROM NTCIP8004-A" to "FROM NTCIP8004-A-2004"
--            Added space before INTEGER in  timeBaseScheduleNumber INTEGER,
--            Changes status of moduleNumber, timeBaseScheduleNumber,
--            dayPlanNumber, dayPlanEventNumber, eventClassNumber,
--            eventConfigID  and thier associated Entry to mandatory to
--            eliminate incompatible status errors
```

```
--            Checked with  -CJ switch to eliminate gratuitous default value
--            warnings on globalTime and eventClassLimit.
--            Restructured IMPORTS around NTCIP8004-A-2004
-- 11/20/06   Added new beginning and end of DST objects. Revised Source
--            information
--            Added header and created standalone MIB without AuxIO objects
-- 07/02/07   Revised new beginning and end of DST objects.
-- 09/18/07   Further revisions of DST objects and updates to this Header.
--
--Copyright 2007 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.
--*****************************************************************************

--NTCIP OBJECTS
NTCIP1201-2007 DEFINITIONS ::= BEGIN

-- NTCIP 8004 Header
--

--For the purpose of this section, the following OBJECT IDENTIFIERS are used:
IMPORTS
   OBJECT-TYPE
   FROM RFC-1212

   DisplayString
   FROM RFC1213-MIB
   devices, protocols, profiles, global
   FROM NTCIP8004-A-2004
   Opaque, Counter, Gauge, null
   FROM RFC1155-SMI;

-- global OBJECT IDENTIFIER ::= { devices 6 } 1.3.6.1.4.1.1206.4.2.6
```

2.2 GLOBAL CONFIGURATION NODE

```
globalConfiguration OBJECT IDENTIFIER ::= { global 1 }
-- This node is an identifier used to group all objects for support of
-- configuration functions that are common to most device types.
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1
```

2.2.1 Global Set ID Parameter

```
globalSetIDParameter OBJECT-TYPE
   SYNTAX   INTEGER (0..65535)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>  Specifies a relatively unique ID (e.g., this could be a
      counter, a check-sum, etc.) for all user-changeable parameters of the
      particular device-type currently implemented in the device.  Often this
      ID is calculated using a CRC algorithm.
      <Informative>  This value should be calculated when a change of any
      static database object has occurred.  The value reported by this object
      should not change unless there has been a change in the static data
      since the last request.  If the actual objects that are to be included
      to create this object value are not defined in the actual device-level
      standard such as 1202 or 1203, than the general guidance is to include
      all configuration objects that stored in a type of memory that survives
      power outages.
      A management station can use this object to detect any change in the
      static database objects by monitoring this value after it has
      established a baseline.

      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.1"
   ::= { globalConfiguration 1}
```

2.2.2 Maximum Modules Parameter

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

2.2.3 Module Table

```
globalModuleTable   OBJECT-TYPE
   SYNTAX   SEQUENCE OF ModuleTableEntry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition>A table containing information regarding manufacturer
      of software and hardware and the associated module models and
      version numbers as well as an indicator if the module is hardware
      or software related. The number of rows in this table shall equal
      the value of the globalMaxModules object.
      <TableType> static
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3"
   ::= { globalConfiguration 3 }

moduleTableEntry OBJECT-TYPE
   SYNTAX   ModuleTableEntry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object defines an entry in the module table.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.1"
   INDEX { moduleNumber }
   ::= { globalModuleTable 1 }


ModuleTableEntry ::= SEQUENCE {
   moduleNumber         INTEGER,
   moduleDeviceNode     OBJECT IDENTIFIER,
   moduleMake           OCTET STRING,
   moduleModel          OCTET STRING,
   moduleVersion        OCTET STRING,
   moduleType           INTEGER }
```

2.2.3.1 Module Number Parameter

```
moduleNumber OBJECT-TYPE
   SYNTAX   INTEGER (1..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object contains the row number (1..255) within
      this table for the associated module.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.1.1"
   ::= { moduleTableEntry 1 }
```

2.2.3.2 Module Device Node Parameter

```
moduleDeviceNode   OBJECT-TYPE
   SYNTAX   OBJECT IDENTIFIER
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object contains the device node number of the
      device-type, e.g., an ASC signal controller would have an OID of
      1.3.6.1.4.1.1206.4.2.1.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.2"
   ::= { moduleTableEntry 2 }
```

2.2.3.3 Module Make Parameter

```
moduleMake   OBJECT-TYPE
   SYNTAX   OCTET STRING
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object specifies the manufacturer of the
      associated module.  A null-string shall be transmitted if this
      object has no entry.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.3"
   ::= { moduleTableEntry 3 }
```

2.2.3.4 Module Model Parameter

```
moduleModel  OBJECT-TYPE
   SYNTAX   OCTET STRING
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object specifies the model number (hardware) or
      firmware reference (software) of the associated module.  A null-
      string shall be transmitted if this object has no entry.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.4"
   ::= { moduleTableEntry 4 }
```

2.2.3.5 Module Version Parameter

```
moduleVersion OBJECT-TYPE
   SYNTAX   OCTET STRING
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object specifies the version of the associated
      module.  If the moduleType has a value of software, the value of
      this object shall include the date on which the software was
      released as a string in the form of YYYYMMDD, it shall be followed
      by a space, a hyphen, another space, the lower-case letter ‘v’,
      followed by a version or configuration number.  Preceding zeros
      shall be required for the date.  For example, version 7.03.02 of
      the software released on July 5, 2002 would be presented as
      20020705 – v7.03.02
      A null-string shall be transmitted if this object has no entry.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.5"
   ::= { moduleTableEntry 5 }
```

2.2.3.6 Module Type Parameter

```
moduleType  OBJECT-TYPE
   SYNTAX   INTEGER {
                  other (1),
                  hardware (2),
                  software (3) }
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>This object specifies if the associated module is a
      hardware or software module.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.3.6"
   ::= { moduleTableEntry 6 }
```

2.2.4 Base Standards Parameter

```
controllerBaseStandards   OBJECT-TYPE
   SYNTAX   OCTET STRING (SIZE (0..256))
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>An ASCII string that shall identify all of the
      standard document numbers that define or reference MIBs upon which
      the device is based.  Where applicable, profiles shall be
      referenced rather than the base standards.  The string shall be
      constructed as follows: The acronym of the standards development
      organization (or other body) that developed and approved the
      standard; a space; the standards document number; a colon; and the
      documents version number as designated by the standards
      development organization (or other body).  Separate entries in the
      list of standards shall be separated by a carriage return (0x0d)
      and line feed (0x0a).
      In the case of NTCIP documents prior to formal approval, the
      version number shall be the version number in the form of lower
      case ‘v’ followed by the major version followed by a period
      followed by the minor revision.  In the case of approved NTCIP
      standards, the version number shall be the four digit year of
      publication followed by a space and the version string indicated
      above. In the case of amended NTCIP standards, it shall consist of
      the four digit year of publication of the published standard
      followed by the upper case letter ‘A’, followed by the amendment
      number.

      For example, a message sign may have the following value for this
      object:

            NTCIP 1201:v02.19
            NTCIP 1203:1997A1
            NTCIP 2101:2001 v01.19
            NTCIP 2103:v01.13
            NTCIP 2201:v01.14
            NTCIP 2301:2001 v01.08

      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.1.4"
   ::= { globalConfiguration 4 }
```

2.3 GLOBAL DATABASE MANAGEMENT NODE

```
globalDBManagement OBJECT IDENTIFIER ::= { global 2 }
```



```
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2

-- This node is an identifier used to group those objects used to manage a
-- transaction.
-- A transaction is a SET of one or more database parameters that have inter-
-- relationships with other database parameters, as such a SET for any one of
-- these objects must be validated against a set of consistency checks and may
-- potentially require the setting of a large number of objects
-- simultaneously.  Thus, the mode described by these objects allow for such a
-- large database download.
-- Any device standard that allows this feature shall define which objects are
-- database parameters versus status or control objects.
```

2.3.1 Database Creation Transaction

```
dbCreateTransaction   OBJECT-TYPE
SYNTAX      INTEGER {   normal (1),
                        transaction (2),
                        verify (3),
                        done (6)
                        }
ACCESS      read-write
STATUS      mandatory

DESCRIPTION
      "<Definition>This object provides transaction control for device
      configuration.  The transaction mode changes the behavior of the
      agent to force buffering of database objects until all related
      database objects have been modified.  In the normal mode, SET
      operations to any database object shall either be stored in a
      device's database immediately with no regard to whether other
      changes will be made or be rejected (as defined in the device-
      specific Information Profile).  In the transaction mode, SET
      operations to any database object shall be buffered until a verify
      state performs a consistency check.  When the consistency check
      completes, the device automatically transitions to the done state
      where a normal or transaction command may be issued.
      A database object is a user provided piece of setup information
      (or it may be defined in an information profile) that is necessary
      for the proper operation of a device.  It is static in nature in
      that the agent would never change it without direction from the
      management station.  For example, a parameter that defines a
      default mode of operation would be a database object.  A parameter
      that indicates the current state of the device would not be a
      database object.
      The states and commands are defined as:
      NORMAL:  SET operations behave as normal SETs and shall have an
      immediate effect on the value of any database objects used by the
      device if none of the objects contained in the operation require
      the use of the transaction mode (as defined in the device-specific
      Information Profile).  A SET operation containing any database
      object that requires the use of transaction mode shall result in a
      genErr.  This is the default state of this object.
      The only command that may be written to dbCreateTransaction while
      in this state is TRANSACTION.  Any other values written to this
      object in this state shall result in an error response of
      'badValue'.
      TRANSACTION: A SET operation of one or more database objects that
      use the same community name as used in the request for the
```

TRANSACTION state are buffered by the agent device for later consistency checks and a normal response is returned. A SET operation of one or more database objects using different community names shall result in a genErr with the index set to zero. A SET operation without a community name field (e.g., an STMP operation) shall be buffered by the agent device for later consistency checks and a normal response is returned. Standard SYNTAX checking shall take place at the time of the SET operation. A transaction may consist of multiple SET operations over multiple frames.
A SET operation for one or more non-database objects shall be processed as normal even if it uses another community name, except for this (i.e., the dbCreateTransaction) object.

A SET operation containing both database and non-database objects shall be processed in full according to these two rules. Thus, if it contains the same community name as used in the request for the TRANSACTION state, the non-database objects shall be stored immediately while the database objects shall be buffered. If it uses a different community name, the entire request will be rejected and a genErr with an index of zero shall be returned.
GET operations on any object shall return the values of the data stored in the controller and shall ignore any values contained in the buffer.
Any valid community name may read this (dbCreateTransaction) object when in this state, but only the community name used to command the object to the transaction mode and the administrator community name can set this object. A set from any other community name shall result in a genErr with an index of zero. The only commands that can be written to dbCreateTransaction while in this state are VERIFY and NORMAL. A VERIFY command will change the state to VERIFY. If a NORMAL command is received, all buffered data is discarded and the state is returned to NORMAL. Any other values written to this object when in this state shall result in an error response of ‘badValue’.
VERIFY: Specific database objects are checked for consistency. When consistency checks are complete the device will automatically advance to the DONE state.
The state of dbCreateTransaction cannot be changed when in the VERIFY state. Any values written to this object in this state shall result in an error response of ‘badValue’.
The consistency check analyzes certain critical objects 'in context' and treats them as an interrelated whole rather than separate non-related data items. The consistency check rules are not defined in this standard. They are device and implementation specific. Where applicable, the consistency check rules are defined in application specific object definition standards. A specific implementation may add additional checks beyond those defined in the standards.

A SET operation containing any database objects while in the VERIFY state shall result in a genErr with the index set to zero.
DONE: This state is entered automatically once consistency checks have completed in the VERIFY mode. The value of dbVerifyStatus and dbVerifyError indicate whether the consistency check found any errors.

A SET operation containing any database objects while in the DONE state shall result in a genErr with the index set to zero.
Any valid community name may read this (dbCreateTransaction) object when in this state, but only the community name used to command the object to the transaction mode and the administrator community name can set this object. A set from any other community name shall result in a genErr with an index of zero. The only commands that can be written to dbCreateTransaction while in this state are NORMAL and TRANSACTION. Any other values written to this object in this state shall result in an error response of 'badValue'.
If a NORMAL command is issued and dbVerifyStatus indicates doneWithNoError, the buffered data is transferred to the device memory and the state is returned to NORMAL. If a NORMAL command is issued and dbVerifyStatus indicates something other than doneWithNoError then the buffered data is discarded and the state is returned to NORMAL.
If a TRANSACTION command is issued, regardless of dbVerifyStatus, no action takes place (the buffered data is not changed) and the TRANSACTION state is re-entered.

		COMMANDED STATE (9)			
		transaction	*verify*	*normal*	*done*
CURRENT STATE	normal	transaction (1)	normal (2)	normal (2)	normal (2)
	transaction	transaction (2)	verify (3)	normal (4)	transaction (2)
	verify (7)	verify (2)	verify(2)	verify (2)	verify (2)
	done (8)	transaction (5)	done(2)	normal (6)	done (2)

Operational procedures and error responses:

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

```
            (7)  The state will automatically change to done when the
            consistency check completes.
            (8)  dbVerifyStatus and dbVerifyError are only valid in this
            state.

            (9)  All SET operations on this (dbCreateTransaction) parameter
            shall be made using a protocol that uses a community name, or
            equivalent field (e.g., SNMP).
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.1"
DEFVAL      {normal}
::= { globalDBManagement 1 }
```

2.3.2 Database Error Type Parameter

-- This object has been deprecated. See Clause D.8 for more information.

```
dbErrorType OBJECT-TYPE
SYNTAX      INTEGER { tooBig (1),
              noSuchName (2),
              badValue (3),
              readOnly (4),
              genError (5),
              updateError (6),
              noError (7)}
ACCESS      read-only
STATUS      deprecated
DESCRIPTION
"This object returns the current error status of the transaction.  The value
of this
object is only valid when the dbCreateTransaction object is in the Done or
Error state.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.2"
::= { globalDBManagement 2 }
```

2.3.3 Database Error ID Parameter

-- This object has been deprecated. See Clause D.8 for more information.

```
dbErrorID  OBJECT-TYPE
SYNTAX       OBJECT IDENTIFIER
ACCESS       read-only
STATUS       deprecated
DESCRIPTION
"This object contains the object identifier of the first object in the
transaction buffer that caused an error while dbCreateTransaction object was
in the Verifying or Updating state.  The value of this object is only valid
when the dbCreateTransaction object is in the Error state.  It is undefined
when the dbCreateTransaction object is in other states.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.3"
::= { globalDBManagement 3 }
```

2.3.4 Database Transaction ID Parameter

-- This object has been deprecated. See Clause D.8 for more information.

```
dbTransactionID  OBJECT-TYPE
SYNTAX      INTEGER (0..255)
ACCESS      read-write
```



```
STATUS      deprecated
DESCRIPTION
"This object contains the transaction ID value that is to be contained in all
SET operation writes while the dbCreateTransaction object is not in the Normal
state.  During transaction operations every SET command shall begin with a
write to this object with the current value of this object.  If a SET
operation is performed without writing to this object, or with a value that
does not match the current value, then an error response of 'genError' shall
be returned.  This mechanism is used to determine that the same management
station that started the transaction is performing the SET operations that are
being buffered or modifying the state of dbCreateTransaction.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.4"
::= { globalDBManagement 4 }
```

2.3.5 Database Make ID Parameter

-- This object has been deprecated. See Clause D.8 for more information.

```
dbMakeID OBJECT-TYPE
SYNTAX      INTEGER (0..255)
ACCESS      read-only
STATUS      deprecated
DESCRIPTION
"This object is used to create unique transaction ID's for management stations
to use when starting transactions using the dbCreateTransaction object.  This
object will be incremented by one every time it is read, so that different
values will be returned for each read.  Management stations wishing to start a
transaction should first read the dbCreateTransaction object to verify that it
is in the Normal state.  If so then the management shall GET dbMakeID to
obtain a transaction ID to use, then SET dbCreateTransaction to startCmd and
dbTransactionID to the value just received.  If the response to the SET
operation is 'noError' then the management station has started a transaction.
If the response to the SET operation is 'genError' then the management station
should read the dbCreateTransaction and dbTransactionID objects to ensure that
the error was not due to a communications retry.  If the dbCreateTransaction
is in the Transaction state, and the dbTransactionID is the same value
returned by the read of this object, then the management station is the owner
of the transaction.  If the dbTransactionID does not match the value
originally returned by this object, then the management station is not the
owner of the transaction and must wait until the dbCreateTransaction object
returns to the Normal state before attempting to start the transaction.
<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.5"
::= { globalDBManagement  5 }
```

2.3.6 Database Verify Status Parameter

```
dbVerifyStatus   OBJECT-TYPE
SYNTAX      INTEGER {   notDone (1),
                        doneWithError (2),
                        doneWithNoError  (3) }
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>This object indicates the current status of verify
      (consistency checking) processing. The value of this object is
      only meaningful when the dbCreateTransaction object is in the
      Verify or Done state.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.6"
::= { globalDBManagement 6 }
```

2.3.7 Database Verify Error Parameter

```
dbVerifyError   OBJECT-TYPE
SYNTAX      OCTET STRING (SIZE (0..255))
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>This object contains a textual description of or a
      reference to an error that was found by the verify (consistency
      checking) processing.  The value of this object is only meaningful
      when the dbCreateTransaction object is in the Done state and the
      dbVerifyStatus object is in the doneWithError state.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.2.7"
::= { globalDBManagement 7 }
```

2.4 GLOBAL TIME MANAGEMENT NODE

```
globalTimeManagement   OBJECT IDENTIFIER
::= { global 3 }
--<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3

-- This node is an identifier used to organize all objects for support of
-- time-related functions that are common to most device types.
```

2.4.1 Global Time Parameter

```
globalTime  OBJECT-TYPE
SYNTAX      Counter
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The number of seconds since the epoch of 00:00:00
      (midnight) January 1, 1970 UTC (a.k.a. Zulu or GMT).
      <Unit>second
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.1"
DEFVAL      { 0 }
::= { globalTimeManagement 1}
```

2.4.2 Global Daylight Savings Parameter

```
-- This object has been deprecated.  See Clause D.9/D.10 for more information.
-- This object has been modified with additional information on the operation
-- of 'other'.  See controllerBeginDSTMonth for additional information.

globalDaylightSaving   OBJECT-TYPE
SYNTAX   INTEGER { other (1),
                   disableDST (2),
                   enableUSDST (3),
                   enableEuropeDST (4),
                   enableAustraliaDST (5),
                   enableTasmaniaDST (6),
                   enableEgyptDST (7),
                   enableNamibiaDST (8),
                   enableIraqDST (9),
                   enableMangoliaDST (10),
                   enableIranDST (11),
                   enableFijiDST (12),
                   enableNewZealandDST (13),
                   enableTongaDST (14),
```

```
                    enableCubaDST (15),
                    enableBrazilDST (16),
                    enableChileDST (17),
                    enableFalklandsDST (18),
                    enableParaguayDST (19) }
ACCESS  read-write
STATUS  deprecated
DESCRIPTION
     "<Definition>This object specifies if the Daylight Savings Time (DST) is
     enabled, disabled or some other form of daylight savings time is active.
     other - DST adjustments by a mechanism not defined within this standard.
     disableDST -  DST clock adjustments shall NOT occur.
     enableUSDST - DST shall begin the first Sunday in April and shall end the
        last Sunday of October.  All changes of time occur at 2:00 AM.
     enableEuropeDST - DST shall start the last Sunday of March at 2:00 AM
        and ends the last Sunday of October at 3:00 AM.
     enableAustraliaDST - DST shall start the last Sunday in October at 2:00
        AM and ends the last Sunday in March at 2:00 AM.
     enableTasmaniaDST - DST shall start the first Sunday in October at 2 a.m.
        and ends the last Sunday in March at 3 a.m.
     enableEgyptDST –   DST shall start the last Friday in April and end the
        last Thursday in September.
     enableNamibiaDST – DST shall start the first Sunday in September and end
        the first Sunday in April.
     enableIraqDST –    DST shall start on April 1 and end on October 1.
     enableMongoliaDST – DST shall start the last Sunday in March and end the
        last Sunday in September.
     enableIranDST – DST shall start the first day of Farvardin and end the
        first day of Mehr
     enableFijiDST – DST shall start the first Sunday in November and end
        the last Sunday in February.
     enableNewZealandDST – DST shall start the first Sunday in October and end
        the first Sunday on or after March 5th.
     enableTongaDST – DST shall start the first Saturday in October and end
        the first Saturday on or after April 15th.
    enableCubaDST – DST shall start April 1st and end last Sunday in October.
    enableBrazilDST – DST shall start the first Sunday in October and end the
       last Sunday in February.
    enableChileDST – DST shall start the first Sunday on or after October 9th
       and end the first Sunday on or after March 9th.
    enableFalklandsDST – DST shall start the first Sunday on or after
       September 8th and end the first Sunday on or after April 8th.
    enableParaguayDST – DST shall start the first Sunday in October and end
       the last Saturday in February.
<Informative> This object is maintained for backward compatibility only. It
has been deprecated and replaced by:
    - controllerBeginDSTMonth
    - controllerBeginDSTOccurances
    - controllerBeginDSTDayOfWeek
    - controllerBeginDSTDayOfMonth
    - controllerBeginDSTSecondsToTransition
    - controllerBeginDSTSecondsToAdjust
    - controllerEndDSTMonth
    - controllerEndDSTOccurances
    - controllerEndDSTDayOfWeek
    - controllerEndDSTDayOfMonth
    - controllerEndDSTSecondsToTransition
    - controllerEndDSTSecondsToAdjust
```

```
Devices which support both this object and the new controllerBeginDST objects
shall follow the rules defined in the controllerBeginDSTMonth object.

<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.2"
REFERENCE
"NEMA TS 2  Clause 3.8.2;
http://fatty.law.cornell.edu/uscode/15/260a.html;
http://www.timing.se/Daylight.htm;
http://www.dstc.qut.edu.au/DST/marg/daylight.html#cutoffs;
http://www.dstc.qut.edu.au/DST/marg/daylight.html#cutoffs;
http://webexhibits.org/daylightsaving/g.html "
DEFVAL      { disableDST }
::= { globalTimeManagement 2 }
```

2.4.3 TimeBase Event Scheduler Node

```
timebase OBJECT IDENTIFIER ::= { globalTimeManagement 3 }
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3
-- This node is an identifier used to organize the main objects for event
-- scheduling.  Device type-specific objects (tables) pointed to are defined
-- within the appropriate MIB.
```

2.4.3.1 Maximum Number of Time Base Schedule Entries Parameter

```
maxTimeBaseScheduleEntries   OBJECT-TYPE
SYNTAX      INTEGER (1..65535)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>The value of this object specifies the maximum number
      of different entries supported by the device as shown by the
      number of rows in the timeBaseScheduleTable.
      <Unit>TimeBaseScheduleEntry
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.1"
::= { timebase 1 }
```

2.4.3.2 Time Base Schedule Table

```
timeBaseScheduleTable   OBJECT-TYPE
SYNTAX      SEQUENCE OF TimeBaseScheduleEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
      "<Definition>A table containing the time base schedule parameters
      for the device.  The number of rows in this table shall be equal
      to the maxTimeBaseScheduleEntries object.  The table references
      the appropriate day plan for the device.  The plan is determined
      by comparing the current month (MONTH), day of week (DOW) and date
      of month (DOM) to the appropriate fields. The settings for MONTH,
      DOW and DOM are connected with a logical AND.  In order to
      determine which timebased event to select, determine the event
      which has the most specific date specified.  Select the more
      specific event based on their MONTH settings; if the same, select
      the most specific DOM; if that is still the same, select the most
      specific DOW;  if that's still the same, the first occurrence
      within the time base event table shall be selected. 'More
      specific' means the least number of bits set within an object.
      All entries in Time Base Schedule Table are expressed in local
      time and date.  A row in the table may be deactivated by setting
      the Month, Day, Date, or DayPlan parameters to zero (0)
```



```
      <TableType> static
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2"
::= { timebase 2 }

timeBaseScheduleEntry   OBJECT-TYPE
SYNTAX      TimeBaseScheduleEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
      "<Definition>Event Parameters for the time based schedule
      programming of the device.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1"
INDEX   { timeBaseScheduleNumber }
::= { timeBaseScheduleTable 1 }

TimeBaseScheduleEntry ::= SEQUENCE {
            timeBaseScheduleNumber     INTEGER,
            timeBaseScheduleMonth      INTEGER,
            timeBaseScheduleDay        INTEGER,
            timeBaseScheduleDate       INTEGER,
            timeBaseScheduleDayPlan    INTEGER }
```

2.4.3.2.1 Time Base Schedule Number Parameter

```
timeBaseScheduleNumber   OBJECT-TYPE
SYNTAX      INTEGER (1..65535 )
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>The time base schedule number for objects in this
      row.  The value of this object shall not exceed the value of the
      maxTimeBaseScheduleEntries object.  The activation of a scheduled
      entry shall occur whenever allowed by all other objects within
      this table.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1.1"
::= { timeBaseScheduleEntry 1 }
```

2.4.3.2.2 Time Base Schedule Month Of Year Parameter

```
timeBaseScheduleMonth   OBJECT-TYPE
SYNTAX      INTEGER (0..65535)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The Month(s) Of the Year that the schedule entry
      shall be allowed.  Each bit represents a specific month.  If the
      bit is set to one (1), then the scheduled entry shall be allowed
      during the associated month.  If the bit is zero (0), then the
      scheduled entry shall not be allowed during the associated month.
      The bits are defined as:
       Bit     Month of Year
       0       Reserved
       1       January
       2       February
       3       March
       4       April
       5       May
       6       June
       7       July
       8       August
```

```
        9       September
        10      October
        11      November
        12      December
        13 - 15 Reserved
      Thus, a value of six (6) would indicate that the entry would only
      be allowed during the months of January and February.  A value of
      zero (0) shall indicate that this row has been disabled.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1.2"
::= { timeBaseScheduleEntry 2 }
```

2.4.3.2.3 Time Base Schedule Day Of Week Parameter

```
timeBaseScheduleDay   OBJECT-TYPE
SYNTAX      INTEGER (0..255)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The Day(s) Of Week that the schedule entry shall be
      allowed. Each bit represents a specific day of the week.  If the
      bit is set to one (1), then the scheduled entry shall be allowed
      during the associated DOW.  If the bit is set to zero (0), then
      the scheduled entry shall not be allowed during the associated
      DOW.  The bits are defined as:
        Bit     Day of Week
        0       Reserved ('Holiday', not defined by this standard)
        1       Sunday
        2       Monday
        3       Tuesday
        4       Wednesday
        5       Thursday
        6       Friday
        7       Saturday
      Thus, a value of six (6) would indicate that the entry would only
      be allowed on Sundays and Mondays.  A value of zero (0) shall
      indicate that this row has been disabled.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1.3"
::= { timeBaseScheduleEntry 3 }
```

2.4.3.2.4 Time Base Schedule Date Parameter

```
timeBaseScheduleDate   OBJECT-TYPE
SYNTAX      INTEGER (0..4294967295)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The Day(s) Of a Month that the schedule entry shall
      be allowed.  Each bit represents a specific date of the month.  If
      the bit is set to one (1), then the scheduled entry shall be
      allowed during the associated date.  If the bit is set to zero
      (0), then the scheduled entry shall not be allowed during the
      associated date.  The bits are defined as:
        Bit     Day Number
        0       Reserved
        1       Day 1
        2       Day 2
        ||
        31      Day 31
```



```
      Thus, a value of six (6) would indicate that the entry would only
      be allowed on the first and second of the allowed months.  A value
      of zero (0) shall indicate that this row has been disabled.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1.4"
::= { timeBaseScheduleEntry 4 }
```

2.4.3.2.5 Time Base Schedule Day Plan Parameter

```
timeBaseScheduleDayPlan   OBJECT-TYPE
SYNTAX      INTEGER (0..255)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>This object specifies what Plan number shall be
      associated with this timeBaseScheduleDayPlan -object.  A value of
      zero (0) shall indicate that this row has been disabled.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.2.1.5"
::= { timeBaseScheduleEntry 5 }
```

2.4.4 Day Plan Parameters

2.4.4.1 Maximum Number of Day Plans- Parameter

```
maxDayPlans   OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>The value of this object specifies the maximum, fixed
      number of different timebased Day Plans supported by the device.
      The value of this object represents the number of day plans
      (primary key into the table) available in the
      timeBaseDayPlanTable.
      <Unit>DayPlan
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.3"
::= { timebase 3 }
```

2.4.4.2 Maximum Number of Day Plan Events - Parameter

```
maxDayPlanEvents   OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>The value of this object specifies the fixed number
      of different timebased Day Plan Events within each Day Plan
      supported by the device.  The value of this object represents the
      number of rows (secondary key into the table) available within
      each of the day plans that are available in the
      timeBaseDayPlanTable.  All day plans shall have the same number of
      day plan events available for use.
      <Unit>DayPlanEvent
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.4"
::= { timebase 4 }
```

2.4.4.3 Day Plan Table

```
timeBaseDayPlanTable OBJECT-TYPE
SYNTAX      SEQUENCE OF TimeBaseDayPlanEntry
ACCESS      not-accessible
```

```
STATUS      mandatory
DESCRIPTION
      "<Definition>A table containing day plan numbers, the times when
      to implement them and the associated actions. The number of rows
      in this table shall be equal to the product of the maxDayPlans
      object and the maxDayPlanEvents object.  The dayPlanNumbers within
      this table shall begin with day plan number 1 and increment by one
      to the maxDayPlans.  The dayPlanEventNumbers within this table
      shall begin with day plan event number 1 and increment by one to
      the maxDayPlanEvents.

      This table is always used in association with device-type specific
      objects specifying device-type specific actions such as activating
      a message on a VMS sign or initiating a pattern for a signal
      controller.  A device MIB that defines an action table should
      define the relative priority of the action table as compared to
      the priority of system and other commands.  The device-type
      specific action will only be initiated when (1) the specific
      DayPlan has been activated, (2) the scheduler has sufficient
      priority to override the current operation of the device, and (3)
      at the indicated time.

      After a power recovery or after a change to globalTime, the
      operational mode called for by the scheduler shall be per the last
      event that would have been called by the currently defined
      schedule; the logic will search for all events that may have
      occurred for at least the previous 24 hours.
      <TableType> static
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5"
::= { timebase 5 }

timeBaseDayPlanEntry   OBJECT-TYPE
SYNTAX      TimeBaseDayPlanEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
      "<Definition>A table containing the timebased day plan parameters
      of a device.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1"
INDEX { dayPlanNumber, dayPlanEventNumber}
::= { timeBaseDayPlanTable 1 }

TimeBaseDayPlanEntry ::= SEQUENCE {
             dayPlanNumber          INTEGER,
             dayPlanEventNumber     INTEGER,
             dayPlanHour            INTEGER,
             dayPlanMinute          INTEGER,
             dayPlanActionNumberOID OBJECT IDENTIFIER }
```

2.4.4.3.1 Day Plan Number

```
dayPlanNumber   OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>This object specifies the day plan number for objects
      in this row. The value shall not exceed the value of the
```



```
      maxDayPlans object.  Day plan numbers are used in the TimeBase
      Event Table to specify day plan numbers to be implemented on
      specific days of the year or as part of the week plans.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1.1"
::= { timeBaseDayPlanEntry 1 }
```

2.4.4.3.2 Day Plan Event Number

```
dayPlanEventNumber   OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>This object identifies day plan event number(s) to be
      scheduled on a specific day plan number.  Several different events
      can be scheduled to take place during a day, and each of these
      events is one entry or row within a specified day plan number.
      The total number of events for one day plan shall not exceed the
      value of the maxDayPlanEvents object. If multiple non-conflicting
      events occur at the same time, they shall be logically executed in
      order of their dayPlanEventNumber with the lowest number occurring
      first.  An implementation shall omit lower number actions that are
      in conflict with higher number actions at the same time.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1.2"
::= { timeBaseDayPlanEntry 2 }
```

2.4.4.3.3 Day Plan Hour Parameter

```
dayPlanHour   OBJECT-TYPE
SYNTAX      INTEGER (0..23)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The Hour of day, as measured by the
      controllerLocalTime object, that the associated event shall become
      active.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1.3"
DEFVAL      {0}
::= { timeBaseDayPlanEntry 3 }
```

2.4.4.3.4 Day Plan Minute Parameter

```
dayPlanMinute   OBJECT-TYPE
SYNTAX      INTEGER (0..59)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>The Minute of the hour (defined in the dayPlanHour),
      as measured by the controllerLocalTime object, that the associated
      event shall become active.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1.4"
DEFVAL      {0}
::= { timeBaseDayPlanEntry 4 }
```

2.4.4.3.5 Day Plan Action Number OID Parameter

```
dayPlanActionNumberOID   OBJECT-TYPE
SYNTAX      OBJECT IDENTIFIER
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
```

```
      "<Definition>This object provides a reference to the device-type
      specific action that shall be executed.  The object shall
      reference the action by its associated object identifier,
      including its instance (i.e., the full OID of the scalar or
      columnar object).  Only objects whose description field explicitly
      states that they may be called by the action table may be
      referenced.  If a management system attempts to set this value to
      any other object identifier, the device shall respond with a
      genErr.
      Any object allowing the action table to reference it shall define
      precisely what action will take place when it is activated and
      whether the action is transitionary or continuous until
      deactivated.  The object shall also define what, if any,
      restrictions may be placed on other operations the device may be
      able to perform.
      If the action to be performed is defined by a row of a table, one
      of the index columns should be identified as the explicit object
      that is referenced.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.5.1.5"
DEFVAL      {null}
::= { timeBaseDayPlanEntry 5 }
```

2.4.4.4 Day Plan Status Parameter

```
dayPlanStatus   OBJECT-TYPE
SYNTAX      INTEGER (0..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>This object indicates the current value of the active
      day PlanNumber-object.  A value of zero (0) indicates that there
      is no dayPlanNumber that is currently active.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.6"
::= { timebase 6 }
```

2.4.4.5 Schedule Status Parameter

```
   timeBaseScheduleTableStatus OBJECT-TYPE
      SYNTAX   INTEGER (0..65535)
      ACCESS   read-only
      STATUS   mandatory
      DESCRIPTION
      "<Definition>This object indicates the number of the TimeBaseSchedule
      which is currently selected by the scheduling logic; the device may or
      may not be using the selected schedule.  The value of zero (0) indicates
      that there is no timeBaseScheduleNumber that is currently selected.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.3.7"
   ::={timebase 7}
```

2.4.5 Global Local Time Differential Parameter

```
-- This object has been deprecated.  See Clause D.9/D.10 for more information.
globalLocalTimeDifferential   OBJECT-TYPE
SYNTAX      INTEGER (-43200..43200)
ACCESS      read-write
STATUS      deprecated
DESCRIPTION
```



```
      "Indicates the number of seconds offset between local time and
      GMT.  Positive values indicate local times in the Eastern
      Hemisphere up to the International Date Line and negative values
      indicate local times in the Western Hemisphere back to the
      International Date Line.  If one of the daylight savings times is
      activated, this value will change automatically at the referenced
      time. For example, Central Standard Time (CST) is -21600 and
      Central Daylight Time (CDT) is -18000.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.4"
::= { globalTimeManagement 4 }
```

2.4.6 Standard Time Zone Parameter

```
controllerStandardTimeZone   OBJECT-TYPE
SYNTAX      INTEGER (-43200..43200)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition> Indicates the number of seconds offset between local
      Standard Time and GMT.  Positive values indicate local times in
      the Eastern Hemisphere up to the International Date Line and
      negative values indicate local times in the Western Hemisphere
      back to the International Date Line.  This value does not change
      in response to a daylight saving time event.
      <Unit>second
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.5"
DEFVAL      {0}
::= { globalTimeManagement 5 }
```

2.4.7 Local Time Parameter

```
controllerLocalTime   OBJECT-TYPE
SYNTAX      Counter
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition> The current local time expressed in seconds since
      00:00:00 (midnight) January 1, 1970 of the same time offset.  This
      value changes by 3600 seconds in response to a daylight saving
      time event.
      <Unit>second
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.6"
::= { globalTimeManagement 6 }
```

2.4.8 Daylight Saving Time Node

```
daylightSavingNode   OBJECT IDENTIFIER
::= { globalTimeManagement 7 }
--<Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7
-- This node is an identifier used to organize all objects for support of
-- the defining daylight saving time.  This function is common to most device
-- types.  See Annex A.2.2 for examples.
```

2.4.8.1 Maximum Daylight Saving Time Table Entries Parameter

```
maxDaylightSavingEntries   OBJECT-TYPE
SYNTAX  INTEGER (1..100)
ACCESS  read-only
```

```
STATUS  mandatory
DESCRIPTION
"<Definition>
      The maximum number of entries (begin and end date pairs) that the
      Daylight Saving transitions table can contain within the device.

      As of July 2007, devices used within the United States only require 1
      entry when using the generic begin and end date method.

      It is expected that for devices using the absolute date method, that the
      device would need to support at least 1 entry per year programmed.
      <Unit>
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.1"
::= { daylightSavingNode 1 }
```

2.4.8.2 Daylight Saving Time Table Parameter

```
daylightSavingTimeTable   OBJECT-TYPE
   SYNTAX   SEQUENCE OF DSTEntry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition> A table containing DST Begin and End
       dates. The number of rows in this table is equal to
       table is equal to the maxDaylightSavingEntries object.
       <TableType> static
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2"
::= { daylightSavingNode 2 }

dstEntry   OBJECT-TYPE
   SYNTAX   DSTEntry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition> The DST Begin and End dates including parameters.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1"
   INDEX   { dstEntryNumber }
::= { daylightSavingTimeTable 1 }

DSTEntry ::= SEQUENCE {
   dstEntryNumber                          INTEGER,
   controllerBeginDSTMonth                 INTEGER,
   controllerBeginDSTOccurrences           INTEGER,
   controllerBeginDSTDayOfWeek             INTEGER,
   controllerBeginDSTDayOfMonth            INTEGER,
   controllerBeginDSTSecondsToTransition   INTEGER,
   controllerBeginDSTSecondsToAdjust       INTEGER,
   controllerEndDSTMonth                   INTEGER,
   controllerEndDSTOccurrences             INTEGER,
   controllerEndDSTDayOfWeek               INTEGER,
   controllerEndDSTDayOfMonth              INTEGER,
   controllerEndDSTSecondsToTransition     INTEGER,
   controllerEndDSTSecondsToAdjust         INTEGER }
```

2.4.8.2.1 Daylight Saving Time Entry Number Parameter

```
dstEntryNumber   OBJECT-TYPE
```



```
   SYNTAX   INTEGER (0..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition> The entry number for the DST objects
       in this row.  This value shall not exceed the
       maxDaylightSavingEntries object value.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.1
       <Unit> dstEntry"
::= { dstEntry 1 }
```

2.4.8.2.2 Daylight Saving Time Beginning Month Parameter

```
controllerBeginDSTMonth   OBJECT-TYPE
SYNTAX  INTEGER { january (1),
                  february (2),
                  march (3),
                  april (4),
                  may (5),
                  june (6),
                  july (7),
                  august (8),
                  september (9),
                  october (10),
                  november (11),
                  december (12),
                  absolute (13),
                  none (14),
                  other (15) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> The month during which Daylight Saving Time begins.  An
     entry of absolute means that controllerBeginDSTSecondsToTransition
     defines an absolute time to begin DST relative to midnight
     January 1, 1970.

     An entry of 'other' shall mean that DST adjustments are by a means other
     than this object. If this object is SET to 'other', the agent shall
     return a genErr.

     An entry of 'none' shall mean that DST is controlled by this object and
     is disabled.
     Devices which support both this object and the older globalDaylightSaving
     object shall follow the following rules.
        a. If globalDaylightSaving is SET to 'other', the device shall return
           a genErr
        b. If globalDaylightSaving is SET to any other value, the device shall
           automatically transition the value of controllerBeginDSTType to
           'other'
        c. If controllerBeginDSTType is SET to 'other', the device shall
           return a genErr
        d. If controllerBeginDSTType is SET to any other value, the device
           shall automatically transition the value of globalDaylightSaving to
           'other'

     Devices which support this object and another mechanism to control
     daylight savings that is not defined by this standard shall follow the
```

```
     following rules:
        a. If controllerBeginDSTType is SET to 'other', the device shall
           return a genErr
        b. If globalDaylightSaving or controllerBeginDSTType is SET to any
           value except 'other' the device shall prevent the other mechanism
           from affecting DST adjustments.
        c. If the other mechanism is enabled in any way, the device shall
           transition the values of globalDaylightSaving and
           controllerBeginDSTType to 'other'
     <Unit>
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.2"
     DEFVAL { march }
::= { dstEntry 2 }
```

2.4.8.2.3 Daylight Saving Time Beginning Occurrence Parameter

```
controllerBeginDSTOccurrences   OBJECT-TYPE
SYNTAX  INTEGER { first (1),
                  second (2),
                  third (3),
                  fourth (4),
                  last (5),
                  secondLast (6),
                  thirdLast (7),
                  fourthLast (8),
                  specificDayOfMonth (9) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> For values of 1-4, the number of occurrences of the
      specific day of week that must occur on or after
      controllerBeginDSTDayOfMonth until the daylight saving transition
      shall take place.

      For values of 5-8, the number of occurrences of the specific day of
      week that must occur on or before controllerBeginDSTDayOfMonth until
      the daylight saving transition shall take place.

      For value = 9, controllerBeginDSTDayOfMonth defines the specific day
      of the month that the daylight saving time transition occurs regardless
      of value in controllerBeginDSTDayOfWeek object.

      NOTE: In order to specify the last occurrence of a specified day of the
      month, simply specify the last occurrence of the specified day of the
      week on or before the last day of the month (e.g. 31).
     <Unit>
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.3"
     DEFVAL { second }
::= { dstEntry 3 }
```

2.4.8.2.4 Daylight Saving Time Beginning Day of Week Parameter

```
controllerBeginDSTDayOfWeek   OBJECT-TYPE
SYNTAX  INTEGER { sunday (1),
                  monday (2),
                  tuesday (3),
                  wednesday (4),
```



```
                  thursday (5),
                  friday (6),
                  saturday (7) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> The Day of the week on which Daylight Saving Time
      begins.  This object shall only apply if the
      controllerBeginDSTOccurrences is 1-8.
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.4"
      DEFVAL { sunday }
::= { dstEntry 4 }
```

2.4.8.2.5 Daylight Saving Time Beginning Day of Month Parameter

```
controllerBeginDSTDayOfMonth  OBJECT-TYPE
SYNTAX  INTEGER (1..31)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition>
      If controllerBeginDSTOccurrences is 1-8: The day of the month from
      which to begin counting occurrences of a specific day of the week
      (forward for values 1-4, and backwards for values 5-8).

      If controllerBeginDSTOccurrences is 9: The specific day of the month
      on which the transition will occur.
     <Unit> day of month
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.5"
     DEFVAL { 1 }
::= { dstEntry 5 }
```

2.4.8.2.6 Daylight Saving Time Beginning Seconds to Transition Parameter

```
controllerBeginDSTSecondsToTransition    OBJECT-TYPE
SYNTAX  INTEGER (0..4294967295)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition>
      If controllerBeginDSTMonth = absolute, then this object defines when
      DST begins based on the seconds from midnight January 1, 1970.

      If controllerBeginDSTMonth = 1-12 (January to December), then this
      object defines the time when DST begins in seconds past midnight.
     <Unit>seconds
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.6"
     DEFVAL { 7200 }
::= { dstEntry 6 }
```

2.4.8.2.7 Daylight Saving Time Beginning Seconds to Adjust Parameter

```
controllerBeginDSTSecondsToAdjust    OBJECT-TYPE
SYNTAX  INTEGER (-32768.. 32767)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
```

```
     "<Definition>This is the offset in seconds that will be added to local
      time to determine the daylight saving time, when daylight saving time
      has begun, but has not yet ended.
     <Unit>seconds
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.7"
     DEFVAL { 3600 }
::= { dstEntry 7 }
```

2.4.8.2.8 Daylight Saving Time Ending Month Parameter

```
controllerEndDSTMonth   OBJECT-TYPE
SYNTAX  INTEGER { january (1),
                  february (2),
                  march (3),
                  april (4),
                  may (5),
                  june (6),
                  july (7),
                  august (8),
                  september (9),
                  october (10),
                  november (11),
                  december (12),
                  absolute (13) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> The month during which Daylight Saving Time ends.  An entry
      of absolute means that controllerEndDSTSecondsToTransition defines an
      absolute time to end DST relative to midnight January 1, 1970.
     <Unit>
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.8"
     DEFVAL { november }
::= { dstEntry 8 }
```

2.4.8.2.9 Daylight Saving Time Ending Occurrences Parameter

```
controllerEndDSTOccurrences   OBJECT-TYPE
SYNTAX  INTEGER { first (1),
                  second (2),
                  third (3),
                  fourth (4),
                  last (5),
                  secondLast (6),
                  thirdLast (7),
                  fourthLast (8),
                  specificDayOfMonth (9) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> For values of 1-4, the number of occurrences of the
       specific day of week that must occur on or after
       controllerEndDSTDayOfMonth until the daylight saving transition
       shall take place.

       For values of 5-8, the number of occurrences of the specific day
       of week that must occur on or before controllerEndDSTDayOfMonth
```



```
       until the daylight saving transition shall take place.

       For value = 9, controllerEndDSTDayOfMonth defines the specific day
       of the month that the daylight saving time transition occurs
       regardless of value in controllerEndDSTDayOfWeek object.

       NOTE: In order to specify the last occurrence of a specified day of
       the month, simply specify the last occurrence of the specified day
       of the week on or before the last day of the month (e.g. 31).
     <Unit>
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.9"
     DEFVAL { first }
::= { dstEntry 9 }
```

2.4.8.2.10 Daylight Saving Time Ending Day of Week Parameter

```
controllerEndDSTDayOfWeek   OBJECT-TYPE
SYNTAX  INTEGER { sunday (1),
                  monday (2),
                  tuesday (3),
                  wednesday (4),
                  thursday (5),
                  friday (6),
                  saturday (7) }
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> The Day of the week on which Daylight Saving Time
     ends.  This object shall only apply if the controllerEndDSTOccurrences
     is 1-8.
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.10"
     DEFVAL { sunday }
::= { dstEntry 10 }
```

2.4.8.2.11 Daylight Saving Time Ending Day of Month Parameter

```
controllerEndDSTDayOfMonth    OBJECT-TYPE
SYNTAX  INTEGER (1..31)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition>
       If controllerEndDSTOccurrences is 1-8: The day of the month from
       which to begin counting occurrences of a specific day of the week
      (forward for values 1-4, and backwards for values 5-8).

      If controllerEndDSTOccurrences is 9: The specific day of the month
      on which the transition will occur.
     <Unit> day of month
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.11"
     DEFVAL { 1 }
::= { dstEntry 11 }
```

2.4.8.2.12 Daylight Saving Time Ending Seconds to Transition Parameter

```
controllerEndDSTSecondsToTransition    OBJECT-TYPE
SYNTAX  INTEGER (0..4294967295)
```

```
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition>
      If controllerEndDSTMonth = absolute, then this object defines when
      DST ends based on the seconds from midnight January 1, 1970.
      If controllerEndDSTMonth = 1-12 (January to December), then this
      object defines the time when DST ends in seconds past midnight.
     <Unit> seconds
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.12"
     DEFVAL { 7200 }
::= { dstEntry 12 }
```

2.4.8.2.13 Daylight Saving Time Ending Seconds to Adjust Parameter

```
controllerEndDSTSecondsToAdjust    OBJECT-TYPE
SYNTAX  INTEGER (-32768.. 32767)
ACCESS  read-write
STATUS  mandatory
DESCRIPTION
     "<Definition> This is the offset in seconds that will be added to local
       time to determine the daylight saving end time, when daylight saving
       time has ended.
     <Unit> seconds
     <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.3.7.2.1.13"
     DEFVAL { 0 }
::= { dstEntry 13 }
```

2.5 REPORT PARAMETER NODE

-- NOTE—The objects originally under this node have been moved to NTCIP 1103.

2.6 STMP OBJECT NODE

-- NOTE: The objects originally under this node have been moved to NTCIP 1103.

2.7 PMPP OBJECT NODE

-- NOTE: These objects will be moved to NTCIP 2101 at some point in the
-- future.

```
profilesPMPP OBJECT IDENTIFIER
::= { profiles 3 }
-- <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3


-- This node is an identifier used to group all objects for support of the
-- PMPP function that are common to all device types.  The objects under this
-- node are placed under the Protocols\Profiles\PMPP subtree within the NEMA
-- node, but they have been listed here due to the lack of a separate document
   that lists these objects.
```

2.7.1 Maximum HDLC Group Address Parameter

```
maxGroupAddresses OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
```



```
DESCRIPTION
      "<Definition>The maximum number of group addresses this device
      supports. This object indicates the maximum number of rows in the
      hdlcGroupAddressTable.
      <Unit>address
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.1"
::= {profilesPMPP 1 }
```

2.7.2 HDLC Group Address Table

```
hdlcGroupAddressTable OBJECT-TYPE
SYNTAX      SEQUENCE OF HdlcGroupAddressEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
      "<Definition> A table containing group addresses at which a device
      may receive frames.
      <TableType> static
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.2"
::= { profilesPMPP 2 }

hdlcGroupAddressEntry   OBJECT-TYPE
SYNTAX      HdlcGroupAddressEntry
ACCESS      not-accessible
STATUS      mandatory
DESCRIPTION
      "<Definition> An entry in the group address table that contains a
      device’s data link layer group address at which it will accept
      frames.
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.2.1"
INDEX { hdlcGroupAddressIndex }
::= { hdlcGroupAddressTable 1 }

HdlcGroupAddressEntry ::= SEQUENCE {
   hdlcGroupAddressIndex            INTEGER,
   hdlcGroupAddress                 INTEGER, -- deprecated
   hdlcGroupAddressNumber           INTEGER }
```

2.7.2.1 HDLC Group Address Index Parameter

```
hdlcGroupAddressIndex   OBJECT-TYPE
SYNTAX      INTEGER (1..255)
ACCESS      read-only
STATUS      mandatory
DESCRIPTION
      "<Definition>The index number for the group address in this row.
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.2.1.1"
::= { hdlcGroupAddressEntry 1}
```

2.7.2.2 HDLC Group Address Parameter

-- This object has been deprecated. See Clause D.19 for more information.

```
hdlcGroupAddress  OBJECT-TYPE
SYNTAX      INTEGER
ACCESS      read-write
STATUS       deprecated
DESCRIPTION
```

```
      "A group address for the data link layer. For PMPP, the syntax is
      an 8 or 16 bit entry with the second low order bit set to a one
      indicating that this is a group address.
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.2.1.2"
REFERENCE
"NEMA TS 3.3 Clause 3.3.3.1"
::= { hdlcGroupAddressEntry 2}
```

2.7.2.3 HDLC Group Address Number Parameter

```
hdlcGroupAddressNumber  OBJECT-TYPE
SYNTAX      INTEGER (0..62)
ACCESS      read-write
STATUS      mandatory
DESCRIPTION
      "<Definition>A group address number prior to any encoding for the
      data link layer. The address of 63 is reserved for the all
      stations address. The value of zero (0) shall disable this row of
      the table.

      NOTE that in PMPP all group addresses are encoded in one byte.
      <Object Identifier> 1.3.6.1.4.1.1206.4.1.2.3.2.1.3"
REFERENCE
"NTCIP 2101"
DEFVAL  { 0 }
::= { hdlcGroupAddressEntry 3}
```

2.8 SECURITY NODE

```
-- NOTE: The objects under this heading have been be moved to NTCIP 1103.

END -- NTCIP1201-2007 DEFINITIONS
```

2.9 NEW AUXILIARY I/O OBJECTS IN THIS STANDARD

```
-- ****************************************************************************
-- Filename:    1201v0302 AuxIO2.mib
-- Description: This MIB represents the data elements that first appeared in
--              NTCIP 1201 v02.31.  In moving the objects from NTCIP 1203 to
--              NTCIP 1201, the objects were registered under a new node but
--              were not given new names.  For backwards compatibility, this
--              MIB retains the registration used in NTCIP 1201 v02.41 but
--              changes the names of the objects.  In the context of
--              implementations that supported NTCIP 1201 v02.31, there are no
--              changes except for the object names.
-- Source:      NTCIP 1201 Av02.09e
-- MIB Revision History:
-- 07/12/06     Created this file from original standard
--              Change formatting so that spacing appears correctly in the
--              text format of the MIB
--              Added but commented registration names and IMPORTS if one uses
--              the new NTCIP 8004 v01.37 SMI structure
--              Added Header and DESCRIPTION subfields per NTCIP 8004
--              Changed all object to deprecated
-- 10/02/06     Changed all STATUSes to mandatory to eliminate checking errors
--              Revised object names
--              Organized objects as standalone MIB
```



```
-- 09/18/07     Updated this header
--
--Copyright 2006 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.
--*****************************************************************************

AuxIO-1201 DEFINITIONS ::= BEGIN
```

```
IMPORTS
   OBJECT-TYPE
   FROM RFC-1212
   DisplayString
   FROM RFC1213-MIB
   global
   FROM NTCIP8004-A-2004;

auxIOv2  OBJECT IDENTIFIER ::= { global 7}
-- <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7
-- This node is an identifier used to group all objects supporting auxiliary
-- I/O functions
--
-- NOTE: An earlier version of the functionality defined by these objects
-- was defined under the experimental node.  For the purposes of backwards
-- compatibility, the names of these  objects, originally defined in the
-- version 2 standard, have been modified to eliminate any conflicts/confusion
-- with the objects defined under the experimental node.  For those agents
-- that may support these objects and those originally defined under the
-- experimental node (see 2.10), the object definitions shall be treated as
-- aliases such that a write to an object in one group shall act as write to
-- the corresponding object in the other group.  As aliases, a read of an
-- object in this group is equivalent to a read of the corresponding object in
-- the auxIO group.
```

2.9.1 Maximum Number of Digital Auxiliary IOs Parameter

```
maxAuxIOv2TableNumDigitalPorts OBJECT-TYPE
   SYNTAX   INTEGER (0..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition> The number of rows contained in the 'auxIOv2Table' with
       the auxPortType set to 'digital'.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.1"
::= { auxIOv2 1}
```

2.9.2 Maximum Number of Analog Auxiliary IOs Parameter

```
maxAuxIOv2TableNumAnalogPorts OBJECT-TYPE
   SYNTAX   INTEGER (0..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>The number of rows contained in the 'auxIOv2Table'
       with the auxPortType set to 'analog'.
      <Unit>port
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.2"
::= {auxIOv2 2}
```

2.9.3 Auxiliary IO Table Parameter

```
auxIOv2Table   OBJECT-TYPE
   SYNTAX   SEQUENCE OF AuxIOv2Entry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition>A table providing the means to access any non-
```



```
       mission-critical or safety-related auxiliary I/O of the
       Controller, this includes reading inputs and setting outputs.
       The number of rows in this table equals the sum of the values
       of the 'maxAuxIOv2TableNumDigitalPorts' and
       'maxAuxIOv2TableNumAnalogPorts' objects.
       This table shall not be used to
       control or monitor any safety related equipment.  The user should
       be aware that the electrical levels used by the ports are not
       standardized by these objects; such information should be
       contained in the hardware manual.
      <TableType> static
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3"
::= { auxIOv2 3}

auxIOv2Entry OBJECT-TYPE
   SYNTAX   AuxIOv2Entry
   ACCESS   not-accessible
   STATUS   mandatory
   DESCRIPTION
      "<Definition>Parameters of the auxiliary I/O table.
      <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1"
INDEX {auxIOv2PortType, auxIOv2PortNumber}
::={auxIOv2Table 1}

AuxIOv2Entry ::= SEQUENCE {
   auxIOv2PortType               INTEGER,
   auxIOv2PortNumber             INTEGER,
   auxIOv2PortDescription        DisplayString,
   auxIOv2PortResolution         INTEGER,
   auxIOv2PortValue              INTEGER,
   auxIOv2PortDirection          INTEGER,
   auxIOv2PortLastCommandedState INTEGER
   }
```

2.9.3.1 Auxiliary Port Type Parameter

```
auxIOv2PortType OBJECT-TYPE
   SYNTAX   INTEGER{
                   other (1),
                   analog (2),
                   digital (3)
                  }
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>Indicates the type of auxiliary I/O, which can be
       analog or digital.
      <Information> From Version 1 to Version 2 of these objects, it was
      determined that ports are either digital, analog, or other.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.1"
::= {auxIOv2Entry 1}
```

2.9.3.2 Auxiliary Port Number Parameter

```
auxIOv2PortNumber OBJECT-TYPE
   SYNTAX   INTEGER (1..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
```

```
      "<Definition>Indicates the port number for the associated port
       type.  Port numbers are used sequentially from one to max for
       each port type.  There can be a port 1 for analog port and port 1
       for digital port.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.2"
::= {auxIOv2Entry 2}
```

2.9.3.3 Auxiliary Description Parameter

```
auxIOv2PortDescription  OBJECT-TYPE
   SYNTAX   DisplayString (SIZE (0..255))
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
      "<Definition>Informational text field describing the device at the
       associated auxiliary I/O
       <Informative> In NTCIP 1203 v01.15, the SYNTAX SIZE was listed
       as (0..50). In all versions of NTCIP 1201 v02 this was changed to
       0..255).
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.3"
::= {auxIOv2Entry 3}
```

2.9.3.4 Auxiliary Resolution Parameter

```
auxIOv2PortResolution  OBJECT-TYPE
   SYNTAX   INTEGER (1..255)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>Defines number of bits used for the IO-port (e.g.
       width of digital, resolution of analog). Thus, this feature
       allows the digital monitoring (via NTCIP) of an analog port on
       the agent.
       <Informative> In NTCIP 1203 v01,15, the SYNTAX constraint was listed as
       (1..255). In all versions of NTCIP 1201 v02 this was changed to
       (1..32).  However, in order to address backwards compatibility and
       the ‘aliasing’ between the version 1 object (‘’) and this amendment
       version, the value range of this object was changed back to (1..255).
       <Unit>bit
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.4"
::= {auxIOv2Entry 4}
```

2.9.3.5 Auxiliary Value Parameter

```
auxIOv2PortValue  OBJECT-TYPE
   SYNTAX   INTEGER (0..4294967295)
   ACCESS   read-write
   STATUS   mandatory
   DESCRIPTION
      "<Definition>For input or bidirectional ports, this contains the
       current value of the input.  For output ports, this is the last
       commanded value of the port.  A genError shall be generated, if
       this object is set and the port is an input. The actual value
       exchanged shall not exceed [2^(auxIOv2PortResolution) – 1]; any
       SET operation to a value in excess of this number shall result in
       a genErr and any GET response in excess of this value shall be
       considered erroneous.
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.5"
::= {auxIOv2Entry 5}
```

2.9.3.6 Auxiliary Port Direction Parameter

```
auxIOv2PortDirection OBJECT-TYPE
   SYNTAX   INTEGER {
                     output (1),
                     input (2),
                     bidirectional (3)}
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>Indicates whether state of this port can be set (output),
       read (input) or both (bidirectional).
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.6"
::= {auxIOv2Entry 6}
```

2.9.3.7 Auxiliary Port Last Commanded State Parameter

```
auxIOv2PortLastCommandedState OBJECT-TYPE
   SYNTAX   INTEGER (0..4294967295)
   ACCESS   read-only
   STATUS   mandatory
   DESCRIPTION
      "<Definition>For bi-directional ports, this object shall indicate
       the last state to which the auxIOv2PortValue object was set.  For
       output ports, this value shall always be equal to the
       auxIOv2PortValue object.  For input ports, this value shall
       always be zero (0).
       <Object Identifier> 1.3.6.1.4.1.1206.4.2.6.7.3.1.7"
::= {auxIOv2Entry 7}

END -- AuxIO-1201 DEFINITIONS
```

2.10 OLD AUXILIARY I/O OBJECTS FROM NTCIP 1203:1997

```
-- ************************************************************************
-- Filename:    1201v0301 AuxIO1.mib
-- Description: This MIB represents the data elements that appeared in
--              NTCIP 1203:1997.  In moving the objects to NTCIP 1201 and
--              registering them under the global node, new definitions had to
--              be created and therefore the original definitions had to be
--              deprecated.  They have been moved here for purposes of
--              maintaining backwards compatibility.  In the context of
--              implementation that supports these objects, there is no
--              difference between what appeared in NTCIP 1203:1997 and what
--              appears here.
-- Source:      NTCIP 1201v03.02
-- MIB Revision History:
-- 07/12/06     Created this file from original standard
--              Change formatting so that spacing appears correctly in the
--              text format of the MIB
--              Added but commented registration names and IMPORTS if one uses
--              the new NTCIP 8004 v01.37 SMI structure
--              Added Header and DESCRIPTION subfields per NTCIP 8004
--              Changed all object to deprecated
-- 11/20/06     Revised filename and source information
-- 09/18/07     Updated this header
--
--
--Copyright 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.
--*****************************************************************************

AuxIO-1203 DEFINITIONS ::= BEGIN

IMPORTS
   OBJECT-TYPE
   FROM RFC-1212
   DisplayString
   FROM RFC1213-MIB
--   experimental
--   FROM NEMA_SMI
   nemaExperimental
   FROM NTCIP8004-A-2004;
-- Replace the previous 2 uncommented lines above with the commented lines
-- when using 8004 v01.37 SMI


--   exp-global OBJECT IDENTIFIER ::= {experimental 1}
   expGlobal OBJECT IDENTIFIER ::= {nemaExperimental 1}
-- Replace the previous uncommented line above with the commented line when
-- using 8004 v01.37 SMI

--   auxiliaryIO OBJECT IDENTIFIER ::= { exp-global 1}
   auxiliaryIO  OBJECT IDENTIFIER ::= { expGlobal 1}
-- Replace the previous uncommented line above with the commented line when
-- using 8004 v01.37 SMI

-- NOTE: These objects are still logically located under the nemaExperimental
-- node and use their originally defined textual names and OIDs.  For the
-- purposes of backwards compatibility, the objects STATUS has been changed to
-- deprecated. For those agents that may support these objects and the
-- new objects under the global node (see 2.9), the object definitions shall
-- be treated as aliases in that a write to an object in one group shall act
-- as write to the corresponding object in the other group. As aliases, a read
-- of an object in one group shall also act as read of the corresponding
-- object in the other group.
```

2.10.1 Maximum Number of Digital Auxiliary IOs Parameter

```
maxAuxIODigital OBJECT-TYPE
   SYNTAX   INTEGER (0..255)
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition> The number of rows contained in the 'auxIOTable' with the
       auxPortType set to 'digital'.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.1 "
::= {auxiliaryIO 1}
```

2.10.2 Maximum Number of Analog Auxiliary IOs Parameter

```
maxAuxIOAnalog OBJECT-TYPE
   SYNTAX   INTEGER (0..255)
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition>The number of rows contained in the 'auxIOTable' with the
       auxPortType set to 'analog'.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.2 "
::= {auxiliaryIO 2}
```

2.10.3 Auxiliary IO Table Parameter

```
auxIOTable  OBJECT-TYPE
   SYNTAX   SEQUENCE OF AuxIOEntry
   ACCESS   not-accessible
   STATUS   deprecated
   DESCRIPTION
      "<Definition> A table providing the means to access the auxiliary I/O of
       the Controller, this includes reading inputs and setting outputs.
       A maximum of 255 auxiliary IOs can be defined for all, digital,
       analog or other types of ports.
       <TableType> static
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3 "
::= { auxiliaryIO 3}

auxIOEntry OBJECT-TYPE
   SYNTAX   AuxIOEntry
   ACCESS   not-accessible
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Parameters of the auxiliary IO table.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1 "
   INDEX {auxIOPortType, auxIOPortNumber}
::={auxIOTable 1}

AuxIOEntry ::= SEQUENCE {
   auxIOPortType        INTEGER,
   auxIOPortNumber      INTEGER,
   auxIODescription     DisplayString,
   auxIOResolution      INTEGER,
   auxIOValue           INTEGER,
   auxIOPortDirection   INTEGER
   }
```

2.10.3.1 Auxiliary Port Type Parameter

```
auxIOPortType OBJECT-TYPE
   SYNTAX   INTEGER{
                  other (1),
                  analog (2),
                  digital (3)
                  }
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Indicates the type of auxiliary I/O, which can be analog,
       digital or other.
       <Object Identifier>  1.3.6.1.4.1.1206.2.1.1.3.1.1 "
::= {auxIOEntry 1}
```

2.10.3.2 Auxiliary Port Number Parameter

```
auxIOPortNumber OBJECT-TYPE
   SYNTAX   INTEGER (1..255)
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Indicates the port number for the associated port type.
       Port numbers are used sequentially from one to max for each port
       type.  There can be a port 1 for analog port and port 1 for digital
       port.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1.2"
::= {auxIOEntry 2}
```

2.10.3.3 Auxiliary Description Parameter

```
auxIODescription OBJECT-TYPE
   SYNTAX   DisplayString (SIZE (0..50))
   ACCESS   read-write
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Informational text field describing the device at the
       associated auxiliary I/O
       <Informative> In NTCIP 1203 v01,15, the SYNTAX SIZE was listed
       as (0..50).  In all versions of NTCIP 1201 v02, auxIO2Description (this
       object's alias) was changed to (0..255).  This does not present a
       backwards compatibility issue if a the NTCIP 1201 v2 management station
       limits the size of the DisplayString to 50 characters.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1.3 "
::= {auxIOEntry 3}
```

2.10.3.4 Auxiliary Resolution Parameter

```
auxIOResolution OBJECT-TYPE
   SYNTAX   INTEGER (1..255)
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Defines number of bits used for the IO-port (e.g. width of
       digital, resolution of analog).
       <Informative> In NTCIP 1203 v01,15, the ACCESS was listed
       as read-write. In this version of NTCIP 1201 v02, this is changed to
       read-only .  Resolution is fixed by the hardware implementation and may
```

```
       not be changed.  The SYNTAX was also changed in all versions of NTCIP
       1201 showing this object but the change to read-only makes this a moot
       point.
       <Unit>bit
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1.4 "
::= {auxIOEntry 4}
```

2.10.3.5 Auxiliary Value Parameter

```
auxIOValue OBJECT-TYPE
   SYNTAX   INTEGER (0..4294967295)
   ACCESS   read-write
   STATUS   deprecated
   DESCRIPTION
      "<Definition> For input or bidirectional ports, this contains the
       current value of the input.  For output ports, this is the last
       commanded value of the port.  A genError shall be generated, if this
       object is set and the port is an input.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1.5 "
::= {auxIOEntry 5}
```

2.10.3.6 Auxiliary Port Direction Parameter

```
auxIOPortDirection OBJECT-TYPE
   SYNTAX   INTEGER {
                  output (1),
                  input (2),
                  bidirectional (3)}
   ACCESS   read-only
   STATUS   deprecated
   DESCRIPTION
      "<Definition> Indicates whether state of this port can be set (output),
       read (input) or both (bidirectional).
       <Informative > The ACCESS has been changed from what originally
       appeared in NTCIP 1203v01.15 because it was an error.
       <Object Identifier> 1.3.6.1.4.1.1206.2.1.1.3.1.6 "
::= {auxIOEntry 6}

END -- AuxIO-1203 DEFINITIONS
```

Section 3
Conformance

NOTE — The conformance requirements previously included in version 1 have been removed from this standard. This document only defines the data that may be useful for a given device; any requirements for supporting a specific piece of data is defined in device-specific standards, such as NTCIP 1202.

Annex A
Concept of Operations
(Normative)

This Annex provides examples of how a management station may interface with a device complying with this standard as envisioned by the authors. Any device claiming conformance with the subject features depicted in these figures shall support the exchanges as shown. However, the flexible design of the NTCIP protocols allows a large number of other possibilities and these figures do not limit any other requirements of these standards. These diagrams are merely provided to promote a common understanding of how systems may be designed in order to increase the likelihood of interchangeability in deployed systems.

Three use cases are presented, as shown in Figure A-1.

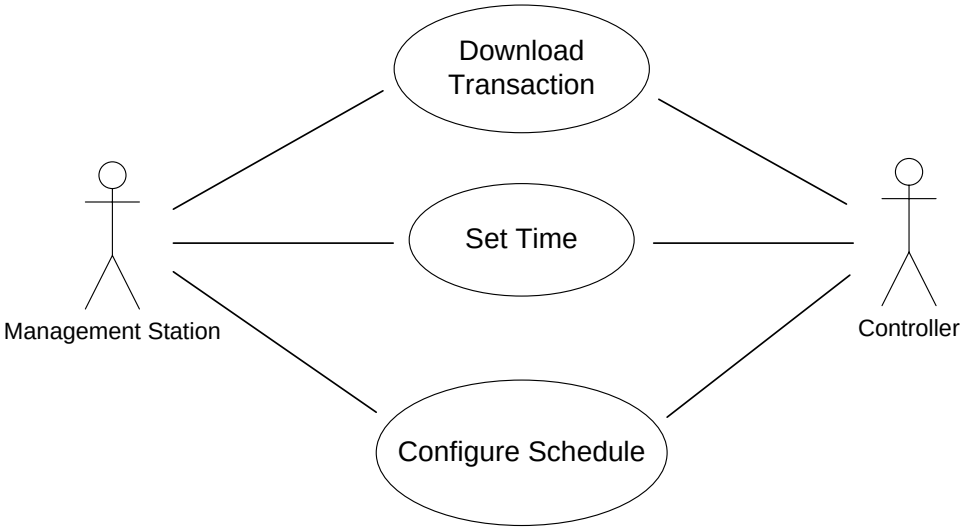

Figure A-1: Global Use Cases

A.1 DOWNLOAD TRANSACTION USE CASE

The first use case is for a Transaction. The intent of this use case is that a management station has a need to download several inter-related parameters to the controller. Because the parameters are inter-related, they must be set simultaneously in order for the set operation to be validated by the controller (e.g., the download may consist of a set of parameters, whose sum must equal the sum of another set of parameters; and the management station wishes to change the sum for both sets).

The parameters that require the use of the transaction mode are device-specific. Some devices may not require support of the transaction feature, while other devices may require SET operations on any database object to be within the transaction mode.

When used, the feature allows a device to buffer a series of set operations on database parameters and to implement all operations simultaneously in order to properly perform controller consistency checks.

The normal, fault-free process is in Figure A-2.

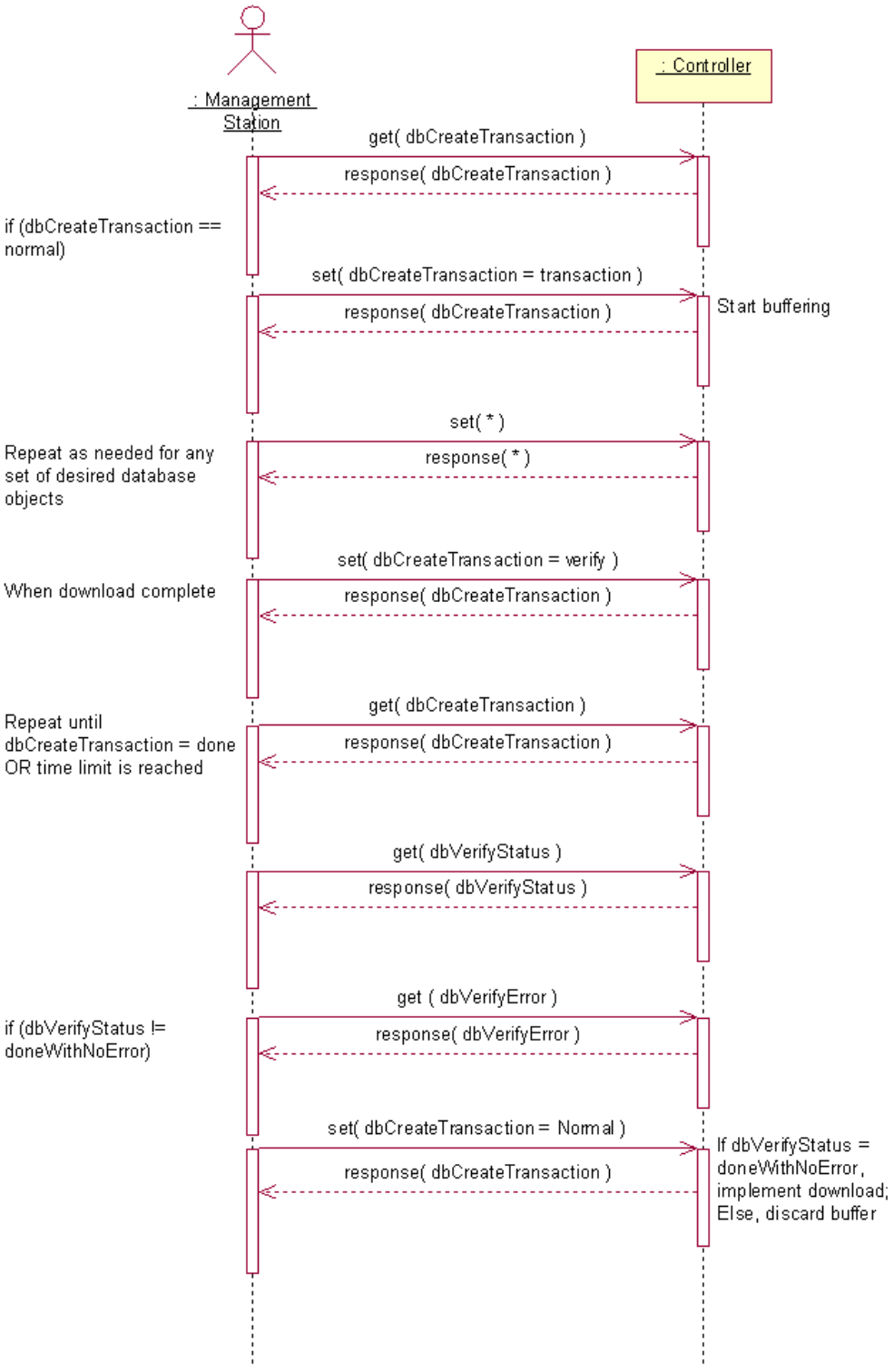

Figure A-2: Fault Free Process Dialog

Within this mode, the controller operates as a state machine as described in the definition of dbCreateTransaction. Figure A-3 supplements this definition and provides a formal UML representation of the state machine.

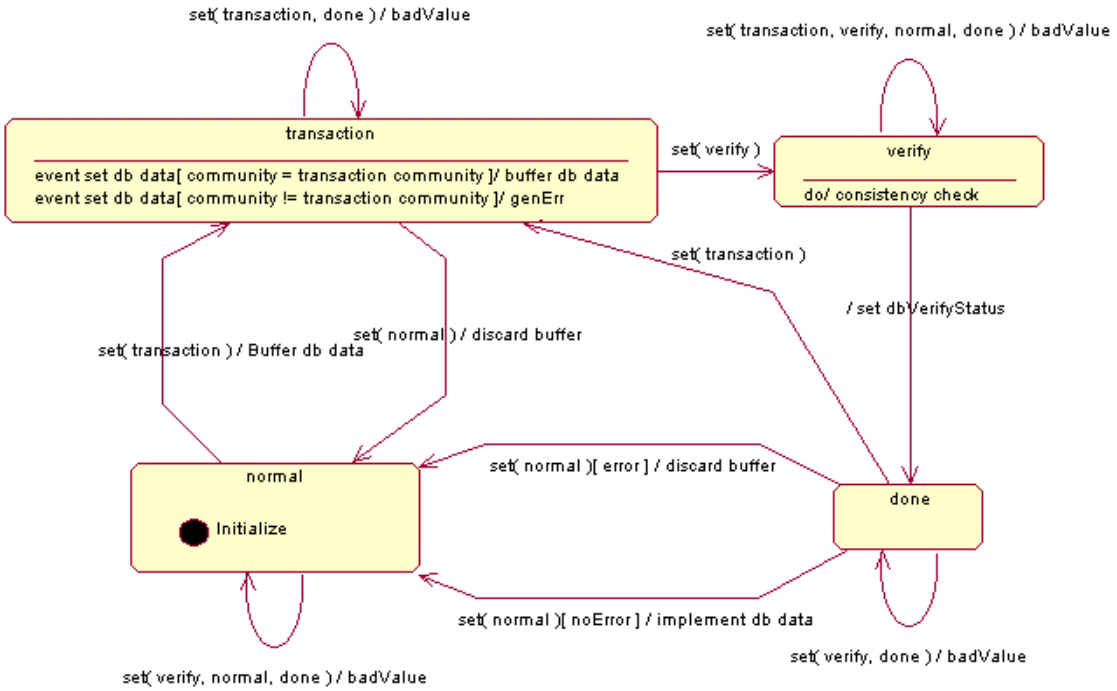

Figure A-3: Controller State Machine

A.2 SET TIME

The second use case is to set the time in a controller. There are four key parameters that affect the local time stored in the controller:

- globalTime (which is time in UTC)
- globalDaylightSaving (deprecated)
- begin and end Daylight Saving Time objects
- controllerStandardTimeZone (which is the offset between local Standard Time and UTC)

All of these parameters are independent from one another and thus a controller shall allow a management station to set any or all of these parameters in any order using one or more set operations and may additionally combine these parameters in any fashion with other parameters.

When setting any one of these values, the indicated object shall be set to the indicated value and the value of controllerLocalTime shall be updated to reflect this new value; but none of the other time objects shall be affected.

A.2.1 Example 1 – Changing Global Time

Original Values:

globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: disableDST
controllerStandardTimeZone: -21600

controllerLocalTime: 1023256800 (6:00 AM 5 June 2002)

Set Operation
globalTime: 1023282000 (1:00 PM 5 June 2002)

Updated Values:
globalTime: 1023282000 (1:00 PM 5 June 2002)
globalDaylightSaving: disableDST
controllerStandardTimeZone: -21600 (-6 hours)
controllerLocalTime: 1023260400 (7:00 AM 5 June 2002)

A.2.2 Example 2 – Changing Daylight Saving Time
Original Values:
globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: disableDST
controllerStandardTimeZone: -21600
controllerLocalTime: 1023256800 (6:00 AM 5 June 2002)

Set Operation
controllerBeginDSTMonth: march (3)
controllerBeginDSTOccurrences: second (2)
controllerBeginDSTDayOfWeek: sunday (1)
controllerBeginDSTDayOfMonth: 1 (meaning the first day of the month)
controllerBeginDSTSecondsToTransition: 7200 (meaning 2 hours after midnight in seconds)
controllerBeginDSTSecondsToAdjust: 3600 (meaning the offset between local time and daylight savings time is 1 hour
controllerEndDSTDayOfMonth: november (11)
controllerEndDSTOccurrences: first (1)
controllerEndDSTDayOfWeek: sunday (1)
controllerEndDSTDayOfMonth: 1 (meaning the first day of the month)
controllerEndDSTSecondsToTransition: 7200 (meaning 2 hours after midnight in seconds)
controllerEndDSTSecondsToAdjust: -3600 (meaning the offset between local time and daylight savings time is -1 hour

Updated Values:
globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: other
controllerBeginDSTMonth: march
controllerBeginDSTOccurrences: second
controllerBeginDSTDayOfWeek: sunday
controllerBeginDSTDayOfMonth: 1
controllerBeginDSTSecondsToTransition: 7200
controllerBeginDSTSecondsToAdjust: 3600
controllerEndDSTMonth: november
controllerEndDSTOccurrences: first
controllerEndDSTDayOfWeek: sunday
controllerEndDSTDayOfMonth: 1
controllerEndDSTSecondsToTransition: 7200
controllerEndDSTSecondsToAdjust: -3600
controllerStandardTimeZone: -21600 (-6 hours)
controllerLocalTime: 1023260400 (7:00 AM 5 June 2002)

A.2.3 Example 3 – Changing Time Zone
Original Values:

globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: disableDST
controllerBeginDSTMonth: other
controllerStandardTimeZone: -21600 9 (-6 hours)
controllerLocalTime: 1023256800 (6:00 AM 5 June 2002)

Set Operation
ControllerStandardTimeZone: -18000

Updated Values:
globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: disableDST
controllerBeginDSTMonth: other
controllerStandardTimeZone: -18000 (-5 hours)
controllerLocalTime: 1023260400 (7:00 AM 5 June 2002)

A.2.4 Example 4 – Changing Multiple Parameters

Original Values:
globalTime: 1023278400 (12:00 noon 5 June 2002)
globalDaylightSaving: other
controllerBeginDSTMonth: none
controllerStandardTimeZone: -21600
controllerLocalTime: 1023256800 (6:00 AM 5 June 2002)

Set Operation
globalTime: 1023282000 (1:00 PM 5 June 2002)
globalDaylightSaving: other
controllerBeginDSTMonth: march
controllerBeginDSTOccurrences: second
controllerBeginDSTDayOfWeek: sunday
controllerBeginDSTDayOfMonth: 1
controllerBeginDSTSecondsToTransition: 7200
controllerBeginDSTSecondsToAdjust: 3600
controllerEndDSTMonth: november
controllerEndDSTOccurrences: first
controllerEndDSTDayOfWeek: sunday
controllerEndDSTDayOfMonth: 1
controllerEndDSTSecondsToTransition: 7200
controllerEndDSTSecondsToAdjust: -3600
controllerStandardTimeZone: -18000

Updated Values:
globalTime: 1023282000 (1:00 PM 5 June 2002)
globalDaylightSaving: other
controllerBeginDSTMonth: march
controllerBeginDSTOccurrences: second
controllerBeginDSTDayOfWeek: sunday
controllerBeginDSTDayOfMonth: 1
controllerBeginDSTSecondsToTransition: 7200
controllerBeginDSTSecondsToAdjust: 3600
controllerEndDSTMonth: november
controllerEndDSTOccurrences: first
controllerEndDSTDayOfWeek: sunday
controllerEndDSTDayOfMonth: 1

controllerEndDSTSecondsToTransition: 7200
controllerEndDSTSecondsToAdjust: -3600
controllerStandardTimeZone: -18000 (-5 hours)
controllerLocalTime: 1023267600 (9:00 AM 5 June 2002)

A.3 CONFIGURE SCHEDULER

This use case depicts an approach to configuring the time base schedule. Figure A-4 indicates that the device-specific action table should be configured first, followed by the day plan parameters, followed by the time base schedule entries. This approach minimizes the likelihood of an invalid reference occurring during download (i.e., a schedule entry referencing an invalid day plan). However, other approaches may be supported.

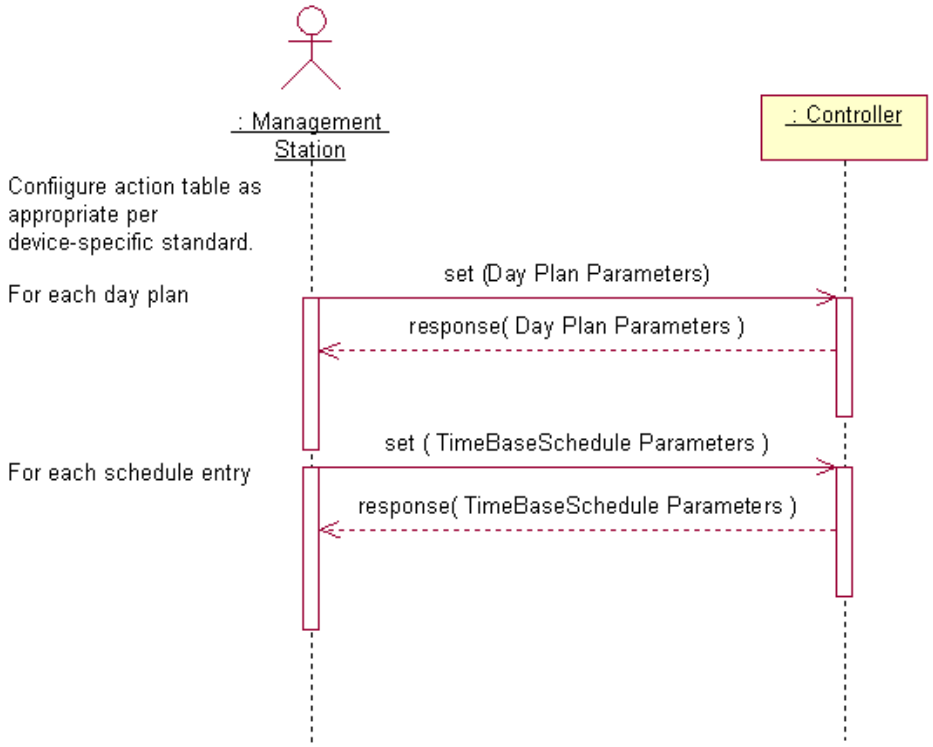

Figure A-4: Configure Scheduler

Annex B
Test Procedures
(Normative)

NOTE — This annex will identify the test procedures associated with the functionalities defined in this standard. In version 1201 v03, this Annex is a placeholder for test procedures that will be added in a future version.

Annex C
Class Diagrams
(Informative)

This annex provides an overview of the data defined by this standard through the use of UML Class Diagrams. The information presented in this annex is formally defined elsewhere in this standard; however, these figures depict key characteristics of these definitions in a concise manner and are provided as a useful reference tool for system designers.

The diagrams conform to the modeling conventions defined by ISO 14817 and were used to develop the ISO 14817 conforming Descriptive Names as shown within each object definition in Section 2 of this standard. The ObjectClassTerm of the descriptive name is indicated by the name of each box within the figures and the propertyTerm is shown as being an item within the box. These Descriptive Names are also used by the on-line ITS Data Registry as the primary name of each data concept.

NOTE — While the discussion within this section indicates that virtually every feature is optional, in order to claim conformance with various NTCIP standards, support for many of these features may be mandatory.

C.1 CONFIGURATION INFORMATION

Figure C-1 depicts the configuration data stored by a controller.

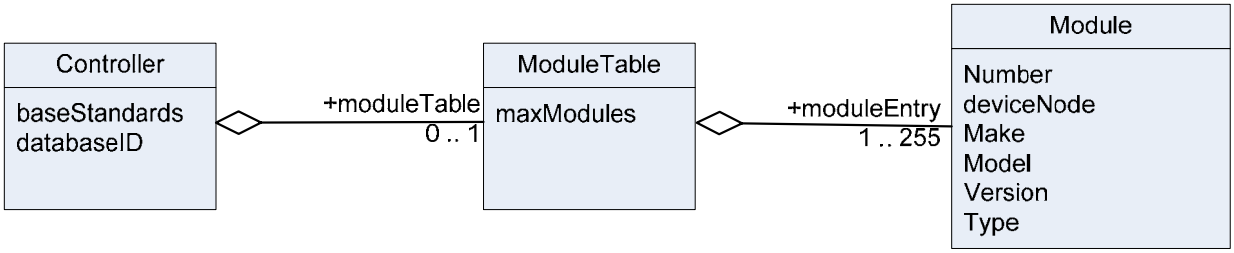

Figure C-1: Class Diagram of the Configuration Data

The figure indicates a controller may have a database identifier and zero or one module tables. If there is a module table, then the controller may additionally support an object defining the maximum number of modules supported within the table, which may be between one and 255, as indicated by the link to the Module class. For each module, the controller may support a variety of information, including:

- The module number
- The device node to which the module relates
- The make of the module
- The model of the module
- The version of the module, and
- The type of module

C.2 TRANSACTION INFORMATION

Figure C-2 depicts the transaction state data stored by a controller.

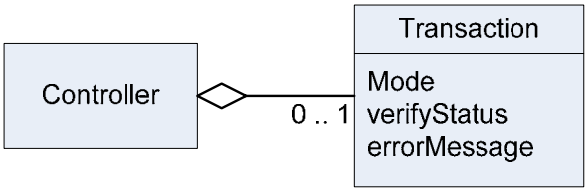

Figure C-2: Class Diagram of the Transaction Service

The figure indicates a controller may support a transaction feature. The feature is characterized by the following information:

- A mode
- A status, and
- An error code

C.3 TIME INFORMATION

Figure C-3 depicts the time related data stored by a controller.

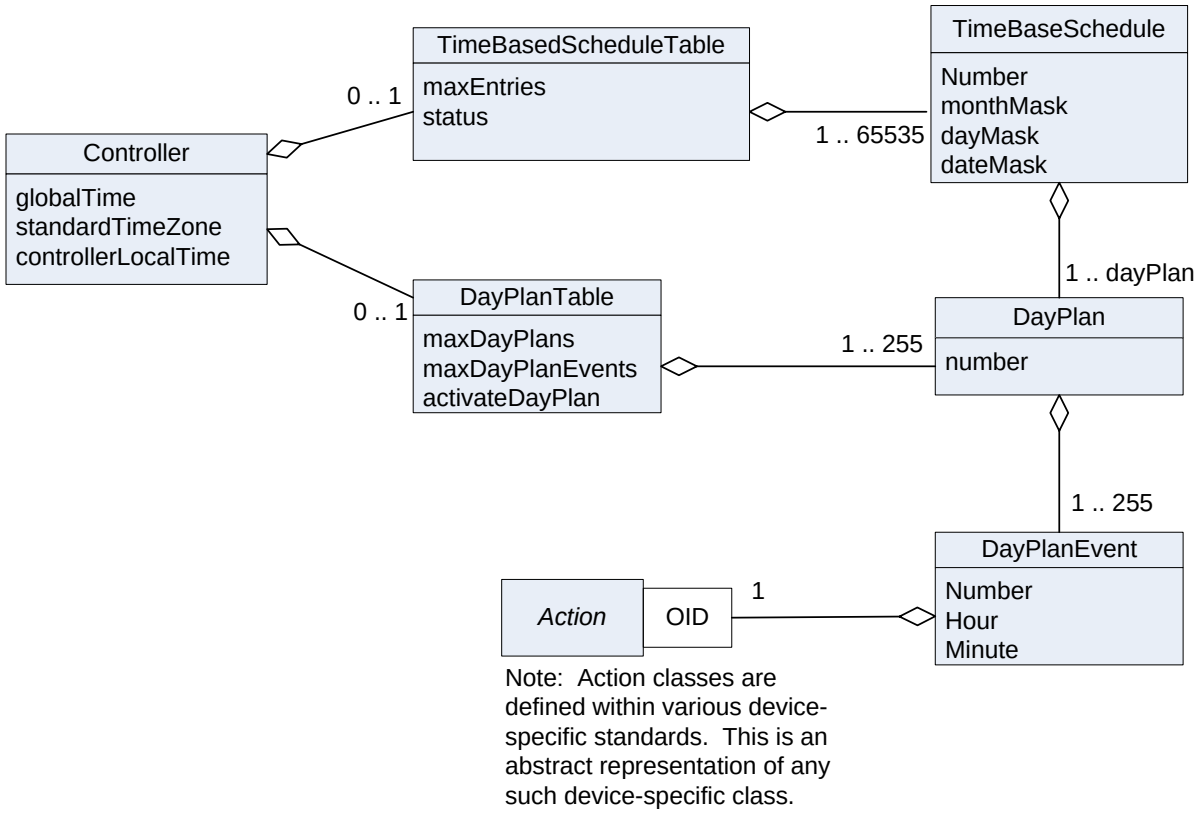

Figure C-3: Class Diagram of Time Information

The figure indicates a controller may store time information, including:

- The current time in UTC
- An indication of the time zone when in standard time
- An indication of the local time, which includes and accounts for daylight saving time

The controller may also support a timebase schedule table. If this is supported, it is characterized by the maximum number of entries that it may contain, which must be at least one and may be no greater than 65535, and a status. For each entry, the following information may be stored:

- A schedule number
- A month mask indicating which months the schedule may be active
- A day mask indicating which days of the week the schedule may be valid
- A date mask indicating which dates of the month the schedule may be active
- A link to a day plan record

In order to have a link to a day plan, the day plan must also be supported; which in turn requires that its container class, the day plan table must also be supported. The day plan table is characterized by:

- The maximum number of day plans that may be stored, which must be between one and 255,
- The maximum number of events that may occur during a day, which must be between one and 255
- An indication of the day plan that is currently active

The day plan itself only consists of the day plan number and a link to between one and 255 day plan events. Each day plan event is described by:

- A number,
- The hour during which the event occurs
- The minute during which the event occurs
- The status of the action, and
- A link to the specific action to be performed

The specific action to be performed is defined elsewhere due to the device specific nature of actions.

C.4 AUXILIARY INPUT/OUTPUT INFORMATION

Figure C-4 depicts the auxiliary input/output data stored by a controller. Two diagrams are shown, one depicting the methods and object definitions defined in version 1 of NTCIP 1201 (originally defined in NTCIP 1203 v1) and the methods and objects defined in version 2 of NTCIP 1201.

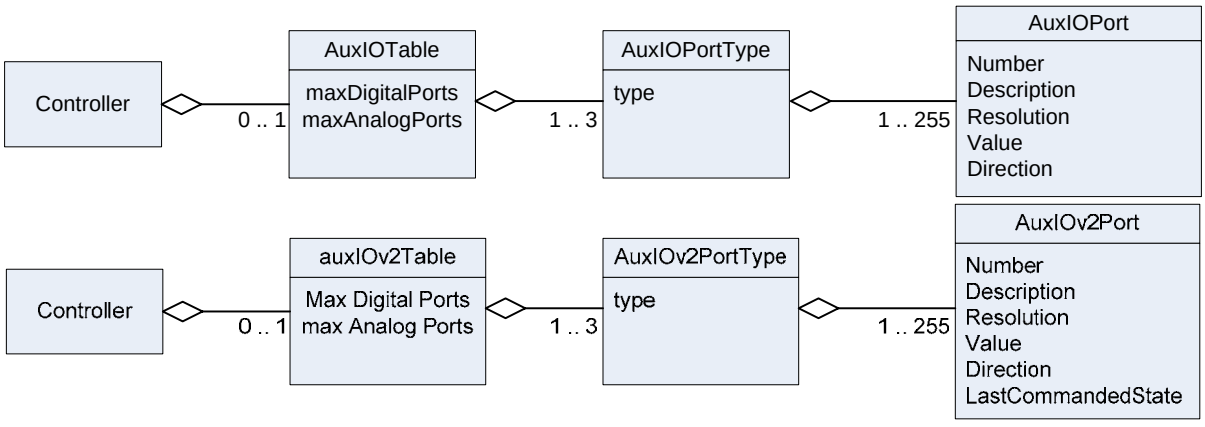

Figure C-4: Class Diagrams for Auxiliary Input/Output Services (version 1 and version 2)

The figure indicates a controller may support an auxiliary input/output table (AuxIO2 and/or AuxIO). If either is supported, it is characterized by the maximum number of digital and analog ports supported by the device. Each port type is allocated to its own sub-table in the AuxIOPortType table, which contains multiple entries, one for each port, where each port is characterized by:

- A number

- A description
- A resolution of the data supported by the port
- A value
- A direction
- The last commanded state (only in version 2)

Annex D
Summary of Changes
(Informative)

To the extent reasonable, the NTCIP community attempts to minimize the number of changes to a document in order to minimize interoperability problems among different versions of the same standard. However, on occasion, problems are identified with existing standards that necessitate a change. The NTCIP effort rectifies such problems while attempting to minimize the impact on existing implementations. This annex explains the problem identified resulting in each change, a description of the change made, and an analysis of the impact of each change on implementations.

The changes in this version of the standard reflect lessons learned from the deployment of version 02 of the standard, incorporate better documentation (in the Annex) of some of the logic required to implement the standards, and to add new features requested by the ITS community. Specific changes made to this standard between the version 01 (1996) including Amendment 1 and this version 03 are documented in the following clauses.

D.1 UPDATED ISO TREE

Due to the various other changes in the standard, the ISO Tree contained in Clause 1.5 was updated to properly reflect the contents of the standard. The update to this figure should not cause any interoperability problems.

D.2 UPDATES TO CONFORM WITH NTCIP 8004

The data stored in field devices are often retrieved by a central system and then may be exchanged with other centers as a part of regional communications. These center-to-center communications use protocols other than SNMP and require the data to be defined according to either IEEE 1489 (or its recently approved update known as ISO 14817). The fact that the version 01s of NTCIP did not define data in this format created ambiguities for center-to-center implementations.

In order to ensure that there would be a single definition for all NTCIP data, regardless of what context it was used in (e.g., center-to-center vs. center-to-field), the NTCIP community defined an enhanced MIB format, as defined in NTCIP 8004, to be used for all new and updated NTCIP standards.

The additions that this update creates (e.g., the <DEFINITION> tags, etc.) should not cause any interoperability problems.

D.3 UPDATED NAME OF THE MIB

Changes to a MIB can affect the way other MIBs import data. Thus, when a MIB imports data from another MIB, it should be able to unambiguously reference the specific version of the MIB that it wants to import. Therefore, every update to an NTCIP standard results in an update to the name of the MIB according to the rules in NTCIP 8004.

Additionally, because there are now two versions of the auxiliary input/output object definitions, two additional MIBs were created, each of which containing one of the AuxIO definitions.

The update to the MIB name should not cause any interoperability problems, and in fact prevents ambiguity as to which version of this MIB may be referenced from another MIB.

D.4 ADDED DEFAULT VALUE STATEMENTS

Interoperability problems can arise when different controllers initialize differently. As a result, this standard has standardized the default initialization value of several configuration and control parameters.

This is a change to the standard and may result in some version 01 devices performing slightly differently than 02 devices. However, this *reduces* interoperability problems overall. Current implementations operate differently from one another and any central system must be customized to handle this uniqueness for each manufacturer. By defining the default value within the standard, this customization can be avoided in the future.

D.6 ENHANCED MODULE VERSION DEFINITION

The module table is intended to provide basic information about the make, model, and version of the controller. However, the original version of the standard provided a generic format for the version that did not adequately allow for proper configuration management of software. The new standard defines a detailed format for the presentation of the version information.

While this is a change to the standard to which some version 01 devices may not conform, it does not present any real interoperability problems between version 01 and version 02 devices.

D.7 ADDED AN OBJECT TO IDENTIFY SUPPORTED STANDARDS

Several integrators have expressed concerns over the ability to be able to quickly determine to which standards and which versions of standards a device claims conformance. By being able to query the device to determine which standards it supports, a central system will be able to quickly determine how to manage the device. Therefore an object providing this information in a standard format has been added.

This addition should not create any interoperability problems. A central system will be able to readily identify any version 01 device since it will return a noSuchName error.

D.8 CORRECTED THE DATABASE TRANSACTION FEATURE

The transaction mode process was modified by NTCIP 1201:1996 Amendment 1, which was approved in 2001. Implementations discovered that the original process did not provide for the desired operation in the presence of multiple management stations (e.g., a central and a local laptop). Specifically, there were problems with the second management station overriding the first operation in order to issue a control command. The solution deprecated dbErrorID, dbTransactionID, and dbMakeID; revised the definition of dbCreateTransaction, and created two new objects labeled dbVerifyStatus and dbVerifyError.

This change was made to resolve existing interoperability problems. While version 01 implementations will have to be changed in order to conform to the new standard, this is an improvement in the sense that the version 01 feature did not work as intended.

NOTE on the dbVerifyStatus object: To align NTCIP 1201 v02 with other NTCIP and several Internet standards, the object definitions in NTCIP 1201 v01.10 that had enumerated values starting with a value of (0) have been changed in NTCIP 1201 v02 to start with a value of (1). The changed definition in NTCIP 1201 v02 makes this object incompatible with NTCIP 1201:1996 v01.10.

D.9 ADDED SUPPORT FOR ADDITIONAL DAYLIGHT SAVING MODES

-- The *globalDaylightSaving* object was deprecated after 1201 version v02.32; therefore, the following paragraphs are no longer relevant. See Clause D.10 for more information.

Several parties located outside of the US are now deploying NTCIP for various devices and have pointed out that the NTCIP should support all of the various daylight savings plans. Thus, these have been added to the daylight savings object.

This addition is fully backwards compatible and should not cause any interoperability problems. It will have no effect on systems in the US; version 01 systems outside of the US have not had a way to offer support of other daylight saving modes in a standard way, but with the version 02 enhancement, they will now be able to offer this feature.

D.10 ADDED NEW OBJECTS TO ADDRESS US DAYLIGHT SAVING TIME MODIFICATIONS

The globalDaylightSaving object was deprecated and a table allowing to define to define the beginning and ending of daylight saving time were added (see controllerBeginDSTMonth). The definition of these parameters in a table allows to define one or more sets of daylight saving time.

These changes were due to new rules enacted by the U. S. Congress to take effect in 2007. Since the DST rules have changed about every 10 years, the new set of DST objects allow users to define the start and end of DST.

D.11 ADDED A SCHEDULE STATUS OBJECT

Some agencies have wanted to be able to monitor the logic of the timebase schedule a little closer and as a result, we have added a status object to the timebase schedule table. This is an extra feature that is fully backwards compatible.

D.12 CLARIFIED DEFINITIONS OF DAY PLAN OBJECTS

Various questions had been raised about the precise meaning of the object definitions for the day plan table. Version 02 clarifies these definitions in response to these questions. However, the clarifications reflect actual implementations and should not result in any interoperability problems; rather they are likely to prevent interoperability problems in the future.

D.13 CORRECTED PROBLEMS WITH THE LOCAL TIME LOGIC

A problem was discovered with the time differential logic in that if the globalTime was set during the one-hour fall-back period of the daylight saving time logic; there was an ambiguity as to what time was intended. Manufacturers overcame this ambiguity in their own implementations in a variety of ways, many of which created interoperability problems with other manufacturers. Several options were considered to correct this flawed logic, but they all resulted in some level of interoperability problems. Thus, the working group concluded that the best solution was to produce the cleanest design which required deprecating the global time differential object (globalLocalTimeDifferential) and adding new objects for local time (controllerLocalTime) and time zone (controllerStandardTimeZone).

Version 02 corrects an existing interoperability problem. This does result in a minor compatibility problem between the variety of version 01 interpretations and the version 02 design, but the working group was unable to find an alternative solution that adequately corrected the problem without presenting new problems. By deprecating objects and creating new objects, any central system will quickly discover (by receiving a noSuchName error) if it tries to access the feature using the wrong version.

D.14 CLARIFIED DEFINITIONS RELATED TO THE EVENT LOG

The WG received a variety of detailed comments about the exact definitions used for objects in the event log. As a result of these comments, the working group made several clarifications, but in all cases, these merely clarified the text and explained how manufacturers had implemented the features. It is not expected that any of these clarifications will result in interoperability problems.

D.15 REORDERED CLAUSES FOR THE EVENT LOG

The order of the subclauses related to the event log proved confusing to some readers and the WG therefore decided to reorder the subclauses. However the OBJECT IDENTIFIERs for the objects have not changed and this is merely an editorial change; therefore there should be no interoperability problems created by this reordering.

After the above changes were made, the report node and event related objects were moved to NTCIP 1103

D.16 ADDED SUPPORT FOR ANOTHER MODE TO THE EVENT LOG

Based on requests from implementers, the WG added a new mode for the event log configuration table (andedWithValue) and provided better explanations of the definitions of each mode.

This addition is fully backwards compatible and the explanations will hopefully prevent future interoperability problems.

After the above changes were made, the report node and event related objects were moved to NTCIP 1103

D.17 ADDED ERROR VALUE TO THE EVENT CONFIGURATION STATUS

Based on requests from implementers, the WG added an error code to the status object of the event configuration table in order to ensure that the controller is not programmed to repeatedly check an invalid condition. The WG also added logic to the object that requires a consistency check whenever the configuration of the row changes.

This change presents only a minor compatibility challenge between versions, but prevents interoperability problems where some manufactures had used the 'other' code to mean error. The WG determined that this was the least problematic solution.

After the above changes were made, the report node and event related objects were moved to NTCIP 1103

D.18 CORRECTED SYNTAX OF EVENT LOG SIZE OBJECT

The original standard indicated that the lower bound of the event log size was zero, however, if the size was zero, there would be no table and this object should not be supported. Thus, in order to avoid this contradiction, the lower limit was redefined to be one.

This will not create any interoperability problems between versions since any implementation supporting this feature will have a value greater than one.

After the above changes were made, the report node and event related objects were moved to NTCIP 1103

D.19 REPLACED THE GROUP ADDRESS OBJECT

Version 01 of the standard had an object defining the PMPP group address to which the device belonged; however the meaning of the value had been interpreted in two different ways. One group held that the intent was that the value was supposed to be the group address number that was encoded in the PMPP address field. The other group contended that the value was the encoded PMPP address field. Due to this conflict and resulting non-interoperability in deployed systems, the existing object (hdlcGroupAddress) was deprecated and a new object defined (hdlcGroupAddressNumber) to resolve the issue.

While the solution of replacing the existing object presents a minor interoperability issue, the solution does provide an unambiguous definition of the object and any central system will be able to readily identify version 01 implementations since they will not support the version 02 object.

D.20 ADDED GENERIC AUXILIARY I/O OBJECTS

The development of the DMS standard identified the need to support auxiliary I/O ports. This need was later realized by several other groups, including the ESS WG. As a result, the auxiliary objects defined in NTCIP 1203 were moved to this standard. An effort by the WG responsible for NTCIP 1201 to just refine, enhance, and rename the object definitions were identified to lead to interoperability problems. Therefore, the original object definitions moved from NTCIP 1203v1 to NTCIP 1201 were retained (see Section 2.10) but deprecated, which means they will be ultimately removed from this standard. The renamed and redefined object definitions are now also defined in this standard (see Section 2.9).

Due to the provision of 2 sets of object definitions in this standard, there will be no interoperability problems with version 01 deployments.

D.21 REMOVED CONFORMANCE STATEMENTS

Deployments using the version 01 NTCIP standards highlighted problems with writing procurement specifications using conformance groups. Problems also arose as working groups began updating their standards as the user needs and requirements for each feature were typically not defined in a clear fashion within the subject version 01 standards.

As a result, the NTCIP community has changed the format of NTCIP standards to follow an outline that is based on a systems engineering process (SEP). The result of this change is that the conformance groups have been eliminated from the standards and replaced with a Protocol Requirements List (PRL). In the case of NTCIP 1201, the PRL is located in the subject device standard that references NTCIP 1201. This table, combined with the referenced Requirements Traceability Matrix (RTM) now defines the conformance requirements for the standard rather than the conformance groups and conformance statement used in version 01 standards.

This change should not present any interoperability problems with version 01 implementations.

D.22 ADDED A CONCEPT OF OPERATIONS

One of the problems that many implementers had in deploying the first version of the standard was in understanding the intended operations of some features. In order to address this issue, the WG included Annex A to explain how various features were intended to operate.

Although this is normative text, these are intended to clarify the text that already existed in the version 01 standard and are therefore not expected to produce any interoperability problems.

D.23 PREPARED COMMUNICATION OBJECTS TO BE MOVED TO 1103

The development of NTCIP 1103 resulted in the realization that the security objects and the event log objects should be moved to the new standard because they relate more to application layer issues than to the end-application. While the WG concluded that they should be moved to NTCIP 1103, they have been left in this standard until the NTCIP 1103 standard is approved and published.

D.24 DELETED ANNEX B TO DOCUMENT DEPRECATED OBJECTS

Version 02 contained an Annex B that contained and documented all objects that were deprecated from the standard (version 01 to version 02). The purpose of this Annex B was to provide future developers

with an understanding of objects that may exist in or are used by legacy equipment. However, the WG subsequently decided that this Annex D will fulfill this purpose and that Annex B is to contain future test procedures.

The inclusion of this information will not result in any interoperability problems and may assist in newer systems being able to communicate with version 01 devices.

D.25 ADDED CLASS DIAGRAMS

Many users of version 01 documentation found it difficult to understand the context of the various objects defined in the Management Information Base (MIB). While the various object definitions provided the detailed definition of each object, it was difficult for them to readily obtain a high-level view of how the data worked together.

Annex C (Informative) was added to the standard in order to provide the reader with high-level graphical images that depict the various relationships among all of the data defined by the standard, including the rules on multiplicity (i.e., how many of one object might exist for a given instance of another object). While all of this information was (and still is) recorded within the textual definition of the objects, providing high-level graphical depictions of these relationships facilitate this understanding.

This addition is marked informative and is only intended for clarification. It is not believed to have any impact on existing systems.

D.26 ADDED GENERIC SNMP INTERFACE DEFINITIONS

NTCIP 1203v2 contains the definitions for a generic Simple Network Management Protocol (SNMP) interface. This definition was brought into this global object definition standard because these operations are generic to all field devices.

Annex E
SNMP INTERFACE
(Normative)

The device shall conform to the requirements for the Simple Network Management Protocol (SNMP) as defined in NTCIP 1103. Sections E.1 through E.4 provide a description of the key services offered by SNMP assuming no errors; precise rules and procedures are defined in NTCIP 1103. Section E.5 extends the requirements of NTCIP 1103 by providing additional requirements that supplement, but do not replace any requirements of NTCIP 1103.

In order to promote interoperability and to the reflect marketplace realities, each NTCIP Technical Working Group must define in the device-specific standard for which they are responsible which of the NTCIP 1103 defined standards (Simple Network Management Protocol [SNMP], Simple Transportation Management Protocol [STMP] and the Simple Fixed Message Protocol [SFMP]) the device must support. For example, the NTCIP WG responsible for the dynamic message sign (DMS) standard (NTCIP 1203) have decided that only the SNMP standard defined in NTCIP 1103 must be supported, and that support of STMP and SFMP is in fact discouraged because "these have not been widely implemented in DMS and thus would likely result in decreased interoperability, limited competition, and increased costs for testing, integration, and maintenance.".

E.1 GENERIC SNMP GET INTERFACE

SNMP defines a generic process by which a management station can retrieve data from a device. This process consists of a Get request (GET) and a GetResponse as depicted in Figure E-3. Both the Get request and the GetResponse messages contain a list of objects as defined by the varBindingList structure (see Section E.4).

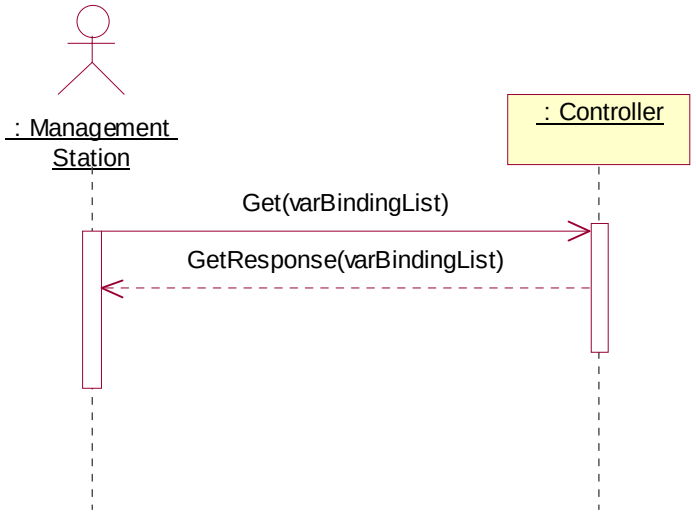

Figure E-1: SNMP Get Interface

E.2 GENERIC SNMP GET-NEXT INTERFACE

SNMP defines a process by which a management station can explore data within a device. This process consists of a GetNext request and a GetResponse as depicted in Figure E-4. Both the GetNext request and the GetResponse messages contain a list of objects as defined by the varBindingList structure (see Section E.4).

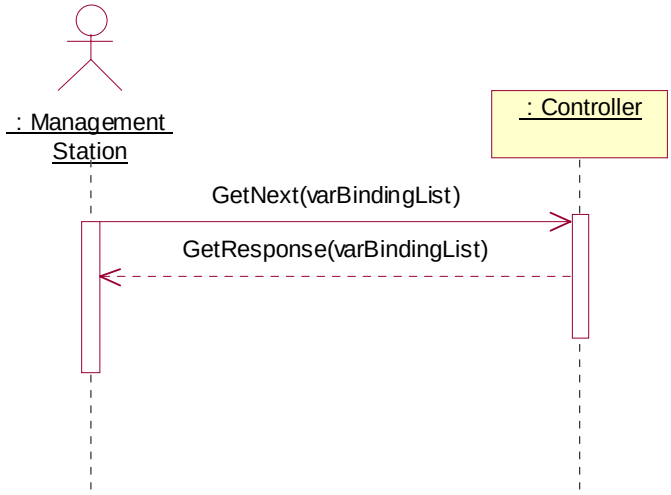

Figure E-2: SNMP GetNext Interface

E.3 GENERIC SNMP SET INTERFACE

SNMP defines a generic process by which a management station can send data to a device. This process consists of a Set request and a GetResponse as depicted in Figure E-5. Both the Set request and the GetResponse messages contain a list of objects as defined by the varBindingList structure (see Section E.4).

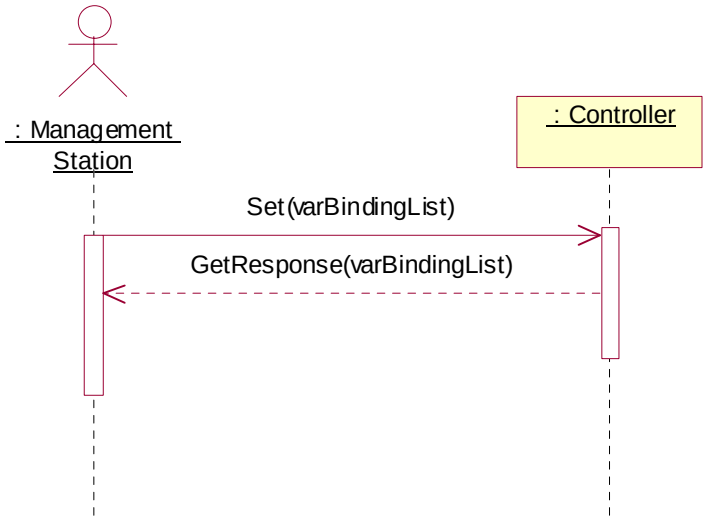

Figure E-3: SNMP Set Interface

NOTE: The response message issued to an SNMP Set request is the same message structure as used to respond to an SNMP Get request. The SNMP standard calls this response message a GetResponse, but it is in fact a response to either a GET or a SET.

E.4 VARIABLE BINDING LIST STRUCTURE

The requests and responses for the Get, Get Next and Set operations, all use the varBindingList structure. NTCIP 1103 defines this structure as containing zero or more varBindings, where each varBinding is defined to consist of an object name (as indicated by an Object Identifier (OID)) and the associated object value. This is relationship is depicted in the following figure.

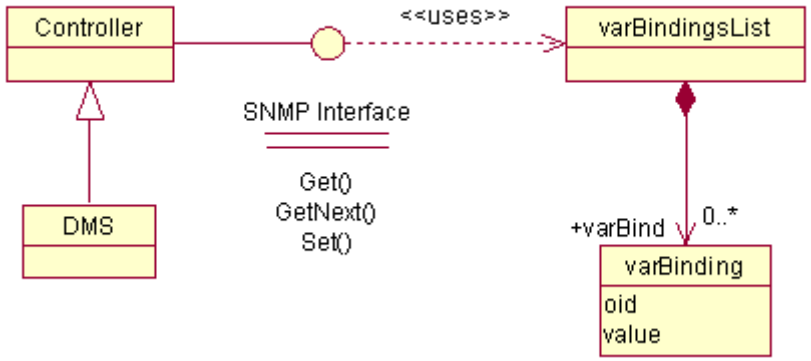

Figure E-4: SNMP Interface - View of Participating Classes

E.5 ADDITIONAL REQUIREMENTS

E.5.1 Grouping of Objects in a Request

The device shall allow the management station to perform a single Get, GetNext, or Set operation on any combination of supported objects with the objects listed in any order within the message, unless otherwise restricted by this standard.

The device shall not associate any semantics to the ordering of objects within the varBindingsList. As required by RFC 1157, Section 4.1.5, each object shall be affected "as if simultaneously set with respect to all other assignments specified in the same message."

E.5.2 Support of Get

The device shall allow the management station to perform the Get operation on any supported object for which support for the Get Operation is indicated in Section E.4.

E.5.3 Support of GetNext

The device shall allow the management station to perform the GetNext operation on any OBJECT IDENTIFIER.

E.5.4 Support of Set

The device shall allow the management station to perform the Set operation on any supported object for which support for the Set Operation is indicated in Section E.4.

E.5.5 Performance

The DMS shall process the Get, GetNext, or Set request in accordance with all of the rules of NTCIP 1103, including updating the value in the database and initiating the transmission of the appropriate

response (assuming that the DMS has permission to transmit) within 1 second of receiving the last byte of the request.

NOTE: If a user desires a shorter response time, s/he will need to specify this in the specifications.

§